

# Data Studio 6.5.1

# User Manual

Issue 01  
Date 2019-05-24



**Copyright © Huawei Technologies Co., Ltd. 2019. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## **Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base  
Bantian, Longgang  
Shenzhen 518129  
People's Republic of China

Website: <http://www.huawei.com>

Email: [support@huawei.com](mailto:support@huawei.com)

---

# Contents

---

<b>1 Data Studio User Manual.....</b>	<b>1</b>
<b>2 About This Manual.....</b>	<b>2</b>
2.1 Overview.....	2
2.2 Intended Audience.....	2
2.3 Change History.....	2
2.4 Document Conventions.....	6
2.5 Third Party Licenses.....	7
2.6 Reference Documents.....	7
<b>3 About Data Studio.....</b>	<b>9</b>
3.1 Overview.....	9
3.2 Supported Functions.....	10
3.3 Constraints and Limitations.....	18
3.4 Structure of Release Package.....	20
3.5 System Requirements.....	21
<b>4 Installing Data Studio.....</b>	<b>24</b>
4.1 Installing and Configuring Data Studio.....	24
4.2 Configuring a Cluster Database.....	33
<b>5 Getting Started.....</b>	<b>39</b>
5.1 Starting Data Studio.....	39
5.2 Data Studio User Interface.....	40
5.3 Data Studio Menus.....	41
5.3.1 File.....	41
5.3.2 Edit.....	42
5.3.3 Run.....	46
5.3.4 Debug.....	46
5.3.5 Settings.....	47
5.3.6 Help.....	47
5.4 Data Studio Toolbars.....	48
5.5 Data Studio Right-Click Menus.....	49
<b>6 Using Data Studio.....</b>	<b>55</b>
6.1 Overview.....	55

6.2 Connection Profiles.....	55
6.2.1 Overview.....	56
6.2.2 Adding a Connection.....	56
6.2.3 Renaming a Connection.....	64
6.2.4 Editing a Connection.....	64
6.2.5 Removing a Connection.....	65
6.2.6 Viewing Connection Properties.....	65
6.2.7 Refreshing a Database Connection.....	65
6.3 Databases.....	69
6.3.1 Creating a Database.....	70
6.3.2 Disconnecting All Databases.....	70
6.3.3 Connecting to Database.....	71
6.3.4 Disconnecting Database.....	71
6.3.5 Renaming a Database.....	72
6.3.6 Dropping a Database.....	72
6.3.7 Viewing a Database Properties.....	72
6.4 Schemas.....	73
6.4.1 Overview.....	73
6.4.2 Creating a Schema.....	73
6.4.3 Exporting Schema DDL.....	74
6.4.4 Exporting Schema DDL and Data.....	75
6.4.5 Renaming a Schema.....	76
6.4.6 Support Sequence DDL.....	77
6.4.7 Grant/Revoke Privilege.....	77
6.4.8 Dropping a Schema.....	78
6.5 Functions/Procedures.....	78
6.5.1 Creating Function/Procedure.....	78
6.5.2 Editing a Function/Procedure.....	80
6.5.3 Grant/Revoke Privilege.....	81
6.5.4 Working with Functions/Procedures.....	82
6.5.4.1 Overview.....	82
6.5.4.2 Debugging a PL/SQL Function.....	82
6.5.4.2.1 Overview.....	82
6.5.4.2.2 Using Breakpoints.....	82
6.5.4.2.3 Controlling Execution.....	93
6.5.4.2.4 Checking Debug Information.....	96
6.5.4.3 Selecting a DB Object in the PL/SQL Viewer.....	98
6.5.4.4 Exporting a Function/Procedure DDL.....	100
6.5.4.5 Viewing Object Properties in the PL/SQL Viewer.....	101
6.5.4.6 Dropping a Function/Procedure.....	102
6.5.4.7 Executing a Function/Procedure.....	103
6.5.4.8 Grant/Revoke Privilege.....	104

6.6 Tables(GaussDB 200&GaussDB 300).....	104
6.6.1 Overview.....	104
6.6.2 Creating Regular Table.....	104
6.6.2.1 Overview.....	104
6.6.2.2 Working with Columns.....	112
6.6.2.3 Working with Constraints.....	114
6.6.2.4 Working with Indexes.....	115
6.6.3 Creating Foreign Table.....	117
6.6.4 Creating Partition Table.....	117
6.6.4.1 Overview.....	117
6.6.4.2 Working with Partitions.....	122
6.6.5 Grant/Revoke Privilege - Regular/Partition Table.....	123
6.6.6 Managing Table.....	123
6.6.6.1 Overview.....	123
6.6.6.2 Renaming a Table.....	125
6.6.6.3 Truncating a Table.....	125
6.6.6.4 Reindexing a Table.....	125
6.6.6.5 Analyzing a Table.....	126
6.6.6.6 Vacuuming a Table.....	126
6.6.6.7 Setting the Table Description.....	126
6.6.6.8 Setting the Tablespace.....	126
6.6.6.9 Setting the Schema.....	127
6.6.6.10 Dropping a Table.....	127
6.6.6.11 Viewing Table Properties.....	127
6.6.6.12 Grant/Revoke Privilege.....	128
6.6.7 Managing Table Data.....	129
6.6.7.1 Overview.....	129
6.6.7.2 Exporting Table DDL.....	129
6.6.7.3 Exporting Table DDL and Data.....	130
6.6.7.4 Exporting Table Data.....	131
6.6.7.5 Showing DDL.....	133
6.6.7.6 Importing Table Data.....	133
6.6.7.7 Viewing Table Data.....	135
6.6.7.8 Editing Table Data.....	136
6.6.8 Editing Temporary Tables.....	141
6.7 Tables(GaussDB 100).....	141
6.8 Sequences.....	147
6.8.1 Creating Sequence.....	147
6.8.2 Grant/Revoke Privilege.....	148
6.8.3 Working with Sequences.....	148
6.9 Views.....	149
6.9.1 Creating a View.....	149

6.9.2 Grant/Revoke Privilege.....	150
6.9.3 Working with Views.....	150
6.10 Tablespaces.....	153
6.10.1 Creating a Tablespace.....	153
6.10.2 Working with Tablespaces.....	156
6.11 Users/Roles.....	159
6.11.1 Create User/Role.....	159
6.11.2 Working with Users/Roles.....	160
6.12 SQL Terminal.....	161
6.12.1 Opening SQL Terminals.....	161
6.12.2 Managing SQL Query Execution History.....	166
6.12.3 Opening And Saving SQL Scripts.....	168
6.12.4 Viewing Object Properties in the SQL Terminal.....	170
6.12.5 Canceling Execution of SQL Queries.....	171
6.12.6 Formatting of SQL Queries.....	172
6.12.7 Selecting a DB Object in the SQL Terminal.....	178
6.12.8 Viewing the Query Execution Plan and Cost.....	179
6.12.9 Viewing the Query Execution Plan and Cost Graphically.....	182
6.12.10 Working with the SQL Terminals.....	185
6.12.11 Exporting Query Results.....	199
6.12.12 Managing SQL Terminal Connections.....	200
6.13 Batch Operation.....	201
6.13.1 Overview.....	201
6.13.2 Dropping Batch of Objects.....	202
6.13.3 Granting/Revoking Privileges.....	203
<b>7 Personalizing Data Studio.....</b>	<b>205</b>
7.1 Overview.....	205
7.2 General.....	205
7.3 Editor.....	209
7.4 Environment.....	213
7.5 Export.....	215
7.6 Result Management.....	215
7.7 Security.....	220
<b>8 References.....</b>	<b>224</b>
8.1 Performance Specification.....	224
<b>9 Troubleshooting.....</b>	<b>226</b>
<b>10 Security Management.....</b>	<b>234</b>
10.1 Overview.....	234
10.2 Login History.....	234
10.3 Password Expiry Notification.....	235
10.4 Securing the Application In-Memory Data.....	235

---

10.5 Data Encryption for Saved Data.....	235
10.6 SQL History.....	235
10.7 SSL Certificates.....	236
10.8 Verify Software Package Integrity.....	241
<b>11 FAQs.....</b>	<b>247</b>
<b>12 Glossary.....</b>	<b>253</b>

# 1 Data Studio User Manual

---

**Data Studio** is an Integrated Development Environment (IDE) that helps database developers to build the application conveniently. It supports essential features of the database.

This tool allows working with database objects with minimal programming knowledge. Data Studio provides you with various features, such as creating and managing database objects (database, schema, functions, procedures, tables, sequences, indexes, views, and tablespaces), executing SQL statements or SQL scripts, editing and executing PL/SQL statements, viewing graphically the query execution plan and cost, exporting table data, and debugging operations. It also provides SQL assistance for various queries/procedures/functions executed in SQL Terminal/PL/SQL Viewer.

# 2 About This Manual

---

- [2.1 Overview](#)
- [2.2 Intended Audience](#)
- [2.3 Change History](#)
- [2.4 Document Conventions](#)
- [2.5 Third Party Licenses](#)
- [2.6 Reference Documents](#)

## 2.1 Overview

This section provides information about this manual.

## 2.2 Intended Audience

This manual is intended for:

- Database Developers
- Database Administrators

The database developer must have a high-level technical understanding of the database.

The database administrator must be able to manage installation, handle operations, and solve problems.

## 2.3 Change History

The following table provides the change history for the Data Studio User Manual:

Date	Version	Change Description
2019-04-15	18.2.0.SPC1	Update and release

Date	Version	Change Description
2019-01-30	18.2.0	Updated: <b>6.5.1 Creating Function/Procedure</b> <b>6.12.3 Opening And Saving SQL Scripts</b> <b>6.6.4.1 Overview</b>
2018-10-31	18.1.0	In this version, Data Studio supports the GaussDB 100 product. Updated: <ul style="list-style-type: none"><li>● <b>Create User/Role</b></li><li>● <b>Working with Users/Roles</b></li><li>● <b>Viewing the Query Execution Plan and Cost</b></li><li>● <b>Viewing the Query Execution Plan and Cost Graphically</b></li><li>● <b>Working with the SQL Terminals</b></li></ul>

Date	Version	Change Description
2018-07-24	18.0.0	<p>Added:</p> <ul style="list-style-type: none"><li>● <a href="#">6.6.8 Editing Temporary Tables</a></li><li>● <a href="#">Comment/Uncomment</a></li><li>● <a href="#">Indent/Unindent Lines</a></li><li>● <a href="#">Insert Space</a></li><li>● <a href="#">Using Templates</a></li><li>● <a href="#">6.12.12 Managing SQL Terminal Connections</a></li><li>● <a href="#">6.13 Batch Operation</a></li><li>● <a href="#">Adding Templates</a></li><li>● <a href="#">Formatter</a></li><li>● <a href="#">SQL History</a></li><li>● <a href="#">Query/Function/Procedure Backup</a></li><li>● <a href="#">Result Data Encoding</a></li></ul> <p>Updated:</p> <ul style="list-style-type: none"><li>● <a href="#">3.2 Supported Functions</a></li><li>● <a href="#">Search Objects</a></li><li>● <a href="#">5.3.6 Help</a></li><li>● <a href="#">6.2.2 Adding a Connection</a></li><li>● <a href="#">6.4.3 Exporting Schema DDL</a></li><li>● <a href="#">6.4.4 Exporting Schema DDL and Data</a></li><li>● <a href="#">6.5 Functions/Procedures</a></li><li>● <a href="#">6.5.1 Creating Function/Procedure</a></li><li>● <a href="#">6.5.4.2 Debugging a PL/SQL Function</a></li><li>● <a href="#">6.5.4.2.2 Using Breakpoints</a></li><li>● <a href="#">6.5.4.4 Exporting a Function/Procedure DDL</a></li><li>● <a href="#">6.6.6.11 Viewing Table Properties</a></li><li>● <a href="#">6.6.7.2 Exporting Table DDL</a></li><li>● <a href="#">6.6.7.3 Exporting Table DDL and Data</a></li><li>● <a href="#">6.6.7.7 Viewing Table Data</a></li><li>● <a href="#">6.6.7.8 Editing Table Data</a></li><li>● <a href="#">Exporting the View DDL</a></li><li>● <a href="#">6.12.2 Managing SQL Query Execution History</a></li><li>● <a href="#">6.12.9 Viewing the Query Execution Plan and Cost Graphically</a></li><li>● <a href="#">Execute SQL Queries</a></li><li>● <a href="#">Reverting to Default Templates</a></li><li>● <a href="#">Removing Templates</a></li></ul>

Date	Version	Change Description
2018-03-01	V100R003C30	<p>Added:</p> <ul style="list-style-type: none"><li>● <a href="#">6.12.9 Viewing the Query Execution Plan and Cost Graphically</a></li><li>● <a href="#">SQL Assistant</a></li></ul> <p>Updated:</p> <ul style="list-style-type: none"><li>● <a href="#">3.3 Constraints and Limitations</a></li><li>● <a href="#">3.5 System Requirements</a></li><li>● <a href="#">6.2.2 Adding a Connection</a></li><li>● <a href="#">6.2.4 Editing a Connection</a></li><li>● <a href="#">6.3.1 Creating a Database</a></li><li>● <a href="#">6.3.3 Connecting to Database</a></li><li>● <a href="#">6.3.5 Renaming a Database</a></li><li>● <a href="#">6.5.1 Creating Function/Procedure</a></li><li>● <a href="#">6.6.2 Creating Regular Table</a></li><li>● <a href="#">6.6.4 Creating Partition Table</a></li><li>● <a href="#">6.6.7.8 Editing Table Data</a></li><li>● <a href="#">6.12.10 Working with the SQL Terminals</a></li><li>● <a href="#">SQL Assistant</a></li><li>● <a href="#">Edit Table Data</a></li><li>● <a href="#">9 Troubleshooting</a></li><li>● <a href="#">11 FAQs</a></li></ul>
2017-11-17	V100R003C20	<p>Added:</p> <ul style="list-style-type: none"><li>● <a href="#">Search Objects</a></li><li>● <a href="#">6.6.3 Creating Foreign Table</a></li><li>● <a href="#">6.6.4 Creating Partition Table</a></li><li>● <a href="#">6.8 Sequences</a></li><li>● <a href="#">Error Locator</a></li></ul> <p>Updated:</p> <ul style="list-style-type: none"><li>● <a href="#">3.3 Constraints and Limitations</a></li><li>● <a href="#">3.5 System Requirements</a></li><li>● <a href="#">6.2.2 Adding a Connection</a></li><li>● <a href="#">6.5 Functions/Procedures</a></li><li>● <a href="#">6.6.7.4 Exporting Table Data</a></li><li>● <a href="#">6.6.7.6 Importing Table Data</a></li><li>● <a href="#">6.6.7.7 Viewing Table Data</a></li><li>● <a href="#">Rename SQL Terminal</a></li><li>● <a href="#">9 Troubleshooting</a></li><li>● <a href="#">11 FAQs</a></li></ul>

Date	Version	Change Description
2017-07-30	V100R003C10	Initial Release

## 2.4 Document Conventions

This section describes the content, symbols, GUI, and text conventions used in this manual.

### Content Conventions

The purchased products, services and features are stipulated by the contract made between Huawei Technologies Co., Ltd. and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

### Symbol Conventions

The symbol conventions that may be found in this document are defined as follows:

Convention	Description
 <b>NOTE</b>	Provides additional information to emphasize or supplement important points of the main text.
 <b>NOTICE:</b>	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.

### GUI Conventions

The GUI conventions that may be found in this document are defined as follows:

Convention	Description
<b>Boldface</b>	Buttons, menus, parameters, tabs, window, and dialog titles are in <b>boldface</b> . For example, click <b>OK</b> .
>	Multi-level menus are in <b>boldface</b> and separated by the ">" signs. For example, choose <b>File &gt; Create &gt; Folder</b> .

## Code Conventions

The code conventions that may be found in this document are defined as follows:

Convention	Description
[]	Brackets enclose one or more optional items.
<>	Brackets indicate user input (value that can be changed by the user).
{ }	Braces enclose two or more items, one of which is required.
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options.
. . . . . .	Vertical ellipsis points indicate that one or more lines of code that are not directly related to the example are omitted.

## 2.5 Third Party Licenses

This section contains the third party licenses applicable to the tool. Refer to *Open Source Software Notice Written Offer.pdf* for license information and details.

**Table 2-1** List of third party software

Third Party Software
Apache Commons Collections 4.1
Apache Jakarta Commons Math 3.2
Commons CSV 1.4
druid 1.1.6
eclipse-jee-mars-R-win32 4.5.2
Eclipse Nebula NatTable 1.5.0
jQuery JavaScript Library 1.12.4
JSqParser 1.1

## 2.6 Reference Documents

The section contains the details about the documents that can be referred for using Data Studio.

**Table 2-2** Reference Documents

Document Name	Description
Open Source Software Notice Written Offer.pdf	Written offer listing the open source software used and their license information
GaussDB 200 Product Documentation GaussDB 300 Product Documentation GaussDB 100 Product Documentation DWS Product Documentation	Database Reference

# 3 About Data Studio

---

- [3.1 Overview](#)
- [3.2 Supported Functions](#)
- [3.3 Constraints and Limitations](#)
- [3.4 Structure of Release Package](#)
- [3.5 System Requirements](#)

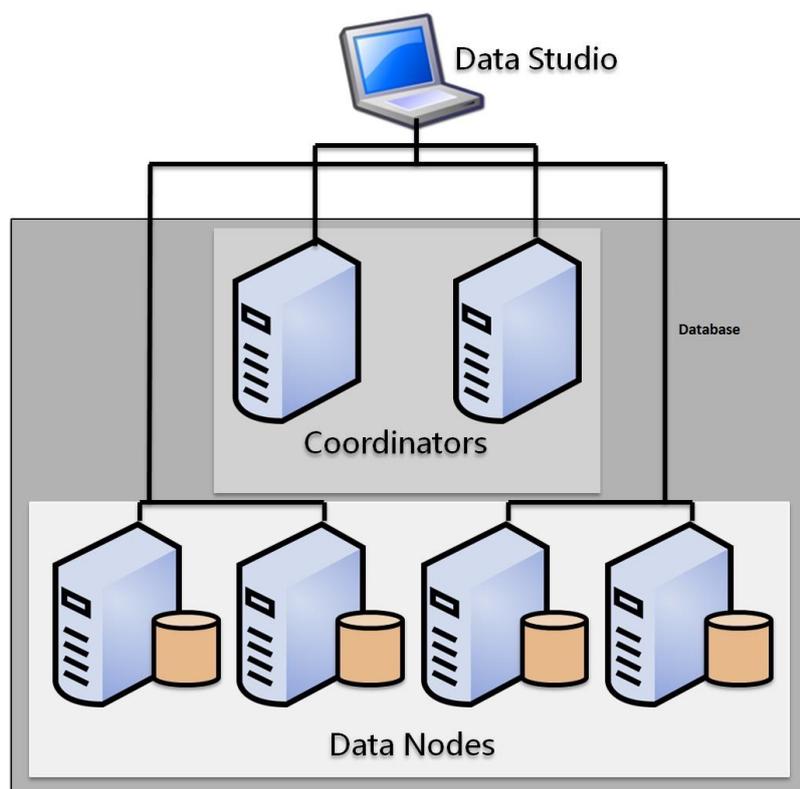
## 3.1 Overview

Data Studio provides a graphical interface which supports essential features of the database. This simplifies database development and application building tasks.

Data Studio allows the database developer to manage and create database objects (database, schema, functions, procedures, tables, sequences, columns, indexes, constraints, views, and tablespaces), executing SQL statements or SQL scripts, editing and executing PL/SQL statements, and importing and exporting table data.

Data Studio also allows the database developer to debug and fix defects in the PL/SQL code using debugging operations such as step into, step out, step over, continue, and terminate.

The following figure provides the operational context of database and Data Studio.



## 3.2 Supported Functions

Data Studio supports Online Analytical Processing and Online Transaction Processing databases, Online Analytical Processing database refer to GaussDB 200, GaussDB300 and HUAWEI CLOUD DWS products, and Online Transaction Processing database refer to GaussDB 100 products.

Below table describes the functions/operations of Data Studio that are supported and unsupported by GaussDB 200, GaussDB300, DWS and GaussDB 100 :

### Connection Profiles

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Adding a Connection</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<b>Renaming a Connection</b>		✓		
<b>Editing a Connection</b>		✓		
<b>Removing a Connection</b>		✓		
<b>Viewing Connection Properties</b>		✓		

### Databases

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating a Database</b>	Supported in V100R002C70 and later.	✓	Supported in 1.2.x and later	Unsupporte d
<b>Disconnecting All Databases</b>		✓		
<b>Connecting to Database</b>		✓		
<b>Disconnecting Database</b>		✓		Supported in V300R001C00
<b>Renaming a Database</b>		✓		Unsupporte d
<b>Dropping a Database</b>		✓		
<b>Viewing a Database Properties</b>		✓		Supported in V300R001C00

### Schemas

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating a Schema</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupporte d
<b>Exporting DDL</b>		✓		
<b>Exporting DDL and Data</b>		✓		
<b>Renaming a Schema</b>		✓		
<b>Dropping a Schema</b>		✓		

### Functions/Procedures

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating Function/ Procedure</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Debugging a PL/SQL Function</b>		✓		Unsupported
<b>Dropping a Function/ Procedure</b>		✓		Supported in V300R001C00
<b>Exporting a Function/ Procedure DDL</b>		✓		Unsupported
<b>Editing a Function/ Procedure</b>		✓		
<b>Breakpoint Operations</b>		✓		
<b>Callstack Operations</b>		✓		
<b>Variables Operations</b>		✓		
<b>View Source</b>		✓		Supported in V300R001C00

 **NOTE**

Creating SQL function for GaussDB 100 database is not supported.

**Tables**

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating Regular Table</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<b>Creating Foreign Table</b>		✓		Unsupported
<b>Creating Partition Table</b>		✓		Supported in V300R001C00

**Managing Table and Managing Table Data**

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Renaming a Table</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Truncating a Table</b>		✓		Supported in V300R001C00
<b>Reindexing a Table</b>		✓		
<b>Analyzing a Table</b>		✓		
<b>Vacuuming a Table</b>		✓		
<b>Setting the Table Description</b>		✓		
<b>Setting the Tablespace</b>		✓		
<b>Setting the Schema</b>		✓		
<b>Dropping a Table</b>		✓		
<b>Viewing Table Properties</b>		✓		
<b>Exporting Table DDL</b>		✓		
<b>Exporting Table DDL and Data</b>	✓			
<b>Exporting Table Data</b>	✓			
<b>Showing DDL</b>	✓			
<b>Importing Table Data</b>	✓			
<b>Viewing Table Data</b>	✓		Supported in V300R001C00	
<b>Editing Table Data</b>	✓		Unsupported	

### Columns

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating New Column</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Rename Column</b>		✓		
<b>Toggle Not Null</b>		✓		
<b>Drop Column</b>		✓		
<b>Set Column Default</b>		✓		
<b>Change Data Type</b>		✓		

### Constraints

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating a Constraint</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Renaming a Constraint</b>		✓		
<b>Dropping a Constraint</b>		✓		

### Indexes

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating a New Index</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Renaming an Index</b>		✓		
<b>Changing the Tablespace</b>		✓		
<b>Changing the Fill Factor</b>		✓		
<b>Dropping an Index</b>		✓		

### Partitions

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Rename a Partition</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Drop a Partition</b>		✓		

### Sequences

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating Sequence</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<b>Dropping a Sequence</b>		✓		
<b>Dropping a Sequence Cascade</b>		✓		Unsupported

### Views

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Creating a View</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<b>Exporting the View DDL</b>		✓		Unsupported
<b>Dropping a View</b>		✓		Supported in V300R001C00
<b>Dropping a View Cascade</b>		✓		Unsupported
<b>Renaming a View</b>		✓		
<b>Setting the Schema for a View</b>		✓		

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<a href="#">Viewing the Show DDL</a>		✓		Supported in V300R001C00
<a href="#">Setting the Default Value for the View Column</a>		✓		Unsupported
<a href="#">Viewing the Properties of a View</a>		✓		Supported in V300R001C00

### Tablespaces

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<a href="#">Creating a Tablespace</a>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<a href="#">Renaming a Tablespace</a>		✓		
<a href="#">Renaming a Data File</a>	Unsupported	✗	Unsupported	
<a href="#">Setting Tablespace Options</a>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<a href="#">Setting Maximum Tablespace Size</a>		✓		
<a href="#">Dropping a Tablespace</a>				✓
<a href="#">Dropping a Data File</a>	Unsupported	✗	Unsupported	Supported in V300R001C00
<a href="#">Viewing the Tablespace DDL</a>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported

### Users/Roles

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Create User/Role</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported
<b>Viewing/Editing User/Role Properties</b>		✓		
<b>Viewing the User/Role DDL</b>		✓		

### Search Objects

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Search Objects</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Unsupported

### SQL Terminal

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Auto Commit</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001 C00
<b>Auto-suggest</b>		✓		
<b>Error Locator</b>		✓		
<b>Execute queries</b>		✓		
<b>SQL Assistant</b>		✗		Supported in V300R001 C00
<b>SQL Formatting</b>		✓		
<b>SQL History</b>		✓		
<b>Visual Explain Plan</b>	Supported in 6.5.RC1 and later	✗		Unsupported

### Results

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Export Results Data</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00
<b>Search Results</b>		✓		
<b>Toggle Show SQL</b>		✓		

### Batch Operations

Functions/Operations	GaussDB 200	GaussDB 300	DWS	GaussDB 100
<b>Batch Drop</b>	Supported in V100R002C70 and later	✓	Supported in 1.2.x and later	Supported in V300R001C00

## 3.3 Constraints and Limitations

The following are known limitations in Data Studio:

### Character Encoding

When viewing SQL, DDL, object names or data containing Chinese characters, Data Studio encoding needs to be set to **GBK** provided OS supports GBK. For more information on changing encoding settings, refer to [Environment > Session Setting](#).

### Connection Management

Comma is considered as delimiter in **Include/Exclude** fields in **Advanced** tab of add and edit connection window. Hence schema name having comma is not supported in **Include/Exclude** fields.

### Database Tables

- In the Create Table wizard > Index tab and in the Create Index wizard, the selected columns between list view, on remove, will not maintain order.
- When an operation has completed, and if the Data Studio window is not the active window of the operating system, then the message dialog is shown only when Data Studio window becomes active.
- The following limitations are applicable for **Edit Table Data** operations:
  - Entering expression values in **Edit Table Data** tab is not supported.

- Data Studio allows editing of only fetched records.
- Edit table filter feature will not highlight search words within HTML tags such as <, &, >.
- A cell containing single '&' in it will not be displayed in tooltip. A cell containing two consecutive '&' will display as single '&' in the tooltip.
- Row focus is not retained on a newly added row. User must click on the desired cell to start editing.

## Function/Procedure

Function/Procedure created in SQL Terminal or **Create Function/Procedure** wizard must end with / to indicate the end of function/procedure. Statements entered after a function/procedure without / at the end will be treated as a single query and may display errors during execution.

## General

- A maximum of 100 tabs can be opened in the editor area. Tabs are based on available resources of the host machine.
- A maximum of 63 characters (text only) is allowed for database object names (database, schema, function, procedure, table, sequence, constraint, index, view, and tablespace). There is no limit to the number of characters that can be used in expressions and descriptions in Data Studio.
- A maximum of 300 result tabs can be opened on a logged instance of Data Studio.
- If there are large objects loaded in Object Browser and Search Object window, then expanding of objects in Object Browser may be slow and Data Studio may become unresponsive.
- Resizing the width of a cell containing data exceeding the available display area may cause DS to become unresponsive.
- When the data in a table cell is more than 1000 characters, it will appear trimmed upto 1000 characters with "..." at the end.
  - If the user copies the data from a cell in a table or Result tab and pastes it on any editor (such as SQL terminal/PLSQL source editor, notepad or any other external editor application), the entire data is pasted.
  - If the user copies the data from a cell in a table or Result tab and pastes it on an editable cell (same or different), the cell shows only the first 1000 characters with "..." in the end.
  - When the table/Result tab data is exported, the exported file contains the whole data.

## Security

Data Studio validates SSL connection parameters only for the first connection profile. If a second connection is opened, then the connection uses the same SSL connection parameters when the Enable SSL field is checked.

### NOTE

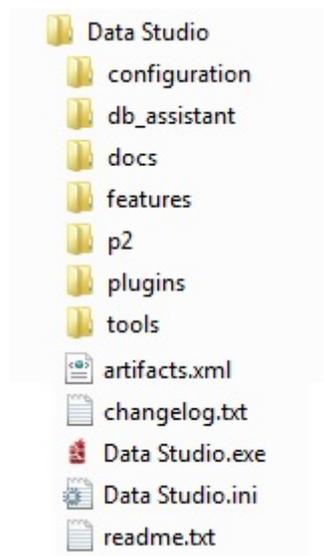
For SSL connection, if security files are corrupted, DS will not be able to proceed with any database operation. To recover from this, please remove security folder under the corresponding profile folder and restart DS.

## SQL Terminal

- Opening an SQL file containing a large number of queries may result in an 'Insufficient Memory' error. For more information, refer to [9 Troubleshooting](#).
- Data Studio does not disable the auto-suggest and hyperlink features in commented text in the **SQL Terminal**.
- Hyperlink feature is not supported if schema or table name have either space or dot (.) in them.
- Auto-suggest is not supported if the object name contains single or double quotes in them.
- If auto-suggest list retrieval takes more than 4 seconds, then "No proposal" is displayed.
- DS supports basic formatting of simple SELECT statements only and may not work as expected for complex queries.

## 3.4 Structure of Release Package

The release package structure of Data Studio is as follows:



Folders/Files	Description
<i>configuration</i>	Contains information about the application launcher and the required <b>Eclipse</b> plug-in path.
<i>db_assistant</i>	Contains SQL assistant related files.
<i>docs</i>	<ul style="list-style-type: none"> <li>● Contains <i>Data Studio User Manual.pdf</i> which provides you with details on using Data Studio.</li> <li>● Contains copyright notices, licenses, and the written offer for the open source libraries used in Data Studio.</li> </ul>
<i>features</i>	Contains Eclipse (rich client protocol-GUI) and Data Studio features.
<i>p2</i>	Contains files required for provisioning and managing <b>Eclipse</b> and <b>Equinox</b> -based applications.

Folders/Files	Description
<i>plugins</i>	Contains the required <b>Eclipse</b> and Data Studio plugins.
<i>tools</i>	Contains Data Studio dependent tools.
<i>UserData/</i> <ul style="list-style-type: none"> <li>● Autosave</li> <li>● Logs/</li> <li>● Preferences/</li> <li>● Profile/ <ul style="list-style-type: none"> <li>- History/</li> </ul> </li> <li>● Security/</li> </ul>	<p>Contains separate folders for each OS user who uses Data Studio.</p> <p><b>Autosave</b> - Contains the auto saved information of unsaved queries and functions/procedures.</p> <p><b>Logs</b> - Contains Data Studio.log which stores log information of all the operations performed in Data Studio.</p> <p><b>Preferences</b> - Contains the <i>Preferences.prefs</i> file which stores the custom preferences.</p> <p><b>Profile</b> - Contains the connection.properties, SQL History and Profiles.txt files required to manage connection profiles in Data Studio.</p> <p><b>Security</b> - Contains files required to manage security in Data Studio.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>● The UserData folder is created only after the first user opens an instance of Data Studio.</li> <li>● Logs folder, language, memory settings and log level are common for all users.</li> <li>● The Logs folder, <i>Data Studio.log</i> file, Preferences folder, Preferences.prefs file, Profile folder, connection.properties file, Profiles.txt file, and security folder are created after launching Data Studio.</li> <li>● If Logs folder path is provided in Data Studio.ini file, then logs are created in the specified path.</li> <li>● When user is not able to log in to the Data Studio due to security keys are corrupted. Follow the steps to generate new security keys: <ol style="list-style-type: none"> <li>1. Delete the security folder from Data Studio folder -&gt; UserData -&gt; security folder</li> <li>2. Restart Data Studio</li> </ol> </li> </ul>
<i>artifacts.xml</i>	Contains the product build information.
<i>changelog.txt</i>	Contains the detailed change log information of release version.
<i>Data Studio.exe</i>	Allows you to connect to the database and perform various operations (like managing database objects, editing or executing PL/SQL programs and so on).
<i>Data Studio.ini</i>	Contains run-time configuration information of Data Studio.
<i>readme.txt</i>	Contains the features or fixed issues of current release version.

## 3.5 System Requirements

This section provides the minimum system requirements for using Data Studio.

## Hardware Requirements

### NOTE

Pre requisite: User's home directory should have at-least 100MB free space.

The following table lists the minimum hardware requirements for Data Studio.

**Table 3-1** List of hardware requirements for Data Studio

Hardware Requirement	Configuration
CPU	x86 32-bit; 64-bit
Available RAM	A minimum of 1 GB of free memory.
Available Hard disk	A minimum of 1 GB of free space in DS installation location and 100MB free space in user's home directory(Usually in C Drive)
Network Requirements	Gigabit Ethernet

## Software Requirements

### Operating System Requirements

The following table lists the details of OS requirements for Data Studio.

**Table 3-2** List of OS environments for Data Studio

Operating System	Version
Microsoft Windows	Windows Server 2008 (32 bit)
	Windows Server 2008 (64 bit)
	7 (32 bit)
	7 (64 bit)
	8 (32 bit)
	8 (64 bit)
	10 (32 bit)
	10 (64 bit)

### Other Software Requirements

The following table lists the details of software requirements for Data Studio.

**Table 3-3** List of software requirements for Data Studio

Software	Specification
Java	Recommended Java version is 1.8.0_141 or above with appropriate bit number

Database Version Supported by Data Studio:

GaussDB 300:1.0.RC2

GaussDB 200: V100R006C10SPC008B003,  
V100R007C00SPC202B015, V100R007C10SPC202B011, V100R008C00SPC100B015, V100  
R008C10B012

GaussDB 100: V300R001C00SPC606T, V300R001C00SPC100B200 Debug 1a3e4bf

HUAWEI CLOUD DWS: 1.2.x, 1.3.x and 2.1.x

# 4 Installing Data Studio

---

[4.1 Installing and Configuring Data Studio](#)

[4.2 Configuring a Cluster Database](#)

## 4.1 Installing and Configuring Data Studio

This section describes the installation and configuration steps to be followed to use Data Studio. It also explains the steps to configure server for debugging PL/SQL Functions.

This section contains the following topics:

[Obtaining a Data Studio Software Package](#)

[Installing Data Studio](#)

[Configuring Data Studio](#)

[Providing Location to Create Log File](#)

[Controlling Exception and Error Logs](#)

[Description of the Log Message](#)

[Different Types of Log Level](#)

### Obtaining a Data Studio Software Package

- Enterprise users  
Log in to <http://support.huawei.com/enterprise/> and choose **Technical Support > Product Support > Software Download**. In the displayed page, use the keyword **Data Studio** to search for the package.
- Carrier users  
Log in to <http://support.huawei.com> and click **Software**. In the displayed page, use the keyword **Data Studio** to search for the package.

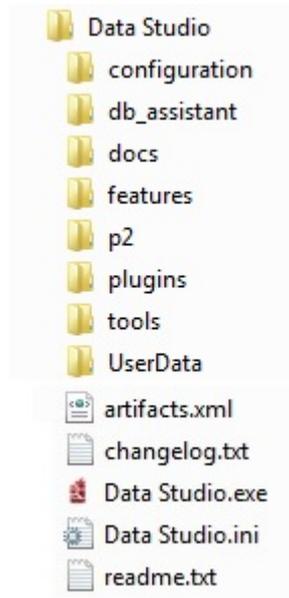
### Installing Data Studio

Data Studio can be run after extraction of package.

Follow the steps below to install Data Studio:

- Step 1** Unzip the required package (32-bit or 64-bit) to the program files (x86) or program files folder respectively. If the user prefer to install in other folder, then admin should control the folder access permissions to users.

You will see the following files and folders:



- Step 2** Locate and double-click *Data Studio.exe* to launch Data Studio.

**NOTE**

UserData folder is created after the first user launches Data Studio. Refer to [5.1 Starting Data Studio](#) in case of any error while launching Data Studio.

----End

To create a new database connection, refer to [6.2.2 Adding a Connection](#).

## Configuring Data Studio

Steps to configure Data Studio using *Data Studio.ini* file:

**NOTE**

Restart Data Studio to view parameter changes. Invalid parameters added in the configuration file are ignored by Data Studio. All the below mentioned parameters are not mandatory.

List of configuration parameters used in Data Studio:

**Table 4-1** Configuration Parameters

Parameter	Description	Value Range	Default Value
-startup	Defines the jar files required to load Data Studio. This information varies based on the version used.	N/A	<i>plugins/ org.eclipse.equinox.l auncher_1.3.100.v20 150511-1540.jar</i>
--launcher.library	Defines the libraries required to load Data Studio. This information varies based on the version used.	N/A	<i>plugins/ org.eclipse.equinox.l auncher.win32.win3 2.x86_1.1.300.v2015 0602-1417</i> or <i>plugins/ org.eclipse.equinox.l auncher.win32.win3 2.x86_64_1.1.300.v2 0150602-1417</i> based on the package used
-clearPersistedState	Removes any cached state of the user interface and reloads Data Studio	N/A	N/A <b>NOTE</b> It is recommended to add this parameter.
-consoleLineCount	Defines the maximum number of lines to be displayed in the <b>Messages</b> window.	1 - 5000	1000
-logfolder	Used to create log folder. The user can specify the path to save logs. If the default value "." is used, then the folder is created in <i>Data Studio\UserData\&lt;user name&gt;\logs</i> . Refer to <b>Providing Location to Create Log File</b> section for more information.	N/A	.
-loginTimeout	Defines the connection open wait time in seconds. Based on the duration value entered Data Studio will try to connect beyond which it throws time out error/ connection failed error	N/A	180

Parameter	Description	Value Range	Default Value
-sessionTimeout	Defines the session timeout in DataStudio. This value is in secs. Values range from 0 to 86400 , where 0 indicates disable timeout Example: - sessionTimeout=600 (10 mins)	0-86400	600
-data	Defines the instance data location for the session.	N/A	@none
@user.home/ MyAppWorkspace	Eclipse workspace is created in this location while Data Studio is being launched. @user.home refers to C:/Users/<username> Eclipse log files are available in @user.home/MyAppWorkspace/.metadata	N/A	N/A
-detailLogging	Defines the criteria with reference to logging error messages. Set to <b>True</b> to log all error messages. Set to <b>False</b> to log only error messages explicitly mentioned by Data Studio. Refer to <b>Controlling Exception and Error Logs</b> for more information. This parameter is not added by default and it can be set manually if logging is required.	True/False	False

Parameter	Description	Value Range	Default Value
-logginglevel	Creates the log files based on the value specified. If the value provided is arbitrary or empty, log files will be created as per WARN value. Refer to <a href="#">Different Types of Log Level</a> for more information.  This parameter is not added by default and it can be set manually if logging is required.	FATAL, ERROR, WARN, INFO, DEBUG TRACE, ALL, OFF	WARN
-focusOnFirstResult	Defines auto focus behavior for <b>Result</b> window.  Set to <b>false</b> to automatically set focus to the last opened <b>Result</b> window.  Set to <b>true</b> to disable the automatic set focus.	True/False	False
<p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>● All the above parameters must be added before -vmargs.</li> <li>● -startup and --launcher.library must be added as first and second parameter respectively.</li> </ul>			
-vmargs	Specifies the start of virtual machine arguments.  <b>NOTE</b> -vmargs must be the last parameter in the configuration file.	N/A	N/A
-vm <file name (javaw.exe) with relative path to Java executable>	Defines the relative path to Java executable	N/A	N/A
- Dosgi.requiredJava Version	Defines the minimum java version required to run Data Studio. This value must not be modified.	N/A	1.8  <b>Note:</b> Recommended Java version is 1.8.0_141

Parameter	Description	Value Range	Default Value
-Xms	<p>Defines the initial heap space that Data Studio consumes. This value must be in multiples of 1024 and greater than 40 MB and less than or equal to -Xmx size. Append the letter k or K to indicate kilobytes, m or M to indicate megabytes, g or G to indicate gigabytes. Few examples:</p> <ul style="list-style-type: none"><li>● -Xms40m</li><li>● -Xms120m</li></ul> <p>Refer to Java documentation for more information.</p>	N/A	-Xms40m
-Xmx	<p>Defines the maximum heap space that Data Studio consumes. This value can be modified based on the available RAM space. Append the letter k or K to indicate kilobytes, m or M to indicate megabytes, g or G to indicate gigabytes. Few examples:</p> <ul style="list-style-type: none"><li>● -Xmx1200m</li><li>● -Xmx1000m</li></ul> <p>Refer to Java documentation for more information.</p>	N/A	-Xmx1200m

Parameter	Description	Value Range	Default Value
-testability	<p>This parameter is used to enable testability features. For the current version the features controlled by this parameter with condition as <b>True</b> are:</p> <ul style="list-style-type: none"> <li>● Enables copy content of last triggered auto-complete operation using Ctrl+Space shortcut key.</li> <li>● <b>Execution Plan and Cost with Analyze</b></li> </ul>  <p>displays tree and graphical view of the explain plan.</p> <p>This parameter will not be available by default and needs to be added manually for testing.</p>	True/False	False
-Duser.language	<p>Defines the language settings for Data Studio. This parameter is added after the language setting is changed.</p>	zh/en	N/A
-Duser.country	<p>Defines the country settings for Data Studio. This parameter is added after the language setting is changed.</p>	CN/IN	N/A

## Providing Location to Create Log File

**Step 1** Open the *Data Studio.ini* file.

**Step 2** Provide the path for the *-logfolder* parameter.

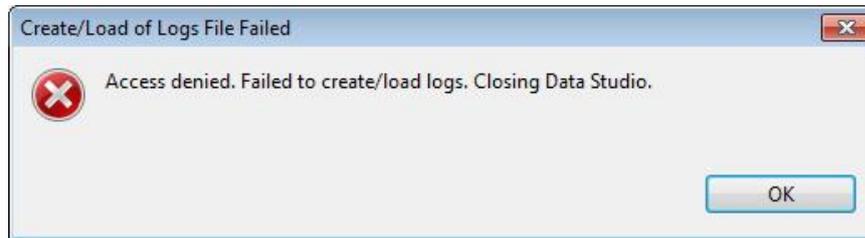
For example:

*-logfolder=c:\test1*

In this case, the *Data Studio.log* file is created in the *c:\test1\<user name>\logs* path.

**NOTE**

If any of the user does not have access to the path mentioned in the *Data Studio.ini* file, then Data Studio closes with the below pop-up message.

**----End**

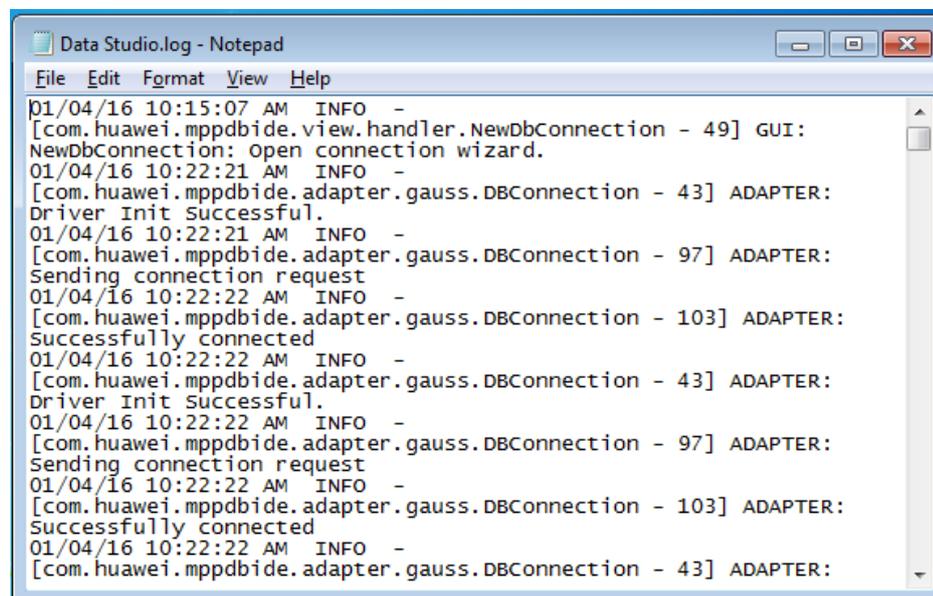
The *Data Studio.log* file will be created in the *Data Studio\UserData\<user name>\logs* path if

- The path is not provided in the *Data Studio.ini* file.  
For example: *-logfolder=.*
- The path provided does not exist.

**NOTE**

Refer to server manual for detailed information.

You can use any text editor to open and view the *Data Studio.log* file.



## Controlling Exception and Error Logs

The stack trace details of exception, error or throw-able are controlled based on the program argument parameter. This parameter is configured in the *Data Studio.ini* file.

*-detailLogging=false*

If the flag value is 'true', then the stack trace details of exception, error or throw-able will be saved in the log file.

If the flag value is 'false', then no stack trace details will be saved in the log file.

## Description of the Log Message

The description of the log message is as follows:



When the *Data Studio.log* file reaches the maximum file size of 10000 KB, it will create a new file and save as *Data Studio.log.1* automatically and the logs in *Data Studio.log* are moved to *Data Studio.log.1*. When *Data Studio.log* file reaches the maximum file size again, it will create a new file and save as *Data Studio.log.2*. The logs in *Data Studio.log* are moved to *Data Studio.log.1* and logs in *Data Studio.log.1* are moved to *Data Studio.log.2*. Latest logs are always written in *Data Studio.log* file. This process continues till *Data Studio.log.5* reaches the maximum file size and the cycle restarts.

### NOTE

To enable performance logging in the server log file, the configuration parameter *log\_min\_messages* must be enabled and value must be set as *debug1* in the configuration file *data/postgresql.conf*, that is, *log\_min\_messages = debug1*.

## Different Types of Log Level

The different types of log levels that are displayed in the *Data Studio.log* file are as follows:

- **TRACE**: The TRACE level provides detailed information than the DEBUG level and is at the top of the hierarchy.
- **DEBUG**: The DEBUG level indicates the granular information events that are most useful for debugging an application.
- **INFO**: The INFO level indicates the information messages that highlight the progress of the application.
- **WARN**: The WARN level indicates potentially harmful situations.
- **ERROR**: The ERROR level indicates error events.
- **FATAL**: The FATAL level indicates event(s) which cause the application to abort.
- **ALL**: The ALL level turns on all the log levels.
- **OFF**: The OFF level turns off all the log levels. This is opposite to ALL level.

### NOTE

- If the user enters an invalid value to log level, then log level will be set to WARN.
- If the user does not provide any log level, then log level will be set to WARN.

The logger outputs all messages equal to or greater than its log level.

The order of the standard log4j levels are as follows:

**Table 4-2** Logging Level

-	FATAL	ERROR	WARN	INFO	DEBUG	TRACE
OFF	✘	✘	✘	✘	✘	✘

-	FATAL	ERROR	WARN	INFO	DEBUG	TRACE
FATAL	✓	✗	✗	✗	✗	✗
ERROR	✓	✓	✗	✗	✗	✗
WARN	✓	✓	✓	✗	✗	✗
INFO	✓	✓	✓	✓	✗	✗
DEBUG	✓	✓	✓	✓	✓	✗
TRACE	✓	✓	✓	✓	✓	✓
ALL	✓	✓	✓	✓	✓	✓
✓ - Log file created   ✗ - Log file not created						

## 4.2 Configuring a Cluster Database

This section describes how to modify the configuration file of a cluster database to:

- (Mandatory) Connect to the database based on a whitelist through a local host running the Windows OS (referred to as a Windows host). The database can be GaussDB 100, GaussDB 300, or GaussDB 200.
  - For details about how to configure a whitelist for GaussDB 200 and GaussDB 300, see [Configuring a Whitelist for GaussDB 200 or GaussDB 300](#).
  - Either of the following two ways can be used to configure a whitelist for GaussDB 100:
    - **Configuring a User Whitelist for GaussDB 100:** Allow specific database users to connect to a database from specific IP addresses.
    - **Configuring an IP Address Whitelist for GaussDB 100:** Allow any database users to connect to a database from specific IP addresses.

 NOTE

Note that GaussDB 100 also supports an IP black list, and the IP address blacklist has a higher priority than an IP address whitelist. If a user or IP address whitelist is configured but database access fails, check whether the IP address for accessing is on the IP address blacklist.

- Query for a configured IP address blacklist.

```
zsqli gaussdba/ChangeMe_123@127.0.0.1:1888
SELECT VALUE FROM V$PARAMETER WHERE NAME = 'TCP_EXCLUDED_NODES';
```

If an IP address to be used for remotely connecting to the database is on the IP address blacklist, record all the IP addresses on the blacklist.

- Delete the IP address from the IP address blacklist.

Run the following commands to reconfigure the IP address blacklist. A new IP address blacklist overwrites the existing IP address blacklist in the system. Therefore, the IP addresses in the following commands are those on the original blacklist and the IP address to be used for remotely connecting to the database must be excluded.

Assume that the original blacklisted IP addresses are **IP1, IP2, ..., IPn** and that the IP address to be used for remote connection is **IP1**. Do as follows:

- If GaussDB 100 is installed on a single host, run the following command when the database is connected (the configuration takes effect immediately):

```
ALTER SYSTEM SET TCP_EXCLUDED_NODES = '(IP2, ..., IPn)';
```

- If GaussDB 100 is installed on multiple hosts, run the following command when the database is disconnected (the configuration takes effect immediately):

```
gs_gucZenith -c "TCP_EXCLUDED_NODES=(IP2, ..., IPn)" -I
CN1_ID,CN2_ID, ... CNn_ID
```

- (Optional) Debug PL/SQL functions, supporting only GaussDB 200 and GaussDB 300. For details, see [Configuring Support for Debugging PL/SQL Functions for GaussDB 200 and GaussDB 300](#)

## Configuring a Whitelist for GaussDB 200 or GaussDB 300

- Configure the **pg\_hba.conf** file to allow users to connect to a database through a local Windows host.
  - a. Log in to any host in a ElkdWSGaussDB 300 cluster as user **omm**. Log in to a node where the MPPDB service resides as the OS user **omm**. Run **source \$ {BIGDATA\_HOME}/mppdb/.mppdbgs\_profile** to start environment variables.
  - b. Configure the host to allow users to connect to the database server through the local Windows host.

The following command allows user **jack** to remotely connect to the database from the 192.168.1.1 client.

```
gs_guc set -Z coordinator -N all -I all -h "host all jack 192.168.1.1/32
sha256"
```

 NOTE

- Connect to the database as a common user, rather than user **omm**.
- Parameters in the command above are as follows:
  - **-Z coordinator** indicates that the instance type is **coordinator**.
  - **-N all** indicates all hosts in the cluster.
  - **-I all** indicates all instances of the host.
  - **-h** indicates statements that need to be added in the **pg\_hba.conf** file.
  - **host all** indicates that the client can connect to any host where a CN resides in the database cluster.
  - **jack** indicates the user that accesses the database. You can run the following command to create the user **jack** in advance:

```
postgres=# CREATE USER jack PASSWORD 'Gaussdba@Mpp';
```
  - **192.168.1.1/32** indicates the host that can connect to the database. You can replace the IP address with the IP address of any local Windows host.  
Configure the parameters based on your network conditions. For example, if **192.168.1.1/32** is specified, only the specified host can connect to the database. If **192.168.1.0/24**, **192.168.0.0/16**, or **192.0.0.0/8** is specified, all hosts on the specified network segment can connect to the database.
  - **sha256** indicates that the password of user **jack** is encrypted using the SHA-256 algorithm.

## Configuring a User Whitelist for GaussDB 100

To configure a user whitelist, modify the **zhba.conf** file on each CN in GaussDB 100. Specifically, add database users and the IP addresses of Windows hosts to the file as a database access whitelist so that the listed users can connect to the database from the Windows hosts using the specific IP addresses.

- If GaussDB 100 is installed on a single host, perform the following operations on only this host.
- If GaussDB 100 is installed on multiple hosts, perform the following operations on each host where a CN resides. In this way, users can connect to the database from any of these hosts.

The configuration procedure is as follows:

1. Log in to a GaussDB 100 host as user **root**.
2. Add the local IP address to the **zhba.conf** file
  - a. Search for the **zhba.conf** file.

```
find / -name 'zhba.conf' -print
```
  - b. Navigate to the path where the **zhba.conf** file is stored.  
Assume that the **zhba.conf** file path is **/home/gaussdba/.viminfo**. Run the following command:

```
cd /home/gaussdba/.viminfo
```
  - c. Edit the **zhba.conf** file.

```
vi zhba.conf
```
  - d. Press **i** to add a user to the whitelist.  
After that, press **ESC**, and enter **:wq** to save the changes and exit.  
Assume that the database user is **jack** and the local IP address is **192.168.1.1**.

```
host jack 192.168.1.1
```

 NOTE

- You can add an IPv4 address, IPv4 network segment, IPv6 address, or IPv6 network segment. For example:

```
host user 192.168.1.1
host user 192.168.1.1/24
host user 20AB::9217:acff:feab:acd0
host user 20AB::9217:acff:feab:acd0/64
```

In this example:

**192.168.1.1** indicates an IPv4 host.

**192.168.1.1/24** indicates all IP addresses on the 192.168.1.0 IPv4 segment.

**20AB::9217:acff:feab:acd0** indicates an IPv6 host.

**20AB::9217:acff:feab:acd0/64** indicates all IP addresses on the 20AB::/64 IPv6 segment.

- You can run the following command to create the user **jack**:

```
create user jack identified by xxxxxxxx;
```

3. Load the user whitelist online to make it take effect.

- a. Switch to user **gaussdba**.

```
su gaussdba
```

- b. Connect to the database.

```
zsql gaussdba/Changeme_123@127.0.0.1:1171
```

- c. Load the user whitelist, which will take effect immediately.

```
ALTER SYSTEM RELOAD HBA CONFIG;
```

 NOTE

After a whitelist is configured, you can run the following command to query the **V\$HBA** view and check whether the whitelist is configured successfully:

```
SELECT * FROM SYS.V$HBA;
```

When GaussDB 100 is deployed on multiple hosts, perform the above operations on the hosts where other CNs reside in the cluster.

After the configuration is complete, you can use Data Studio to remotely connect to a GaussDB 100 database server by using a listed username, its password, the server IP address, and the port number.

## Configuring an IP Address Whitelist for GaussDB 100

To configure an IP address whitelist, modify the **engine.ini** file on each host where a CN resides in GaussDB 100. Specifically, add the IP addresses of Windows hosts to the file as a database access whitelist so that any users can connect to the database from the Windows hosts using the listed IP addresses.

- If GaussDB 100 is configured on a single host, perform the following operations to configure a new IP address whitelist:

- a. Log in to a GaussDB 100 host as user **root** and switch to the user **gaussdba**, an administrator.

- b. Connect to the database.

```
zsql gaussdba/Changeme_123@127.0.0.1:1888
```

- c. Query for a configured IP address whitelist, and record it.

```
SELECT VALUE FROM V$PARAMETER WHERE NAME = 'TCP_INVITED_NODES';
```

- d. Reconfigure the IP address whitelist.

Add new IP addresses to the IP address whitelist, which will overwrite original addresses. In other words, the original IP address whitelist is deleted.

```
ALTER SYSTEM SET TCP_INVITED_NODES = '(192.168.1.1, 192.168.2.*)';
```

 **NOTE**

After the configuration is complete, ensure that the IP whitelist detection function is enabled. Otherwise, the configuration does not take effect. To enable the IP address whitelist detection function online, run the following command:

```
ALTER SYSTEM SET TCP_VALID_NODE_CHECKING = true;
```

The command needs to be executed only once across the lifecycle of a cluster. The function takes effect immediately after the command execution, and there is no need to restart the database.

Then, you can use Data Studio to remotely connect to a GaussDB 100 database server by using any username, its password, the server IP address, and the port number.

- If GaussDB 100 is configured on multiple hosts, perform the following operations to configure a new IP address whitelist:
  - a. Log in to a GaussDB 100 host as user **root** and switch to the user **gaussdba**, a cluster administrator.
  - b. Query for CN IDs in the cluster, and record them.
 

```
gs_om -t status
```
  - c. Connect to the database.
 

```
zsql gaussdba/Changeme_123@127.0.0.1:8000
```
  - d. Query for a configured IP address whitelist, and record it.
 

```
SELECT VALUE FROM V$PARAMETER WHERE NAME = 'TCP_INVITED_NODES';
```
  - e. Disconnect the database.
 

```
exit
```
  - f. Reconfigure the IP address whitelist.
    - Add new IP addresses to the IP address whitelist, which will overwrite original addresses. In other words, the original IP address whitelist is deleted.
    - Configure the whitelist on all the CNs so that users can connect to the database through any host where a CN resides in the cluster.
 

```
gs_gucZenith -c "TCP_INVITED_NODES=(127.0.0.1,::1,192.168.1.1,192.168.1.2)" -I cn_402,cn_401
```
  - g. Check whether the IP address whitelist detection function is enabled.
 

```
SELECT VALUE FROM V$PARAMETER WHERE NAME = 'TCP_VALID_NODE_CHECKING';
```
  - h. If the result is **FALSE**, enable the whitelist detection function. The configuration takes effect immediately, and you do not need to restart the database.
 

```
gs_gucZenith -c "TCP_VALID_NODE_CHECKING=true" -I cn_401,cn_402
```

Then, you can use Data Studio to remotely connect to a GaussDB 100 database server by using any username, its password, the server IP address, and the port number.

## Configuring Support for Debugging PL/SQL Functions for GaussDB 200 and GaussDB 300

To debug PL/SQL functions, configure the **postgresql.conf** file and restart the cluster.

1. Log in to any host in a GaussDB 200/ElkDWS/GaussDB 300 cluster as user **omm**. Run **source \${BIGDATA\_HOME}/mppdb/mppdbgs\_profile** to start environment variables.
2. Configure the **postgresql.conf** file to provide support for debugging PL/SQL functions.
 

```
gs_guc set -Z coordinator -N all -c "shared_preload_libraries = '\$libdir/plugin_debugger'" -I all
```
3. Restart the cluster for the configuration to take effect.

```
gs_om -t stop  
gs_om -t start
```

# 5 Getting Started

[5.1 Starting Data Studio](#)

[5.2 Data Studio User Interface](#)

[5.3 Data Studio Menus](#)

[5.4 Data Studio Toolbars](#)

[5.5 Data Studio Right-Click Menus](#)

## 5.1 Starting Data Studio

This section describes the steps to be followed to start Data Studio.

### Prerequisites

The *StartDataStudio.bat* batch file checks the version of Operating System (OS), Java and Data Studio as a prerequisite to run Data Studio.

**Step 1** In the [Release package](#) navigate to Tools folder, locate and double-click *StartDataStudio.bat* to execute and check Java version compatibility.

The batch file checks the version compatibility and will launch Data Studio or display appropriate message based on OS, Java and Data Studio version installed.

If the Java version installed is below 1.8, then appropriate [error message](#) is displayed.

The scenarios checked by the batch file to confirm the required versions of the OS and Java for DS.

DS Installation (32/64bit)	OS (bit)	Java (bit)	Outcome
32	32	32	Launches Data Studio
32	64	32	Launches Data Studio
32	64	64	<a href="#">Error message</a> is displayed

DS Installation (32/64bit)	OS (bit)	Java (bit)	Outcome
64	32	32	<b>Error message</b> is displayed
64	64	32	<b>Error message</b> is displayed
64	64	64	Launches Data Studio

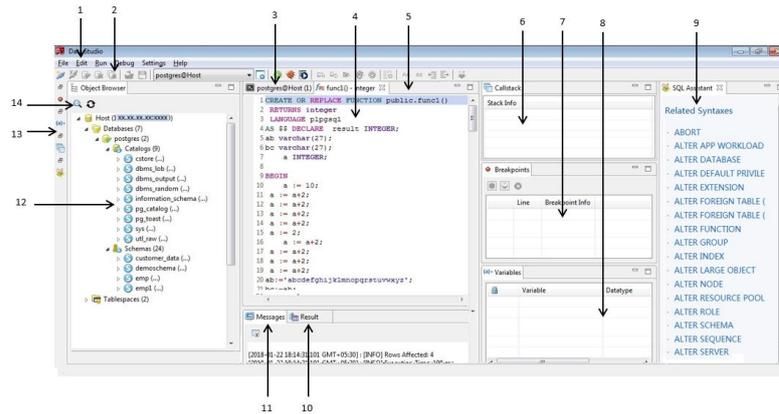
---End

## 5.2 Data Studio User Interface

This section describes the user interface of Data Studio.

The Data Studio user interface contains the following:

1. **Main Menu** provides basic operations.
2. **Toolbar** contains buttons for easy access to frequently used operations.
3. **SQL Terminal** tab is used to execute SQL statements and functions/procedures.
4. **PL/SQL Viewer** tab displays the content of functions/procedures.
5. **Editor Area** is used to perform edit operations.
6. **Callstack** pane shows the execution stack.
7. **Breakpoints** pane shows any breakpoints that have been set.
8. **Variables** pane shows variables and their values.
9. **SQL Assistant** tab displays suggestion or reference for the information entered in the SQL Terminal and PL/SQL Viewer.
10. **Result** tab displays the result(s) of an executed function/procedure, or an SQL statement.
11. **Messages** tab displays the output of a process, such as standard input, standard output, and standard error(s).
12. **Object Browser** contains a hierarchical tree display of database connection(s) and related database objects to which the user has access. All default created schemas except for public are grouped under **Catalogs** and user schemas are grouped under **Schemas** below the respective database.
13. **Minimized Window Panel** is used to open Callstack, Breakpoints and Variables pane. This panel is displayed only when Callstack, Breakpoints or Variables pane or all three are minimized.
14. **Search Toolbar** is used to search objects from the Object browser.



## 5.3 Data Studio Menus

### 5.3.1 File

The **File** menu contains database connection options. Click **File** from main menu or press **Alt +F** to open the **File** menu.

Function	Button	Shortcut Key	Description
New Connection		Ctrl+N	Creates and adds a new database connection to the <b>Object Browser</b> and <b>SQL Terminal</b> .
Remove Connection		-	Deletes the selected database connection from the <b>Object Browser</b> .
Connect To DB		-	Connects to the database.
Disconnect From DB		Ctrl+Shift+D	Disconnects from the selected database.
Disconnect All		-	Disconnects all the databases of a specified connection.
Open		Ctrl+O	Loads SQL queries into the <b>SQL Terminal</b> .
Save		Ctrl+S	Saves the SQL scripts of the <b>SQL Terminal</b> in an SQL file.
Save As		CTRL+ALT+S	Saves the SQL scripts of the <b>SQL Terminal</b> in a new SQL file
Exit	-	Alt+F4	Exits from Data Studio and closes the connection <b>NOTE</b> Any unsaved changes will be lost.

## Closing Data Studio

Follow the steps below to close Data Studio:

**Step 1** Click the  button.

Alternatively choose **File > Exit**.

**Exit Application** dialog box is displayed prompting you to take the required action.

**Step 2** Click the appropriate buttons based on your requirement.

- **Force Exit** button - To exit the application without saving the SQL History information.

 **NOTE**

Clicking on Force Exit button might not save the SQL History, if not saved yet.

- **Graceful Exit** button - To exit the application after saving the SQL History information to disk in case the save is not complete at this point of time.
- **Cancel** button - To cancel exiting from the application.

----End

## 5.3.2 Edit

The **Edit** menu contains clipboard, **Format**, **Find and Replace**, and **Search Objects** operations to use in the **PL/SQL Viewer** and **SQL Terminal** tab. Press **Alt+E** to open the **Edit** menu.

Function	Button	Shortcut Key	Description
Cut		Ctrl+X	Cuts the selected text
Copy		Ctrl+C	Copies the selected text or qualified object name
Paste		Ctrl+V	Pastes the selected text or qualified object name
Format		Ctrl+Shift+F	Formats all SQL statements and functions/procedures.
Select All	-	Ctrl+A	Selects all the text in SQL Terminal
Find and Replace		Ctrl+F	Finds and replaces text in SQL Terminal
Search Objects		Ctrl+Shift+S	Searches for objects within a connected database.
Undo		Ctrl+Z	Undoes the previous operation
Redo		Ctrl+Y	Redoes the previous operation

Function	Button	Shortcut Key	Description
UPPERCASE		Ctrl+Shift+U	Changes the case of the selected text to uppercase
lowercase		Ctrl+Shift+L	Changes the case of the selected text to lowercase
Go To Line		Ctrl+G	Skips to a specific line in the Terminal or PL/SQL Viewer

## Copy

Copy can also be used to copy objects from Object Browser.

The format of copied object name is:

Object Type	Copied Format
Functions/Procedures	schema.object name(parameter name parameter type,...)
Databases	object name
Schemas	object name
Tablespaces	object name
Columns	object name
Constraints	object name
Partition names	object name
All other object types	schema.object name

### NOTE

This feature is not supported by GaussDB 100 database.

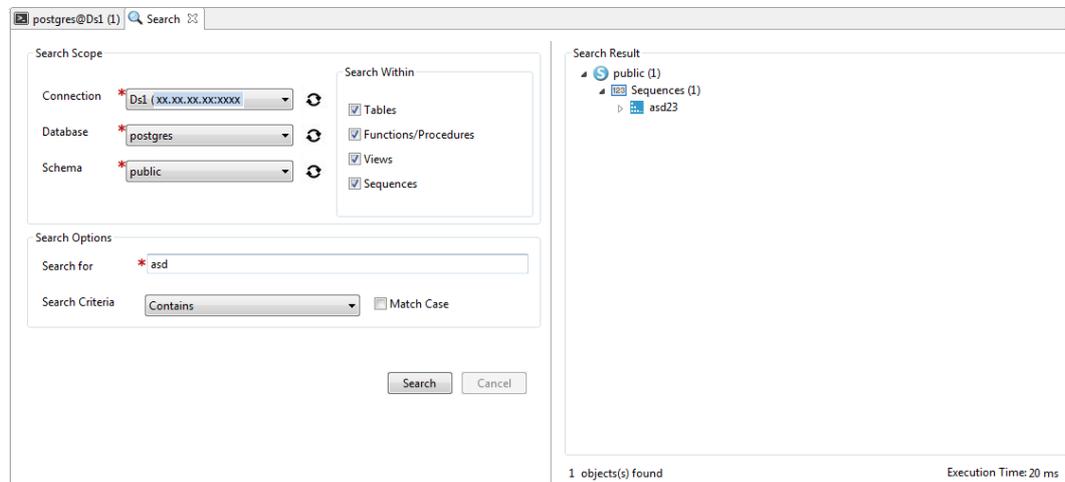
## Search Objects

Search Objects option allows you to search for object(s) from the Object Browser based on the search criteria entered. The search operation can be executed either from **Edit > Search**

**Objects** menu or by clicking the  from the Object Browser toolbar. The result of search displays tree structure similar to Object Browser. Right-click operations except for **Refresh** can be performed on the search result objects. Modified objects as a result of drop, set schema, rename, and so on can be viewed only from the main Object Browser after refresh. Right-click options on group names (tables, schema, views and so on) are not allowed on search result objects. Only objects to which you have access can be searched. Objects that you do not have access do not appear in the **Search Scope**.

 **NOTE**

Newly added objects can be viewed in the **Search** window by clicking the refresh option at the end of the object type.



Supported Search Options:

Search Options	Search Behavior
Contains	A search text which contains the searched characters will be displayed.
Starts With	A search text which starts with the searched character will be displayed.
Exact Word	A search text which matches exactly with searched characters will be displayed.
_	<p>A search text with _ (underscore) in it considers the underscore as a wildcard of single character. This does not apply to regular expression, starts with and exact word search.</p> <p><b>Example:</b></p> <ul style="list-style-type: none"> <li>● _ed, this displays all objects that starts with any single character followed by "ed".</li> <li>● D_t_e, this displays all objects that has character "d", followed by any single character, followed by character "t", followed by any single character, and followed by character "e".</li> </ul>
%	<p>A search text with % (percentage) in it considers the percentage as a wildcard of multiple characters. This does not apply to regular expression, starts with and exact word search.</p> <p><b>Example:</b></p> <ul style="list-style-type: none"> <li>● %ed, this displays all objects that has "ed" pattern in it.</li> <li>● D%t%e, this displays all objects that has character "d", followed by any number of characters, followed by character "t", followed by any number of characters, and followed by character "e".</li> </ul>

Search Options	Search Behavior
Regular Expression	<p>A search text with regular expression searches for similar pattern in Object Browser that fulfills the search condition. Select Regular Expression from <b>Search Criteria</b> drop-down to perform this search. For more information refer to POSIX Regular Expressions rules.</p> <p><b>Example:</b></p> <ul style="list-style-type: none"> <li>● ^a, this displays all objects that start with the letter a.</li> <li>● ^[^A-Za-z]+\$ , this displays all objects that do not have alphabets in them.</li> <li>● ^[^0-9]+\$ , this displays all objects that do not have numbers in them.</li> <li>● ^[a-t][^r-z]+\$ , this displays all objects whose name starts between a and t and excludes those that have characters between r and z in them.</li> <li>● ^e.*a\$, this displays all objects that starts with the letter e and ends with letter a.</li> <li>● ^[a-z]+\$ and select <b>Match Case</b>, this displays all objects that contains only alphabets in lower case.</li> <li>● ^[A-Z]+\$ and select <b>Match Case</b>, this displays all objects that contains only alphabets in upper case.</li> <li>● ^[A-Za-z]+\$ and select <b>Match Case</b>, this displays all objects that contains only alphabets in lower case and upper case.</li> <li>● ^[A-Za-z0-9]+\$ and select <b>Match Case</b>, this displays all objects that contains only alphabets in lower case, upper case and numbers.</li> <li>● ^".*\$", this displays all objects that are within in quotes.</li> </ul>

Underscore and % search:

Search Value	Search Behavior
_	<p>A search text with _ (underscore) in it considers the underscore as a wildcard of single character. This does not apply to regular expression, starts with and exact word search.</p> <p><b>Example:</b></p> <ul style="list-style-type: none"> <li>● _ed, this displays all objects that starts with any single character followed by "ed".</li> <li>● D_t_e, this displays all objects that has character "d", followed by any single character, followed by character "t", followed by any single character, and followed by character "e".</li> </ul>

Search Value	Search Behavior
%	<p>A search text with % (percentage) in it considers the percentage as a wildcard of multiple characters. This does not apply to regular expression, starts with and exact word search.</p> <p><b>Example:</b></p> <ul style="list-style-type: none"> <li>● %ed, this displays all objects that has "ed" pattern in it.</li> <li>● D%t%e, this displays all objects that has character "d", followed by any number of characters, followed by character "t", followed by any number of characters, and followed by character "e".</li> </ul>

Match case runs the search to match with the search text case.

### 5.3.3 Run

The **Run** menu contains options to execute a database object in the **PL/SQL Viewer** tab and to execute SQL statements in the **SQL Terminal** tab. Press **Alt+R** to open the **Run** menu.

Function	Button	Shortcut Key	Description
Execute DB Object		Ctrl+E	Starts execution (in normal mode) of the specified function/procedure. Displays the result in <b>Result</b> tab. Displays the information on actions performed in <b>Messages</b> tab.
Compile/Execute Statement		Ctrl+Enter	Compiles the function/procedure. Starts execution of SQL statements in the <b>SQL Terminal</b> tab.
Cancel		Shift+Esc	Cancels the executing query. Displays the result in <b>Result</b> tab. Displays the information on actions performed in <b>Messages</b> tab.

### 5.3.4 Debug

The **Debug** menu contains debug operations to use in the **PL/SQL Viewer** and **SQL Terminal** tab. Press **Alt+D** to open the **Debug** menu.

Function	Button	Shortcut Key	Description
Debug DB Object		Ctrl+D	Starts the debugging process.

Function	Button	Shortcut Key	Description
Continue		F9	Continues with debugging.
Terminate Debugging		F10	Terminates debugging.
Step Into		F7	Steps through the code statement.
Step Over		F8	Steps over the function.
Step Out		Shift+F7	Steps out of the function.

## 5.3.5 Settings

The **Settings** menu contains the option to change the language. Press **Alt+G** to open the **Settings** menu.

Function	Shortcut Key	Description
Language	-	Set the language for Data Studio user interface.
Preferences	-	Set the user preferences in Data Studio.

## Setting the Language

Follow the steps below to change the language from English to Chinese:

**Step 1** Choose **Settings > Language > (zh\_CN)中文(简体)(C)**.

**Restart Data Studio** dialog box is displayed.

### NOTE

Save all data before changing the language.

**Step 2** Click **Yes**.

If you click **No**, the language settings will be applied the next time you restart Data Studio.

The language will be set after you restart Data Studio.

----End

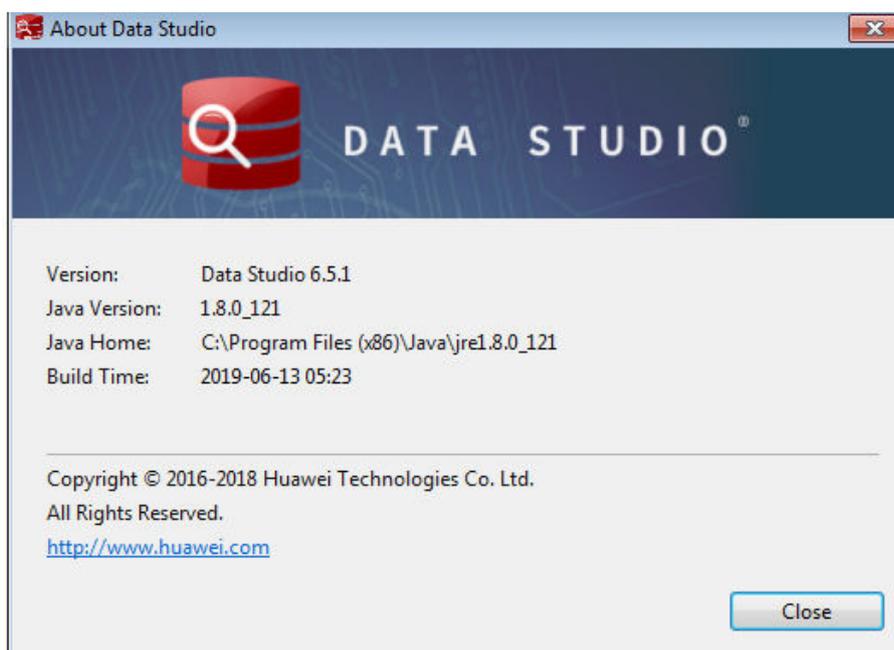
## 5.3.6 Help

The **Help** menu contains the user manual and version information of Data Studio. Press **Alt +H** to open the **Help** menu.

Function	Shortcut Key	Description
User Manual	F1	Opens the Data Studio User Manual.
About Data Studio	-	Displays the current version and copyright information of Data Studio.

**NOTE**

The *Version* displayed below is indicative and may not reflect the current version of Data Studio.

**NOTE**

Refer <https://java.com/en/download/help/path.xml> to set Java Home path.

## 5.4 Data Studio Toolbars

The following image shows the toolbar:



The toolbar contains the following actions:

- **New Connection**
- **Remove Connection**
- **Connect To DB**
- **Disconnect From DB**
- **Disconnect All**
- **Open SQL Script**

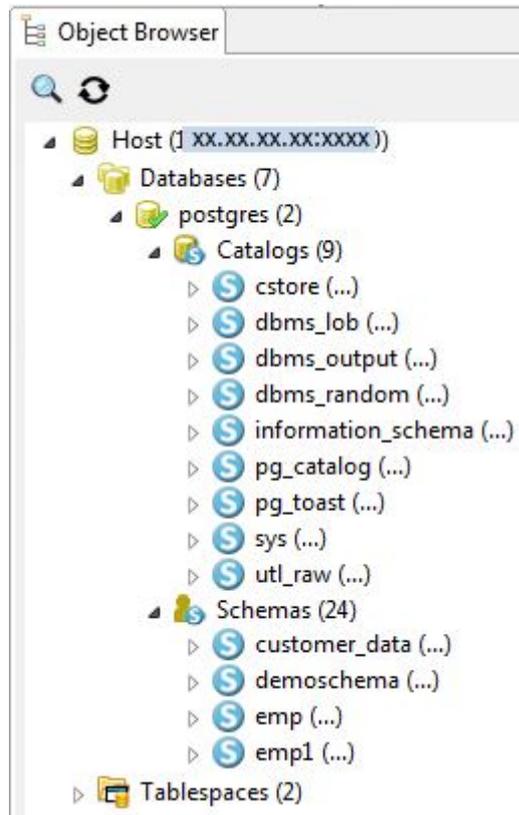
- **Save SQL Script**
- **Connection Profile Drop-down List**
- **Open New SQL Terminal**
- **Execute DB Object**
- **Debug DB Object**
- **Compile a Function/Procedure**
- **Step Into**
- **Step Out**
- **Step Over**
- **Terminate Debugging**
- **Continue Debugging**
- **Execution Plan and Cost**
- **Execution Plan and Cost - Graphical**
- **Format**
- **Upper Case**
- **Lower Case**
- **SQL Assistant**

## 5.5 Data Studio Right-Click Menus

This section describes the right-click menus of Data Studio.

### Object Browser Pane

The following image shows the **Object Browser** pane:



Right-clicking the connection profile allows you to select **Rename**, **Edit**, and **Remove Connection** along with **Refresh** options.

Menu Item	Shortcut Key	Description
Rename Connection	-	Renames a connection name.
Edit Connection	-	Modifies connection profile details.
Remove Connection	-	Removes the existing database connection.
Properties	-	Shows the properties of the connection.
Refresh	F5	Refreshes the connection.

Right-clicking the **Databases** group allows you to select **Create Database**, **Disconnect All**, and **Refresh** options.

Menu Item	Shortcut Key	Description
Create Database	-	Creates a new database in this connection.
Disconnect All	-	Disconnects all the databases of this connection.
Refresh	F5	Refreshes the databases group.

Right-clicking the active database allows you to select **Disconnect from DB**, **Open Terminal**, **Properties**, and **Refresh** options.

Menu Item	Shortcut Key	Description
Disconnect from DB	Ctrl+Shift+D	Disconnects the database
Open Terminal	Ctrl+T	Open a Terminal in this connection
Properties	-	Displays the properties of the database
Refresh	F5	Refreshes the database

Right-clicking the inactive database allows you to select **Connect to DB**, **Rename Database**, and **Drop Database** options.

Menu Item	Shortcut Key	Description
Connect to DB	-	Connects the database
Rename Database	-	Renames the database name
Drop Database	-	Drops the database

Right-clicking the **Catalogs** group allows you to select **Refresh** option.

Menu Item	Shortcut Key	Description
Refresh	F5	Refreshes the schema

Right-clicking the **Schemas** group allows you to select **Create Schema** and **Refresh** option.

Menu Item	Shortcut Key	Description
Create Schema	-	Creates a new schema
Grant/Revoke	-	Grant/Revoke access to schema group
Refresh	F5	Refreshes the schema

Right-clicking the schema allows you to select **Refresh**, **Rename Schema**, **Drop Schema**, **Export DDL**, and **Export DDL and Data** options.

Menu Item	Shortcut Key	Description
Export DDL	-	Exports DDL of the schema
Export DDL and Data	-	Exports DDL and data of the schema
Rename Schema	-	Renames a schema
Drop Schema	-	Drops a schema
Grant/Revoke	-	Grant/Revoke Access to schema
Refresh	F5	Refreshes the schema

Right-clicking **Functions/Procedures** allows you to select **Refresh** and **Create Function/Procedure** options.

Menu Item	Shortcut Key	Description
Create PL/SQL Function	-	Creates PL/SQL Function
Create PL/SQL Procedure		Creates PL/SQL Procedure
Create SQL Function		Creates SQL Function
Create C Function		Creates C Function
Grant/Revoke		Grant/Revoke access to Function/Procedure
Refresh	F5	Refreshes the Function/Procedure

Right-clicking **Tables** allows you to select **Refresh** and **Create table** options.

Menu Item	Shortcut Key	Description
Create Regular Table	-	Creates Regular table
Create Partition Table	-	Creates partition table
Grant/Revoke	-	Grant/revokes access of the table
Refresh	F5	Refreshes the table

Right-clicking **Views** allows you to select **Refresh** and **Create View** options.

Menu Item	Shortcut Key	Description
Create View	-	Creates View
Grant/Revoke	-	Grant/revokes the access of views

Menu Item	Shortcut Key	Description
Refresh	F5	Refreshes the View

Right-clicking the **PL/SQL Viewer** allows you to **Cut, Copy, Paste, Select All, Compile, Execute, and Debug** in the **PL/SQL Viewer** tab.

Right Click Options	Shortcut Key	Description
Cut, Copy, Paste	Ctrl+X, Ctrl+C, Ctrl+V	Clipboard operations
Select All	Ctrl+A	Selects the content in PL/SQL Viewer
Comment/Uncomment Lines	-	Comment/Uncomment selected line
Comment/Uncomment Block	-	Comment/Uncomment selected Block.
Compile	-	Compiles the function/procedure
Execute	-	Executes the function/procedure
Add Variable To Monitor	-	Adds selected variable to the monitor window
Debug with Rollback		Debug Function/Procedure and rollback changes after debug completion.
Debug	-	Debugs the function/procedure

Right-clicking the **SQL Terminal** allows you to **Cut, Copy, Paste, Select All, Open, Save, Execute Statement, Find and Replace, Execution Plan, Comment/Uncomment, and Cancel** in the **SQL Terminal** tab.

Right Click Options	Shortcut Key	Description
Cut, Copy, Paste	Ctrl+X, Ctrl+C, Ctrl+V	Clipboard operations
Select All	-	Selects all text
Execute Statement	-	Executes query
Open	-	Opens file
Save	-	Saves the query
Find and Replace	-	Finds and replaces text in SQL Terminal
Execution Plan	-	Executes the query

Right Click Options	Shortcut Key	Description
Comment/Uncomment Lines	Ctrl+/'	Comment/Uncomment selected lines
Comment/Uncomment Block	Ctrl+Shift+/'	Comment/Uncomment selected block of lines
Cancel	-	Cancels execution
Save As	CTRL+ALT+S	Saves the query in a new file .
Format	CTRL+SHIFT +F	Formats the selected SQL statements.

Right-clicking in the **Messages** tab allows you to **Copy**, **Select All**, and **Clear** the contents of the **Messages** tab.

Right Click Options	Shortcut Key	Description
Copy	Ctrl+C	Copies the text
Select All	Ctrl+A	Selects all text
Clear	-	Clears the text

# 6 Using Data Studio

---

- 6.1 Overview
- 6.2 Connection Profiles
- 6.3 Databases
- 6.4 Schemas
- 6.5 Functions/Procedures
- 6.6 Tables(GaussDB 200&GaussDB 300)
- 6.7 Tables(GaussDB 100)
- 6.8 Sequences
- 6.9 Views
- 6.10 Tablespaces
- 6.11 Users/Roles
- 6.12 SQL Terminal
- 6.13 Batch Operation

## 6.1 Overview

This section provides details on how to use features of Data Studio. Data Studio requires a valid database connection to perform operations on the database.

Refer to [3.2 Supported Functions](#) to see the list of functions and database(s) supported by Data Studio.

## 6.2 Connection Profiles

## 6.2.1 Overview

When Data Studio is started, the **New Database Connection** dialog box will open by default. To perform any DB operations, Data Studio must be connected to at least one database.

Enter the connection parameters to create a new database connection between Data Studio and the database in the server. Hovering over the connection name will display the server information.

### NOTE

You need to fill all the mandatory parameters, that are marked with asterisk (\*) to complete the operation successfully.

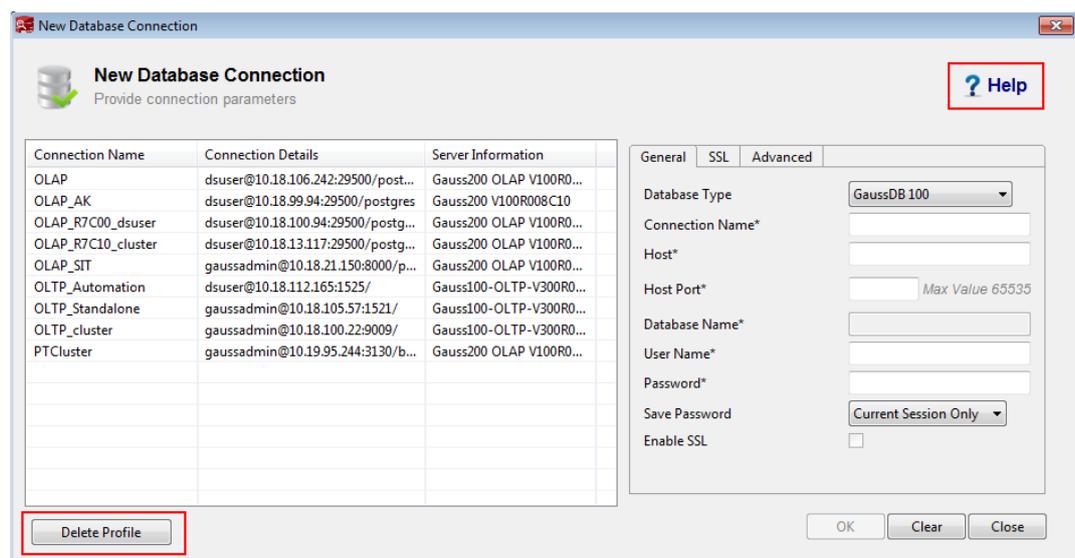
## 6.2.2 Adding a Connection

Follow the steps to establish a new database connection:

**Step 1** Choose **File > New Connection** from the main menu, or

click  on the toolbar, or press **Ctrl+N** to connect to the database.

The **New Database Connection** dialog box is displayed.



### NOTE

While establishing a connection, if the preference file is corrupted or the preferences values are invalid, then an error message is displayed informing you that preference values are invalid and default values are set for preferences. To complete establishing a new database connection operation, click **OK**.

**Step 2** The table on the left lists the details of the existing connection profile(s) used to connect to the database along with the server information.

### NOTE

The server information will be displayed only after one successful connection.

- Double clicking a connection name populates the connection parameters such as **Connection Name**, **Host**, and **Host Port**.

 **NOTE**

If password is corrupted for any of the existing connection profile or the key is corrupted, then the password field needs to be filled in for all created connections.

- Clicking  displays different pop-up messages based on the connection status of database.
  - If the database connection is active, then **Remove Connection Confirmation** pop-up is displayed. Click **Yes** to disconnect all databases.
  - If the database connection is not active, then **Remove Connection Confirmation** pop-up is displayed.
- Clicking  without a connection name displays a pop up stating to select at least one connection profile.

**Step 3** Provide the following credentials to enter a new set of parameters to connect to the database:

 **NOTE**

- You can click **Clear** to clear all fields in the **New Database Connection** dialog box.
- Use shortcut key (Ctrl+V) to paste data in Connection window. Data Studio does not support right-click options for all dialog boxes.

Field Name	Description	Example
Database Type	Select the database type.	GaussDB 200DWS
Connection Name	Provide a connection name.	My_Connection_DB

Field Name	Description	Example
Host	<p>Provide the IP address (IPv4) or server domain name.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>● If domain name length is greater than 25 characters, then the complete domain name will not be displayed. <b>Example:</b> <i>test1(db.dws...com:25xxx)</i></li> <li>● Hovering over the connection name once the connection is established will show the server IP and version.</li> <li>● Entry made in this field will be validated for IP address if it has format of digits with three separators (.). Any entry not meeting this validation will be considered as domain name.</li> <li>● A typical domain name: <ul style="list-style-type: none"> <li>- Starts with a letter.</li> <li>- Allows letters, digits, hyphens (-), and period (.). All other special characters are not allowed.</li> <li>- Does not allow space/tabs.</li> <li>- Length cannot exceed 253 characters and a maximum of 63 characters is allowed in between periods.</li> <li>- Domain name connection login is not supported by GaussDB 100 database.</li> </ul> </li> </ul>	<p>db.dws.mycloud.com</p> <p>10.xx.xx.xx</p>
Host Port	Provide the port address.	25001
Database Name	Provide the database name. This field is not required for GaussDB 100 database.	postgres
User Name	Provide the user name to connect to the selected database.	-

Field Name	Description	Example
Password	Provide the password to connect to the database. The password text is masked.	-

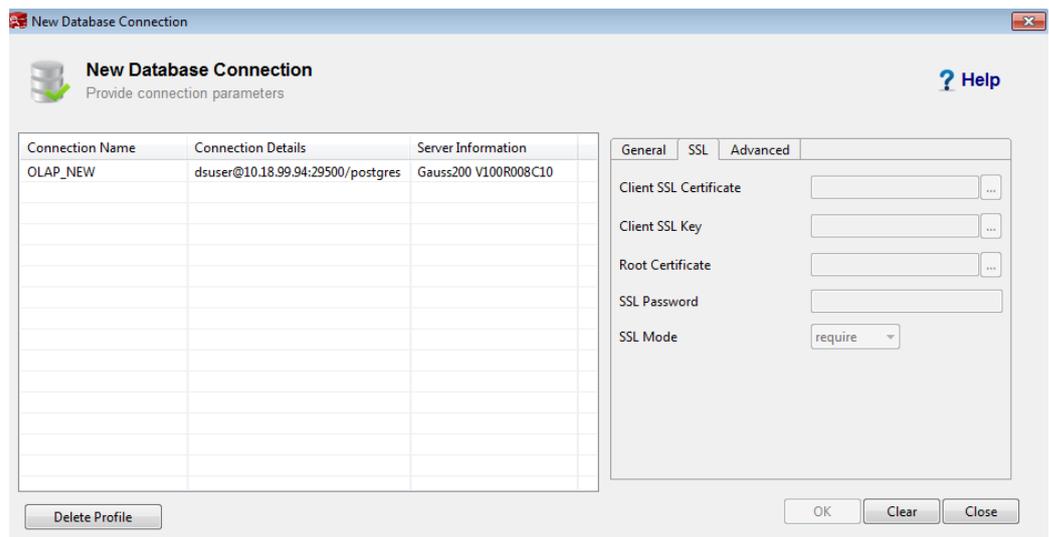
- Select an option from the **Save Password** drop-down list. The options available are:
  - **Permanently**: Saves the password even after exiting Data Studio. While establishing the connection for the first time this option will not be available. Refer to the [Save Password Permanently](#) section for information to hide/view this drop-down option.
  - **Current Session Only**: Saves the password only for the current session.
  - **Do Not Save**: Does not save the password. If this option is selected, Data Studio will prompt for the password for certain operations like:
    - [6.3.1 Creating a Database](#)
    - [6.3.5 Renaming a Database](#)
    - [6.5.4.2 Debugging a PL/SQL Function](#)
    - [6.12.10 Working with the SQL Terminals](#)
- **Enable SSL** check box is selected by default.

**Step 4** Follow the steps below to enable SSL:

**NOTE**

SSL connectivity is not supported by GaussDB 100 database.

1. Select the **Enable SSL** option.
2. Click the **SSL** tab.



3. Provide the following information. The following files are required for secured connection. Refer to [10.7 SSL Certificates](#) section.
  - To select the **Client SSL Certificate**, click and select the Client SSL Certificate.

- To select the **Client SSL Key**, click  and select the Client SSL key.
- To select the **Root Certificate**, click  and select the Root Certificate.
- Select the SSL Mode from **SSL Mode** drop-down. Refer to table below for description of different SSL modes.

 **NOTE**

- If the **SSL Mode** is selected as verify-ca or verify-full, then it is mandatory to select the Root Certificate.
- GaussDB 200 V100R002C80 requires selecting postgres.cert as Client SSL Certificate, postgres.key as Client SSL key, and root.crt as Root Certificate.

SSL Mode	Description
require	Selecting require will not check the validity of the certificates since a non-validating SSL factory is used.
verify-ca	Selecting verify-ca checks if the ca is correct using a validating SSL factory.
verify-full	Selecting verify-full checks if the ca and host is correct using a validating SSL factory.

 **NOTE**

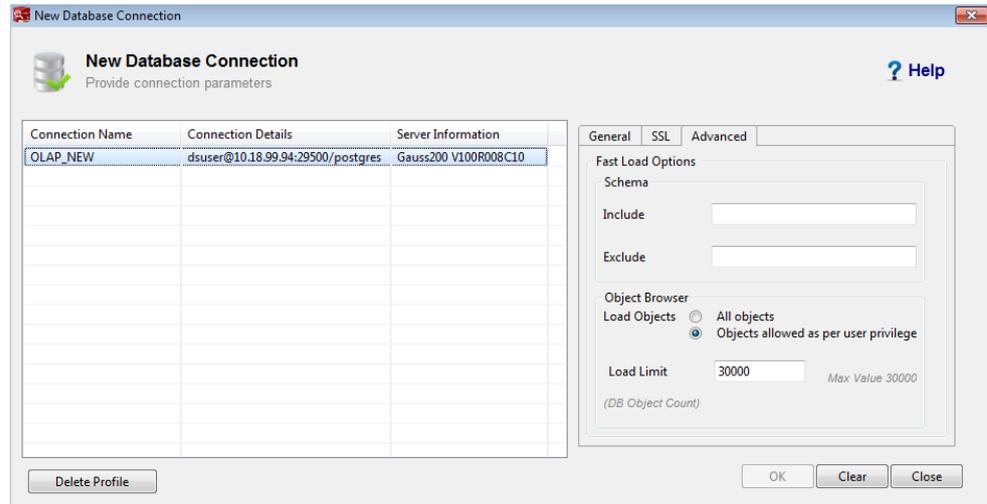
- Selecting **Client SSL Certificate** and **Client SSL Key** ensures secured connection for export of DDL and data using Data Studio.
- Selecting invalid file for Client SSL Certificate and/or Client SSL Key will result in export failure. Refer to [Troubleshooting](#).
- If you deselect **Enable SSL** check box and proceed, then **Connection Security Alert** dialog box is displayed. Refer to [Security Disclaimer](#) for information to display this security alert or not.
  - Continue** - Clicking Continue proceeds with unsecured connection.
  - Cancel** - Clicking Cancel proceeds to enable SSL.
  - Do not show again** - Checking this field hides the **Connection Security Alert** dialog box for subsequent connections for current logged instance of Data Studio.
- Refer to server manual for detailed information.

**Step 5** Follow the steps below to set the **Fast Load Options**:

 **NOTE**

Fast Load feature is not supported by GaussDB 100 database.

1. Click the **Advanced** tab.



2. Enter the schema names using comma separator to load on priority while establishing a connection in the **Include** field.
3. Enter the schema names using comma separator to avoid loading on priority while establishing a connection in the **Exclude** field.
4. Select an option from the **Load Objects** options. The options available are:
  - **All Objects** - Loads all objects.
  - **Objects allowed as per user privilege** - Loads only objects that the user has access. Refer to **Minimum Privileges Requirement** table for the minimum access required for objects to be listed in Object Browser.

**NOTE**

The default value is **Objects allowed as per user privilege**.

5. Enter the load limit in **Load Limit** field. The maximum value allowed is 30000. This is the database object count.

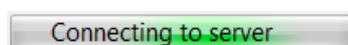
**NOTE**

- If the number of object type (tables, view..) of the schema mentioned in the **Include** field is greater than the value entered in the **Load Limit** field, then the only the parent objects for a schema will be loaded. This implies that child objects like columns, constraints, indexes, functions with more than three parameters, and so on will not be loaded.
- Schema names provided in the Include and Exclude lists are validated.
- If you do not have access to the schema name entered in the **Include** field, then an appropriate error message is displayed for that schema during connection.
- If you do not have access to the schema name entered in the **Exclude** field, then the schema will not be loaded in **Object Browser** after connection is established.

**Step 6** Click **OK** to establish the connection successfully.

The status bar displays the status of the completed operation.

While Data Studio is connecting to the database, the following status bar shows the status:



Once the connection is established, all schema objects will be displayed in the **Object Browser** pane.

 **NOTE**

- Data Studio allows you to login even if the password has expired with a message informing that some operations may not work as expected. Refer to [Password Expiry](#) for information to change this behavior.
- Refer to [Cancel Connection](#) section to cancel the connection.
- Postgres specific schemas are not displayed in the Object Browser.

----End

## Cancel Connection

Follow the steps below to cancel the connection operation:

**Step 1** Click **Cancel**.

The **Cancel Connection** dialog box is displayed.

**Step 2** Click **Yes**.

A message confirmation dialog box is displayed.

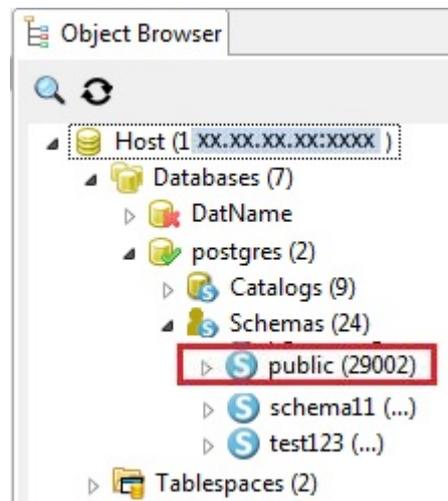
**Step 3** Click **OK**.

----End

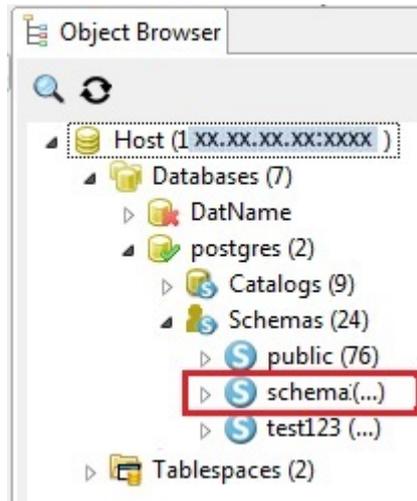
## Lazy Loading

Lazy loading feature delays the loading of objects until required.

When you connect to a database only child objects of schema saved under **search\_path** will be loaded as shown below:



Unloaded schemas are represented as "schema (...)".



To load child objects expand the schema. During expansion of schema, the objects under the schema will show as loading:



**NOTE**

If you try to load an unloaded object while loading is in progress for another object, a pop-up message is displayed informing you that another loading is in progress. The  icon next to the unloaded object disappears. Refresh at the object or database level to display this icon again for loading. Expand schema to load and view the child objects. The Object Browser can load child objects of only one schema at a time.

If **search\_path** is modified after establishing connection, then the changes will be reflected only after reconnecting the database. Auto-suggest works on schema names, table names, views, and table name aliases for all schema objects that you have access.

A maximum of 50,000 objects will be loaded in the **Object Browser** pane within 1 minute.

Database connection timeout is set as 3 minutes (or 180 seconds) by default, beyond which connection time out error is displayed.

You can set the *loginTimeout* value in *Data Studio.ini* file. The file is present in the *Data Studio\Data Studio.ini* path.

**NOTE**

When user log in to the Data Studio, *pg\_catalog* is loaded automatically

## 6.2.3 Renaming a Connection

Follow the steps below to rename a database connection:

- Step 1** In the **Object Browser** pane, right-click the selected connection name and select **Rename Connection**.

A **Rename Connection** dialog box is displayed prompting you to provide the new name for the connection.

- Step 2** Enter the new connection name. Select the **OK** to rename the connection.

The status bar displays the status of the completed operation.

 **NOTE**

The new connection name must be unique else the rename operation will fail.

---End

## 6.2.4 Editing a Connection

Follow the steps below to edit the database connection properties:

- Step 1** In the **Object Browser** pane, right-click the selected connection name and select **Edit Connection**.

Editing an active connection will require closing the connection and then reopening the connection with the new properties. A warning message about connections being reset is shown.

The **Edit Connection** dialog box is displayed.

- Step 2** Click **OK** to proceed or **Cancel** to exit the operation.

 **NOTE**

The **Connection Name** field cannot be modified.

- Step 3** Edit the connection parameters. Connection parameters are explained in [6.2.2 Adding a Connection](#).

- Step 4** Click **OK** to save the updated connection information.

 **NOTE**

- You can click **Clear** to clear all fields in the **Edit Database Connection** dialog box.
- If you click **OK** without modifying any of the connection parameters, no changes saved dialog message is displayed. Until a connection parameter is changed the dialog message is displayed.
- Data Studio allows you to login even if the password has expired with a message informing that some operations may not work as expected. Refer to [Password Expiry](#) for information to change this behavior.
- Refer to [Cancel Connection](#) section to cancel the connection.

If SSL is not enabled, then a **Connection Security Alert** dialog box is displayed.

- Step 5** Click **Continue** to proceed with unsecured connection or click **Cancel** to return to the **Edit Connection** dialog to enable SSL.

 **NOTE**

**Do not show again** option is used to hide the **Connection Security Alert** dialog box for subsequent connections.

The **Remove Server Confirmation** is displayed to confirm closing databases for the edited connection.

- Step 6** Click **Yes** to proceed to updating the connection information and reconnecting the connection with the updated parameters.

The status bar displays the status of the completed operation.

----End

## 6.2.5 Removing a Connection

Follow the steps below to remove an existing database connection:

- Step 1** Right-click the selected connection name and select **Remove Connection**.

A confirmation dialog box is displayed to remove the connection.

- Step 2** Click **Yes** to remove the server connection.

The status bar displays the status of the completed operation.

This action will remove the connection from the **Object Browser**. Any unsaved data will be lost.

----End

## 6.2.6 Viewing Connection Properties

Follow the steps below to view the properties of a connection:

- Step 1** Right-click the selected connection and select **Properties**.

The status bar displays the status of the completed operation.

Properties of the selected connection is displayed.

### NOTE

If the property of a connection is modified for the connection that is already opened, then open the properties of the connection again to view the updated information on the same opened window.

----End

## 6.2.7 Refreshing a Database Connection

Follow the steps below to refresh the database connection:

- Step 1** In the **Object Browser** pane, right-click the selected connection name and select **Refresh** or press **F5**.

The status bar displays the status of the completed operation.

----End

The time taken to refresh the database depends on the number of objects present in the database. For a large database, it is recommended to perform this operation only if required.

- If you right-click the connection name and select **Refresh**, the connection profile is refreshed. During refresh, the connection will be updated with the latest copy from the server.
- If you right-click the Schema  and select **Refresh**, all functions/procedures and tables under the schema are refreshed. During refresh, all functions/procedures and tables are updated with the latest copy from the server.  
If any stored function/procedure is deleted from the database before the refresh operation, then it will be removed from the **Object Browser** only after you perform the refresh operation.
- If you right-click a specific function/procedure and select **Refresh**, the specific function/procedure is refreshed. During refresh, the specific function/procedure is updated with the latest copy from the server.
- If you refresh at database level or connection profile level, then all the child objects of schema in **search\_path** along with the schema already expanded by the user will be loaded.
- If you re-connect to the Database, then only schema objects saved under **search\_path** will be loaded. Previously expanded objects will not be loaded.
- Database and multiple objects under a database cannot be refreshed simultaneously.

## Exporting/Importing Connection Details

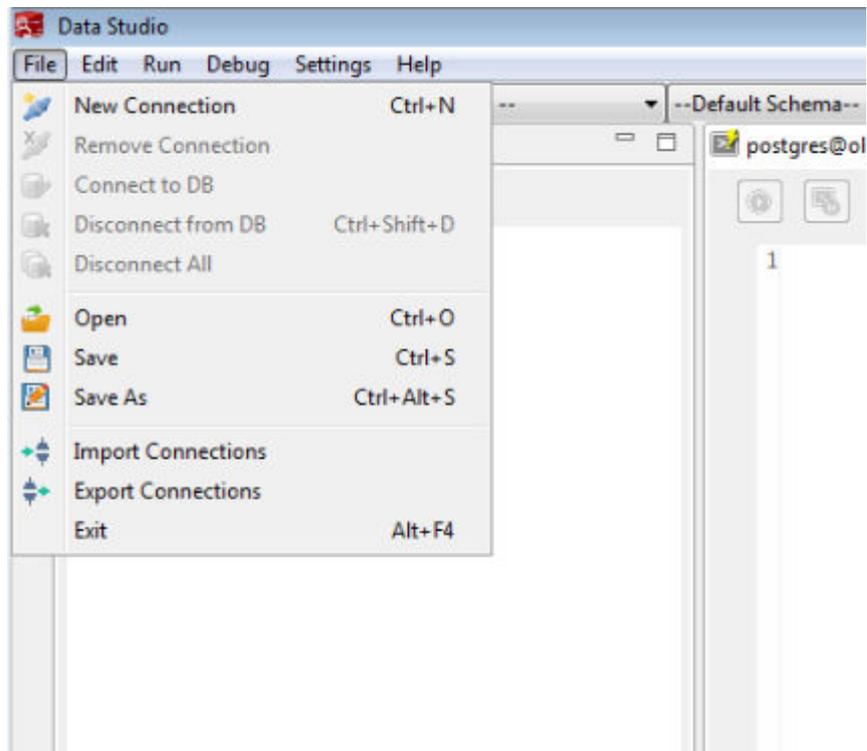
Data Studio provides the option to export/import connection details from the connection dialog for future reference.

Following fields are exported:

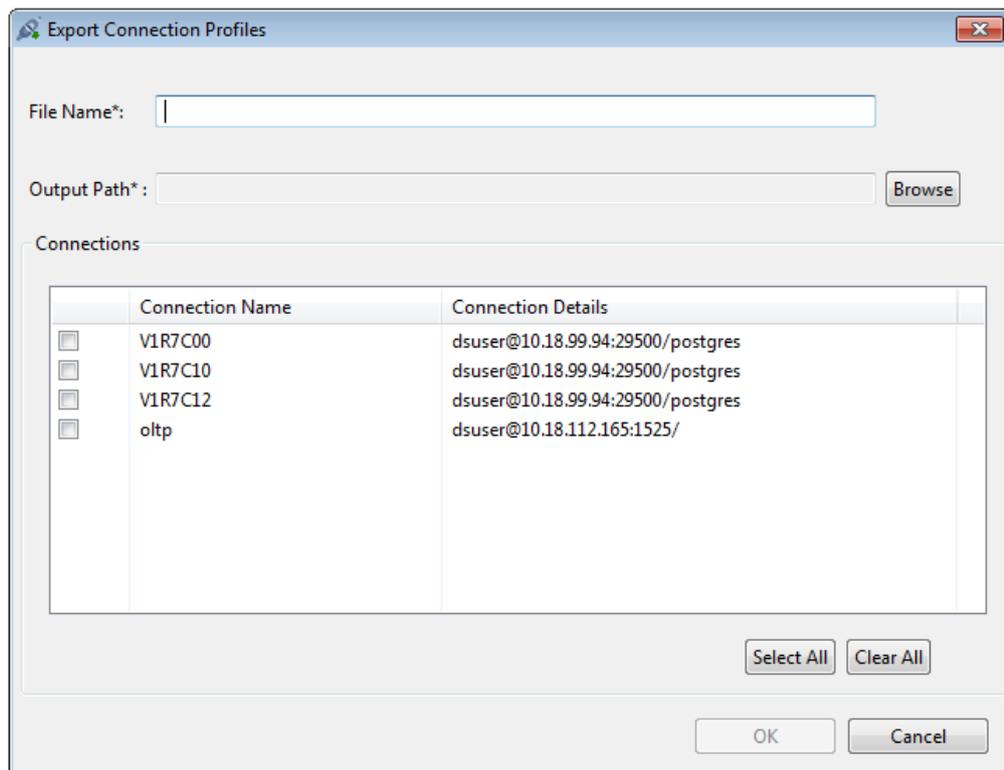
- SSL Mode
- Connection name
- Server IP
- Server Port
- Database Name
- Username
- clSSLCertificatePath
- clSSLKeyPath
- profileId
- rootCertFilePathText
- connctionDriverName
- schemaExclusionList
- schemaInclusionList
- loadLimit
- privilegeBasedObAccess
- databaseVersion
- savePrdOption
- dbType
- version

Follow the steps to access the feature:

**Step 1** Click **File** on Menu Bar. Following window is displayed:

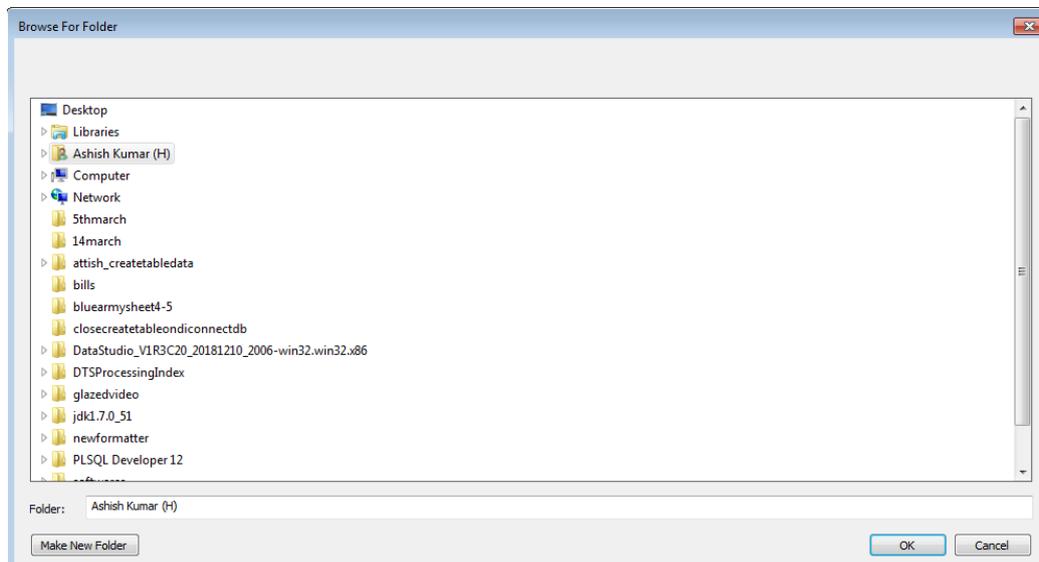


**Step 2** Select **Export Connections** to export the connection profiles. **Export Connection Profiles** window is displayed to the user to select the connections which needs to be exported.

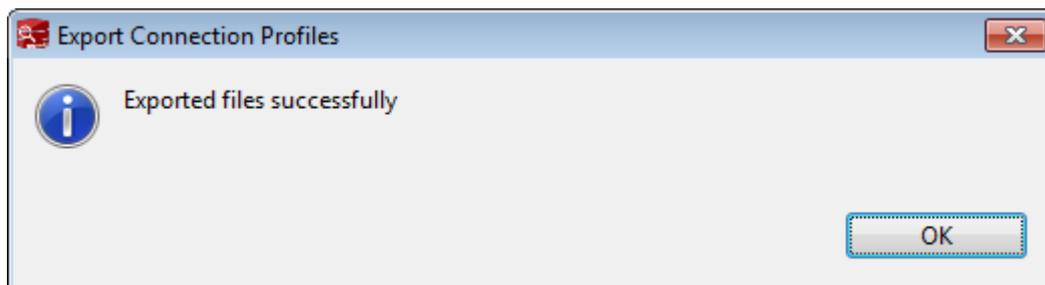


Select the connections you want to export and enter a file name where the exported connections will be saved. Click **OK**

Select the location where you want to save the file and Click **OK**

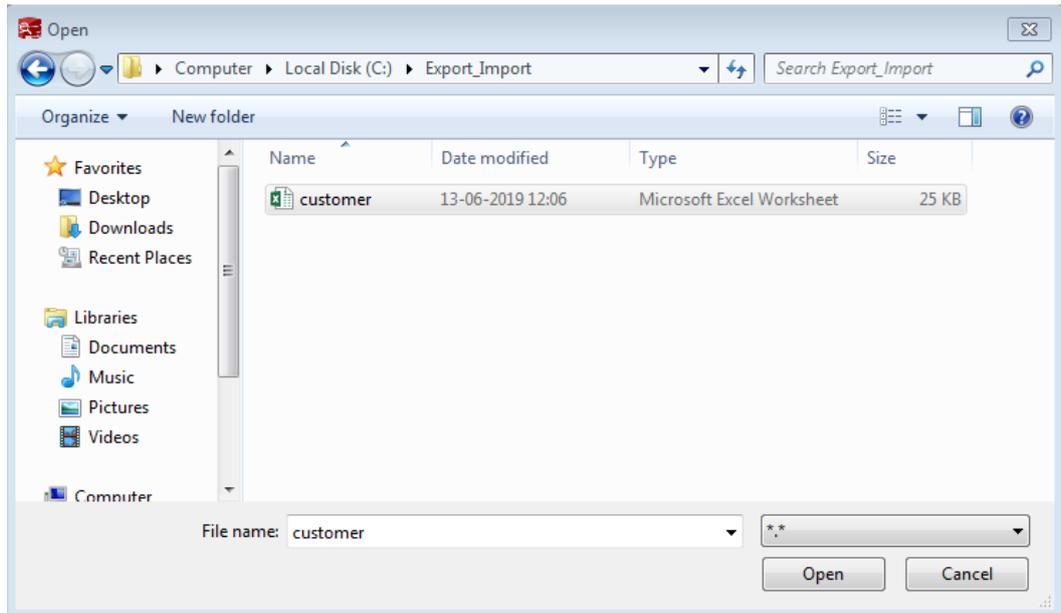


A dialog box is displayed once the connections are exported successfully.

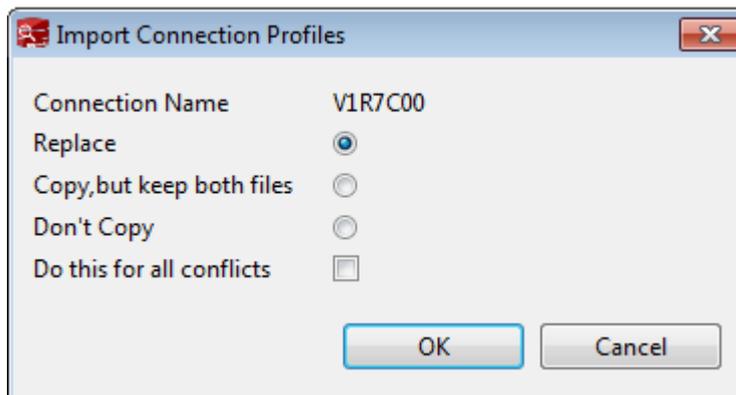


**Step 3** To import the connection profiles select **Import Connections**.

**Step 4** Select the file you want to import and click **Open**.



If there is any match between the connections being imported and the existing connections, a dialog box is displayed .



- **Replace** - The imported connection profile will be replaced with the existing one.
- **Copy, but keep both files** - The imported connection profile will be renamed.
- **Don't Copy** - The existing connection profile will remain as it was.
- **Do this for all conflicts** - Same action will be repeated for all the matches.

Click any of the given options as per the scenario and click **OK**

----End

 **NOTE**

Password and SSL password field will not be exported.

## 6.3 Databases

## 6.3.1 Creating a Database

A relational database is a database that has a set of tables which is manipulated in accordance with the relational model of data. It contains a set of data objects used to store, manage, and access data. Examples of such data objects are tables, views, indexes, functions and so on.

Follow the steps below to create a database:

- Step 1** In the **Object Browser** pane, right-click the selected **Databases** group and select **Create Database**.

 **NOTE**

This operation can be performed only when there is at least one active database.

A **Create Database** dialog box is displayed prompting you to provide the necessary information to create the database.

- Step 2** Enter the database name. Refer to server manual for database naming rules.
- Step 3** Select the required type of encoding character set from the **Database Encoding** drop-down list.

The database supports **UTF-8**, **GBK**, **SQL\_ASCII**, and **LATIN1** types of encoding character sets. Creating the database with other encoding character sets may result in erroneous operations.

- Step 4** Select the **Connect to the DB** check box and click **OK**.

The status bar displays the status of the completed operation.

You can view the created database in the **Object Browser**. The system related schema present in the server is automatically added to the new database.

 **NOTE**

Data Studio allows you to login even if the password has expired with a message informing that some operations may not work as expected when no other database is connected in that connection profile. Refer to [Password Expiry](#) for information to change this behavior.

----End

## Cancel Connection

Follow the steps below to cancel the connection operation:

- Step 1** Double-click the status bar to open the **Progress View** tab.
- Step 2** In the **Progress View** tab, click .
- Step 3** In the **Cancel Operation** dialog box, click **Yes**.

The status bar displays the status of the cancelled operation.

----End

## 6.3.2 Disconnecting All Databases

You can disconnect all the databases from a connection.

Follow the steps below to disconnect a connection from the database:

**Step 1** In the **Object Browser** pane, right-click the selected **Databases** group and select **Disconnect All**. This will disconnect all the databases under that connection.

 **NOTE**

This operation can be performed only when there is at least one active database.

A confirmation dialog box is displayed to disconnect all databases for the connection.

**Step 2** Click **Yes** to disconnect.

The status bar displays the status of the completed operation.

Data Studio populates all the connection parameters (except password) that were provided during the last successful connection with the database. To reconnect, you need to enter only the password in the connection wizard.

----End

## 6.3.3 Connecting to Database

You can connect to the database.

Follow the steps below to connect a database:

**Step 1** In the **Object Browser** pane, right-click the selected database name and select **Connect to DB**.

 **NOTE**

This operation can be performed only on an inactive database.

The database is connected.

The status bar displays the status of the completed operation.

 **NOTE**

- Data Studio allows you to login even if the password has expired with a message informing that some operations may not work as expected when no other database is connected in that connection profile. Refer to [Password Expiry](#) for information to change this behavior.
- Refer to [Cancel Connection](#) section to cancel the connection to database.

----End

## 6.3.4 Disconnecting Database

You can disconnect the database.

Follow the steps below to disconnect a database:

**Step 1** In the **Object Browser** pane, right-click the selected database name and select **Disconnect from DB**.

 **NOTE**

This operation can be performed only on an active database.

A confirmation dialog box is displayed to disconnect database.

**Step 2** Click **Yes** to disconnect.

The database is disconnected.

The status bar displays the status of the completed operation.

----End

## 6.3.5 Renaming a Database

Follow the steps below to rename a database:

**Step 1** In the **Object Browser** pane, right-click the selected database and select **Rename Database**.

 **NOTE**

This operation can be performed only on an inactive database.

A **Rename Database** dialog box is displayed prompting you to provide the necessary information to rename the database.

**Step 2** Enter the new database name. Select the **Connect to the DB?** check box and click **OK**.

A confirmation dialog box is displayed to rename the database.

**Step 3** Click **OK** to rename the database.

The status bar displays the status of the completed operation.

You can view the renamed database in the **Object Browser**.

 **NOTE**

Refer to [Cancel Connection](#) section to cancel the connection to database.

----End

## 6.3.6 Dropping a Database

Individual or batch drop can be performed on databases. Refer to [6.13.2 Dropping Batch of Objects](#) section for batch drop.

Follow the steps below to drop a database:

**Step 1** In the **Object Browser** pane, right-click the selected database and select **Drop Database**.

 **NOTE**

This operation can be performed only on an inactive database.

A confirmation dialog box is displayed to drop the database.

**Step 2** Click **OK** to drop the database.

A popup message and status bar displays the status of the completed operation.

----End

## 6.3.7 Viewing a Database Properties

Follow the steps below to view the properties of a database:

**Step 1** Right-click the selected database and select **Properties**.

 **NOTE**

This operation can be performed only on an active database.

The status bar displays the status of the completed operation.

The properties of the selected database is displayed.

 **NOTE**

If the property of a database is modified for the database that is already opened, then refresh and open the properties of the database again to view the updated information on the same opened window.

----**End**

## 6.4 Schemas

### 6.4.1 Overview

This section describes working with database schemas. All system schemas are grouped under **Catalogs** and user schemas under **Schemas**.

### 6.4.2 Creating a Schema

In relational database technology, schemas provide a logical classification of objects in the database. Some of the objects that a schema may contain include functions/procedures, tables, sequences, views, and indexes.

Follow the steps below to define a schema:

**Step 1** In the **Object Browser** pane, right-click the selected **Schemas** group and select **Create Schema**.

 **NOTE**

Only refresh can be performed on **Catalogs** group.

**Step 2** Enter the schema name and click **OK**. You can create the schema only if the database connection is active.

You can view the new schema in the **Object Browser** pane.

The status bar displays the status of the completed operation.

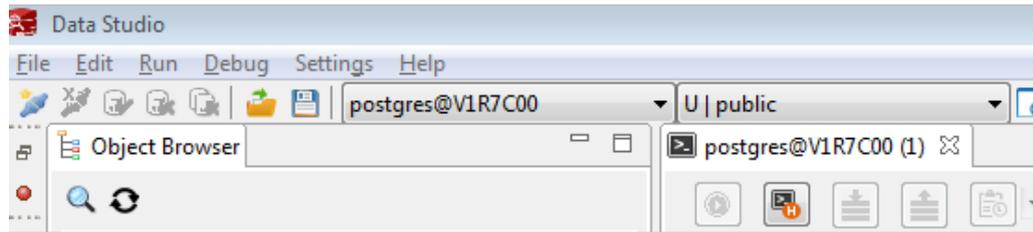
----**End**

You can perform the following actions on a schema:

- Refresh a Schema - To refresh a schema, right-click the selected **Schema Name** and select **Refresh Schema**. All the objects under that schema will be refreshed.
- Rename Schema (Refer to [6.4.5 Renaming a Schema](#) for more details)
- Drop Schema (Refer to [6.4.8 Dropping a Schema](#) for more details)
- Export DDL (Refer to [6.4.3 Exporting Schema DDL](#) for more details)
- Export DDL and Data (Refer to [6.4.4 Exporting Schema DDL and Data](#) for more details)

### Displaying Default Schema

Data studio displays default schema of the user in the toolbar.



When a create query without mentioning the schema name is executed from SQL Terminal, the corresponding objects are created under the default schema of the user.

When a select query is executed in SQL terminal without mentioning the schema name, default schemas are searched to find these objects.

When Data Studio starts, the default schemas are set to <username>, public schemas in same priority.

If another schema is selected in drop down, this selected schema will be set as second priority schema, <username> schema being the first priority.

The selected schema is set as default schema for all active connections of the database (selected in connection profile).

 **NOTE**

This feature is not available for OLTP database.

## 6.4.3 Exporting Schema DDL

Exporting the schema DDL exports the DDLs of functions/procedures, tables, sequences and views of the schema.

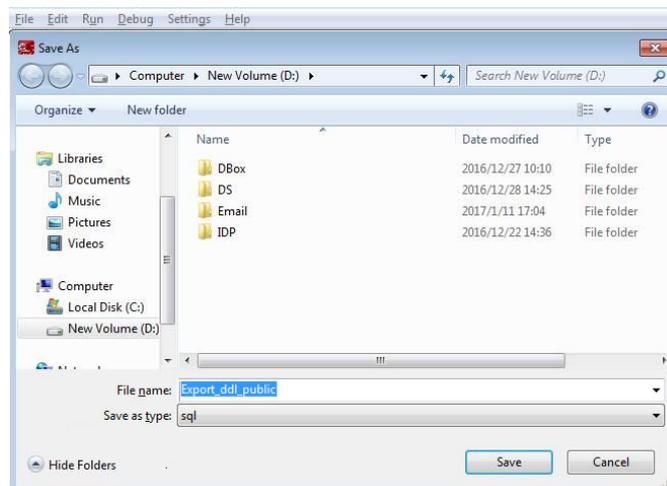
Follow the steps below to export the schema DDL:

**Step 1** In the **Object Browser** pane, right-click the selected schema and select **Export DDL**.

The **Data Studio Security Disclaimer** dialog box is displayed. You can turn off this security disclaimer message. Refer to [Security Disclaimer](#) section for more information.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.



**Step 3** In the **Save As** dialog box, select the location to save the DDL and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click . For more information refer to [Canceling the export table data operation](#).
- The exported file name will not be the same as schema name, if the schema name contains characters which are not supported by Windows.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

 **NOTE**

Multiple objects can be selected to export DDL. Refer to [Batch Export](#) section for list of objects not supported for export DDL operation.

----End

## 6.4.4 Exporting Schema DDL and Data

Exporting the schema DDL and data exports the following:

- DDLs of functions/procedures of the schema.
- DDLs and data of tables of the schema.
- DDLs of views of the schema.
- DDLs of sequences of the schema.

Follow steps below to export the schema DDL and data:

**Step 1** In the **Object Browser** pane, right-click the selected schema and select **Export DDL and Data**.

The **Data Studio Security Disclaimer** dialog box is displayed. You can turn off this security disclaimer message. Refer to [Security Disclaimer](#) section for more information.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** In the **Save As** dialog box, select the location to save the DDL and data and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click . For more information refer to [Canceling the export table data operation](#).
- The exported file name will not be the same as schema name, if the schema name contains characters which are not supported by Windows.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

 **NOTE**

Multiple objects can be selected to export DDL and Data. Refer to [Batch Export](#) section for list of objects not supported for export DDL and Data operation.

---End

## 6.4.5 Renaming a Schema

Follow the steps below to rename a schema:

**Step 1** In the **Object Browser** pane, right-click the selected schema and select **Rename Schema**.

**Step 2** Enter the schema name and click **OK**.

You can view the renamed schema in the **Object Browser**.

The status bar displays the status of the completed operation.

----End

## 6.4.6 Support Sequence DDL

Data Studio provides the option to show sequence DDL or allow users to export sequence DDL. It provides "Show DDL", "Export DDL", "Export DDL and Data"

Follow the steps to access the feature:

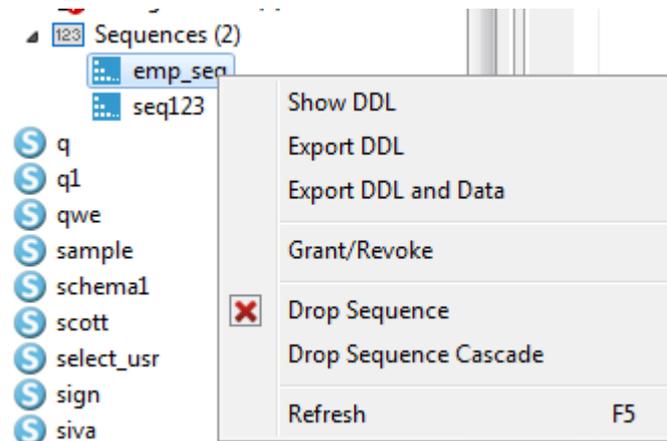
**Step 1** In **Object Browser** right click on any object under **Sequences**. A menu option will appear

**Step 2** Select **Show DDL** option to see the DDL statements.

Or Select **Export DDL** menu option to export DDL statements.

Or Select **Export DDL and Data** menu option to export DDL statements and the select statement.

Refer to the following image:



### NOTE

If user is sequence owner or sysadmin or has select privilege of the sequence , then only the operation can be performed.

----End

## 6.4.7 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click schema group and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Select the objects to grant/revoke privilege from **Object Selection** tab and click **Next**.

**Step 3** Select the role from **Role** drop-down in **Privilege Selection** tab.

**Step 4** Select **Grant/Revoke** in **Privilege Selection** tab.

**Step 5** Select/unselect the required privileges in **Privilege Selection** tab.

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided.

**Step 6** Click **Finish**.

----End

## 6.4.8 Dropping a Schema

Individual or batch drop can be performed on schemas. Refer to [6.13.2 Dropping Batch of Objects](#) section for batch drop.

Follow the steps below to drop a schema:

**Step 1** In the **Object Browser** pane, right-click the selected schema and select **Drop Schema**.

A confirmation dialog to drop the schema is displayed.

**Step 2** Click **OK** to drop the schema. This action will remove the schema from the **Object Browser**.

A popup message and status bar displays the status of the completed operation.

----End

## 6.5 Functions/Procedures

### 6.5.1 Creating Function/Procedure

Follow the steps below to create a function/procedure and SQL function:

**Step 1** In the **Object Browser** pane, right-click **Functions/Procedures** under the particular schema where you want to create the function/procedure and select either **Create PL/SQL Function** or **Create SQL Function** or **Create PL/SQL Procedure** or **Create C Function** based on your requirement.

Data Studio opens a new tab with the selected template.

#### NOTE

Creating SQL function for GaussDB 100 is not supported.

**Step 2** Add the function/procedure by right-clicking in the tab and selecting **Compile**, or choosing **Run > Compile/Execute Statement** from the main menu, or pressing **Ctrl+Enter** to compile the function/procedure.

The **Created function/procedure Successfully** dialog box is displayed, and the new function/procedure is displayed under the **Object Browser**. Click **OK** to close the *NewObject()* tab and add the debug object in the **Object Browser**.

Refer to the [Execute SQL Queries](#) section for information on the reconnect option in case connection is lost during execution.

**Step 3** The \* symbol next to the function/procedure name indicates that the function/procedure is not compiled or added in the **Object Browser**.

You must refresh (using **F5**) the **Object Browser** to view the newly added debug object.

**NOTE**

- A popup message displays the status of the completed operation. The status bar does not display the status of this operation.
- For C functions, debug operation is not supported.

---End

## Supporting Compile Function

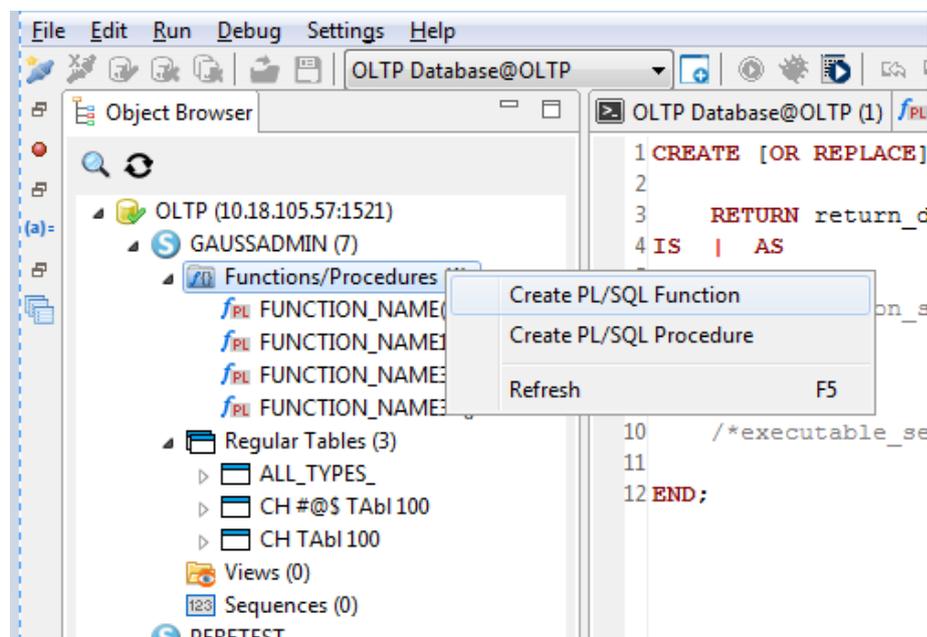
When a new PLSQL Object, either from the template or by editing an existing PLSQL Object, the newly created SQL Object will open in a new tab.

This new function will open in a new tab.

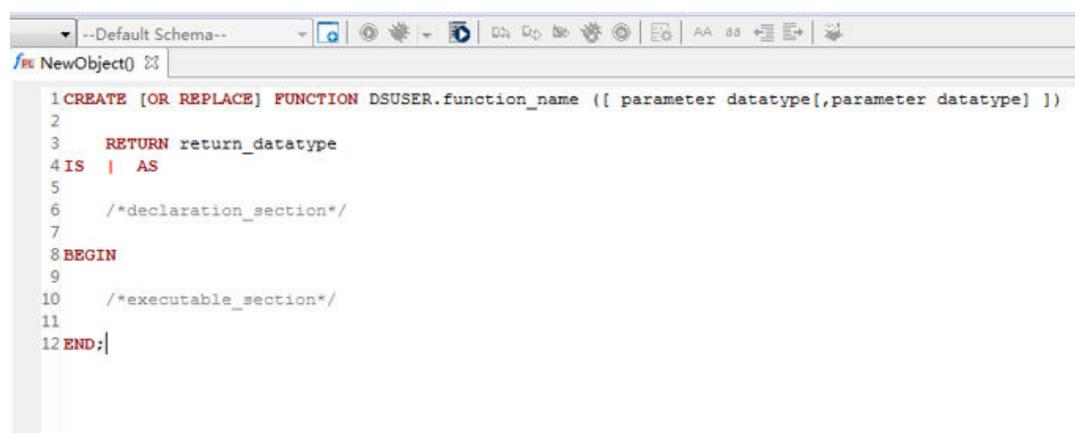
Follow the steps to compile function:

**Step 1** Select **Function/Procedures** on object browser.

**Step 2** Right click on **Function/Procedures** . A menu option is displayed.

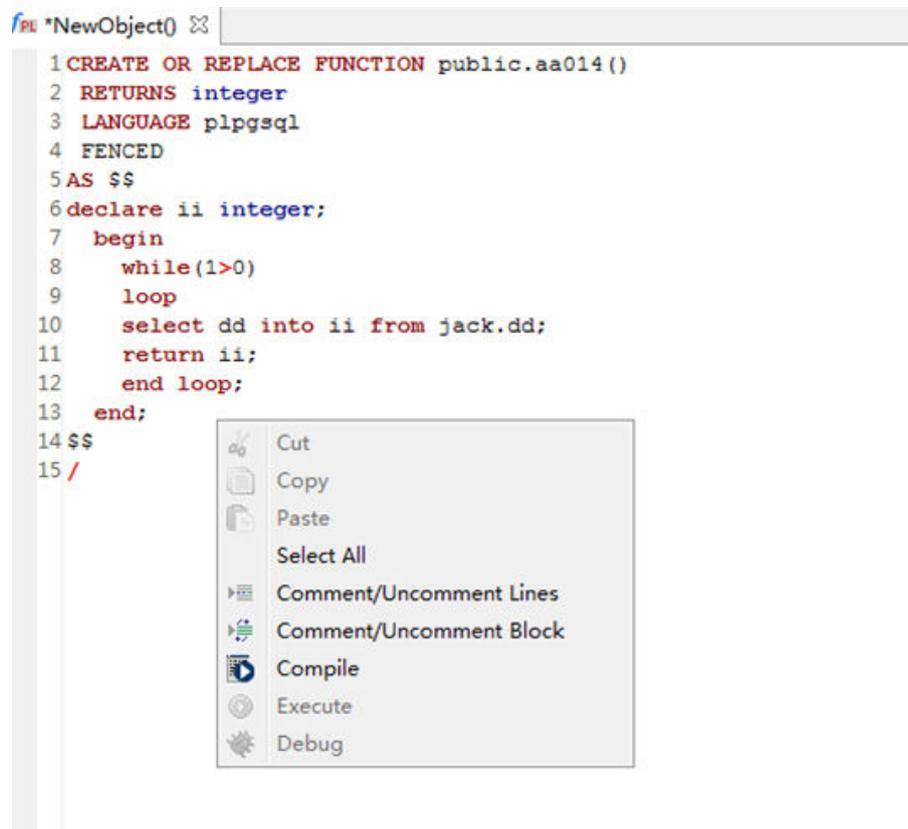


**Step 3** Click **Create PL/SQL Function**. The new function/procedure tab is opened.

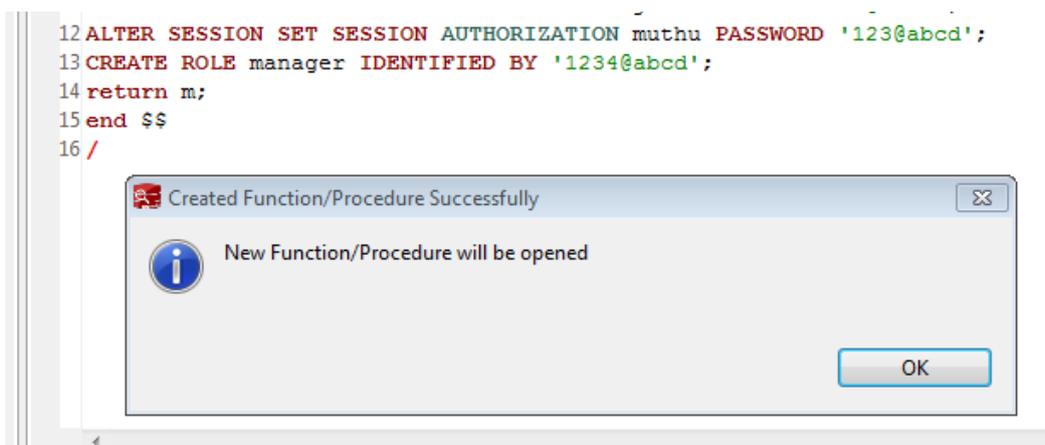


**Step 4** Edit the code.

**Step 5** Right click on the tab. A menu option is displayed.



**Step 6** Click **Compile**. A pop-up message is displayed as following.



This function is displayed in a new tab.

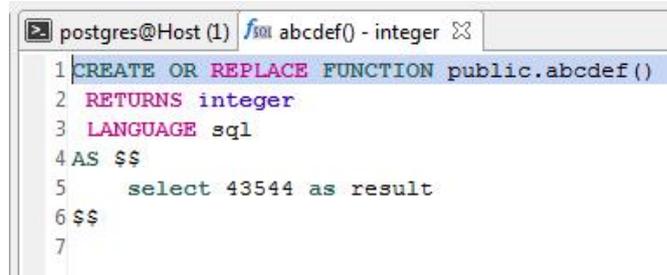
----End

## 6.5.2 Editing a Function/Procedure

Follow the steps below to open and edit the function/procedure or SQL function:

- Step 1** In the **Object Browser** pane, double-click the required function/procedure or SQL function or right-click the function/procedure or SQL function and select **View Source**. You must refresh the **Object Browser** to view the latest DDL.

The function/procedure or SQL function based on your selection is displayed.



```
postgres@Host (1) f301 abcdef() - integer ⌵
1 CREATE OR REPLACE FUNCTION public.abcdef()
2 RETURNS integer
3 LANGUAGE sql
4 AS $$
5     select 43544 as result
6 $$
7
```

Only one function/procedure or SQL function with the same schema, name, and input parameters can be opened in Data Studio.

- Step 2** After editing or updating, compile and execute the PL/SQL program or SQL function. For more details, refer to [6.5.4.7 Executing a Function/Procedure](#).

If you execute the function/procedure or SQL function before compiling, a **Source Code Change** dialog box is displayed.

- Step 3** Click **Yes** to compile and execute the function/procedure.

The **Messages** tab displays the status of the completed operation.

Refer to [Execute SQL Queries](#) section for information on reconnect option in case connection is lost during execution.

- Step 4** After compiling the function/procedure or SQL function, refresh the **Object Browser** (using F5) to view the updated code.

----End

## 6.5.3 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

- Step 1** Right-click functions/procedures group and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

- Step 2** Select the objects to grant/revoke privilege from **Object Selection** tab and click **Next**.

The **Privilege Selection** tab is displayed.

- Step 3** Select the role from **Role** drop-down..

- Step 4** Select **Grant/Revoke**.

- Step 5** Select/unselect the required privileges.

The **SQL Preview** tab displays the SQL query automatically generated for the inputs provided.

- Step 6** Click **Finish**.

----End



This feature is only supported in OLAP, not in OLTP.

## 6.5.4 Working with Functions/Procedures

### 6.5.4.1 Overview

This section provides you with details on working with functions/procedures and SQL functions in Data Studio.



Data Studio supports plpgsql and SQL languages for the operations listed below:

- [6.5.1 Creating Function/Procedure](#)
- [6.5.2 Editing a Function/Procedure](#)
- [6.5.4.4 Exporting a Function/Procedure DDL](#)
- [6.5.4.6 Dropping a Function/Procedure](#)

### 6.5.4.2 Debugging a PL/SQL Function

#### 6.5.4.2.1 Overview

During debugging operation if the connection is lost and the database is still connected in Object Browser, then **Connection Error** dialog box is displayed:

- **Yes** - The connection is reestablished and restarts debug operation.
- **No** - Disconnects database in Object Browser.



SQL language function does not support debugging operations.

#### 6.5.4.2.2 Using Breakpoints

This section contains the following topics:

- [Using the Breakpoints Pane](#)
- [Setting or Adding Breakpoints on a Line](#)
- [Enabling or Disabling a Breakpoint on a Line](#)
- [Removing a Breakpoint on a Line](#)
- [Source Code Change](#)
- [How to debug a PL/SQL program using breakpoints?](#)

A breakpoint is used to suspend the execution of a PL/SQL program at the line where the breakpoint is set. You can use breakpoints to control the execution and debug the function.

- An enabled breakpoint suspends the execution of the PL/SQL program whenever a breakpoint is encountered. When the execution hits the line of breakpoint, the execution will stop and you will be able to carry out other debug operations. Data Studio supports the following breakpoint operations:
  - Setting or Adding breakpoint on a line
  - Enabling or Disabling a breakpoint on a line



 **NOTE**

- Disabling a breakpoint prevents the execution from pausing at the breakpoint, but leaves the definition in place (to enable the breakpoint later).
- Deleting a breakpoint removes it permanently.
- The content of the **Breakpoints** pane can be copied to the clipboard using **Alt+Y**.

## Setting or Adding Breakpoints on a Line

Follow the steps below to set or add breakpoints on a line:

**Step 1** Open the PL/SQL function where you want to add the breakpoint.

**Step 2** In the PL/SQL Viewer, double-click the breakpoint ruler on the left side of the line number column. The added breakpoint is indicated by an enable breakpoint sign [  ] in the PL/SQL Viewer.

 **NOTE**

If the execution of the function does not break or stop during debugging, the breakpoint that is already set will not be validated.

---End

## Enabling or Disabling a Breakpoint on a Line

Once a breakpoint is set, you can temporarily disable it by selecting the corresponding check box in the left-side of the **Breakpoints** pane and clicking  at the top of the **Breakpoints** pane. Disabled breakpoints will be grayed out [  ] in the **PL/SQL Viewer** and **Breakpoints** pane.

To enable a disabled breakpoint, select the corresponding breakpoint (using check box) and click .

## Removing a Breakpoint on a Line

If you no longer require the breakpoint, you can remove it using the same actions used to create it.

In the **PL/SQL Viewer** tab, open the function in which you want to remove the breakpoint.

Double-click the enable breakpoint sign [  ] in the **PL/SQL Viewer** to disable the breakpoint. The breakpoint is removed from the workspace.

You can also enable or disable breakpoints using the **Breakpoints** pane as explained above.

## Source Code Change

During debugging, if the source code is changed after it is fetched from the server and debug is continued, Data Studio displays an error.

It is recommended to refresh the object and perform the debug operation again.

 **NOTE**

If the source code is changed after it is fetched from the server, and if you perform the execution or debug operation with no breakpoint set, then the result of the source code at the server will be displayed by Data Studio. It is always recommended to refresh before performing debug or execute operation.

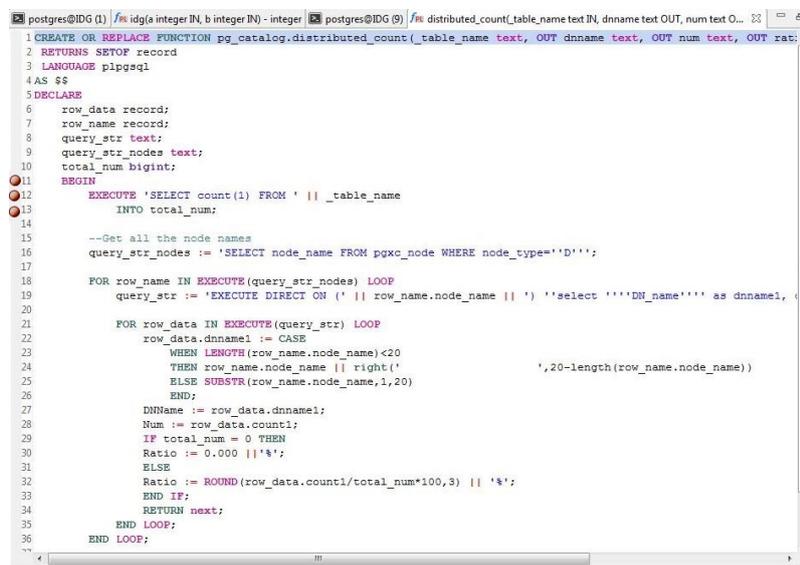
## How to debug a PL/SQL program using breakpoints?

Follow the steps below to debug a PL/SQL program using breakpoints:

**Step 1** Open the PL/SQL program and add a breakpoint at the line where you want to perform debug operation.

For example:

Line 11, 12, 13.



```
1 CREATE OR REPLACE FUNCTION pg_catalog.distributed_count(_table_name text, OUT dname text OUT, num text OUT, OUT rat:
2 RETURNS SETOF record
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6     row_data record;
7     row_name record;
8     query_str text;
9     query_str_nodes text;
10    total_num bigint;
11    BEGIN
12        EXECUTE 'SELECT count(1) FROM ' || _table_name
13        INTO total_num;
14
15    --Get all the node names
16    query_str_nodes := 'SELECT node_name FROM pgxc_node WHERE node_type='D''';
17
18    FOR row_name IN EXECUTE(query_str_nodes) LOOP
19        query_str := 'EXECUTE DIRECT ON (' || row_name.node_name || ') 'select ''DN_name'' as dname1,
20
21        FOR row_data IN EXECUTE(query_str) LOOP
22            row_data.dname1 := CASE
23                WHEN LENGTH(row_name.node_name)<20
24                THEN row_name.node_name || right('
25                ELSE SUBSTR(row_name.node_name,1,20)
26            END;
27            DNName := row_data.dname1;
28            Num := row_data.count1;
29            IF total_num = 0 THEN
30                Ratio := 0.000 || '%';
31            ELSE
32                Ratio := ROUND(row_data.count1/total_num*100,3) || '%';
33            END IF;
34            RETURN next;
35        END LOOP;
36    END LOOP;
```

**Step 2** To start debugging, click  or press **Ctrl+D**, or right-click the selected PL/SQL program in the **Object Browser** and select **Debug**. The **Debug Function/Procedure** dialog box appears prompting for your input.

 **NOTE**

If there is no input parameter, then **Debug Function/Procedure** dialog box will not appear.

**Step 3** Provide your input and click **OK**. For varchar and date datatype, provide the input value in single quotes and for numeric datatype, provide the input value with or without single quotes.

To set NULL as the parameter value, enter *NULL* or *null*.

On clicking the **Debug** button, you will see an arrow  pointing to the line where breakpoint is set. The arrow indicates the line number at which execution will resume from.

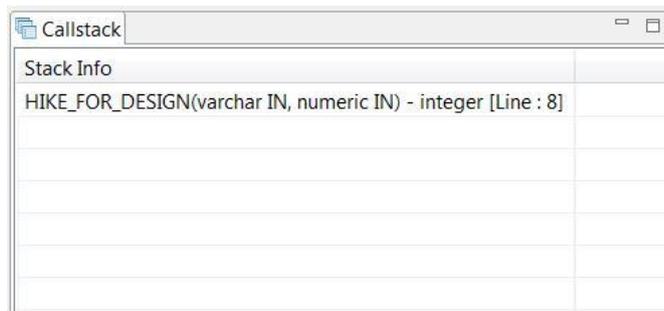
```

1 CREATE OR REPLACE FUNCTION public.func_long_src_code()
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$ DECLARE result INTEGER;
5 ab varchar(27);
6 bc varchar(27);
7     a INTEGER;
8
9 BEGIN
10     a := 10;
11     a := a+2;
12     a := a+2;
13     a := a+2;
14     a := a+2;
15     a := a+2;
16     a := a+2;
17     a := a+2;
18     a := a+2;
19     a := a+2;
20 ab:='abcdefghijklmnopqrstuvwxy';
21 bc:=ab;
22 ab:='abcasddfsdfksdfksdklflkla';
23 a := a+2;
24 a := a+2;
25 a := a+2;
26 a := a+2;

```

You can terminate debugging by clicking  from the toolbar, or pressing **F10**, or select **Terminate Debugging** from the **Debug** menu. After stopping the debug operation, the execution of the function will not break for any breakpoint and the execution will proceed normally.

The **Callstack** and **Variables** panes are populated.



The **Variables** pane shows the current value of variables. Mouse over the variable in the function/procedure also shows the current value of variables.

Variable	Datatype	Value
__gsdb_cursor_attri_found__	bool	
__gsdb_cursor_attri_isopen__	bool	f
__gsdb_cursor_attri_notfound__	bool	
__gsdb_cursor_attri_rowcount__	integer	
__gsdb_sql_cursor_attri_found__	bool	
__gsdb_sql_cursor_attri_isopen__	bool	f
__gsdb_sql_cursor_attri_notfound__	bool	
__gsdb_sql_cursor_attri_rowcount__	integer	
emp_age	integer	30
emp_designation	varchar	Developer
emp_id	integer	100
emp_name	varchar	Emp1
emp_sal	numeric	100000.224

You can step through the code using Step Into, Step Out or Step Over. For more details, refer to [6.5.4.2.3 Controlling Execution](#) .

**Step 4** Click **Continue**  to continue the execution till the next breakpoint (if any). The result of the executed PL/SQL program is displayed in the **Result** tab and the **Callstack** and **Variables** panes are cleared. You can copy the content of the **Result** tab, by clicking  .

To remove the breakpoint, do the following:

- Double-click again on the breakpoint to remove it from the **PL/SQL Viewer**.
- Select the breakpoint in the breakpoint check box and click  in the **Breakpoints** pane.

----End

## Support Rearrangement Of Variable Window

This feature enables the Variable Window and columns to be rearranged. You are able to arrange Variable Window to the following places:

- Beside SQL Assistant Tab
- Beside SQL Terminal Tab
- Beside Object Browser Tab
- Beside Resultset Tab
- Beside Breakpoints Tab
- Beside Callstack Tab
- Below Object Browser Tab

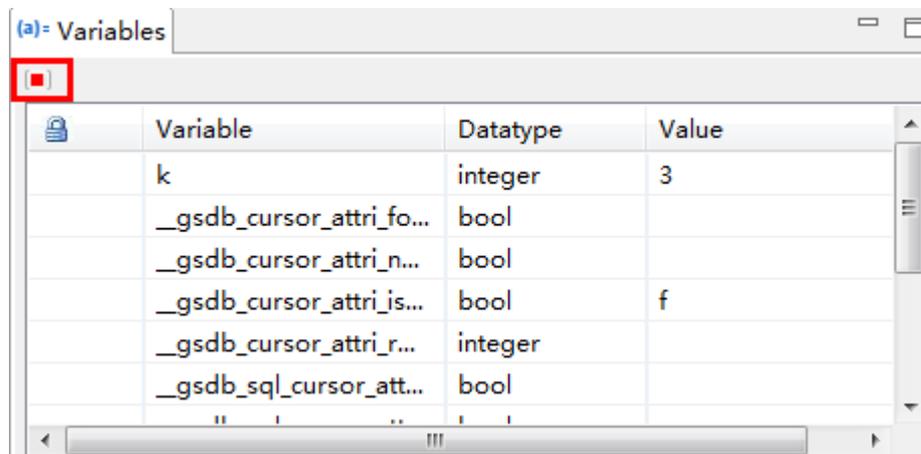
### NOTE

When debugging is finished, the variable window will be minimized even if the variable window is rearranged while debugging. If variable window is rearranged as a Terminal Tab or Result Tab, on completion of debugging, the tab should be minimized manually. The position of variable window is maintained after it is rearranged.

## Enable/Disable System Variables

System Variables are displayed by default. You can disable the system variables whenever required.

**Step 1** Click the red button under Variables to disable System variables



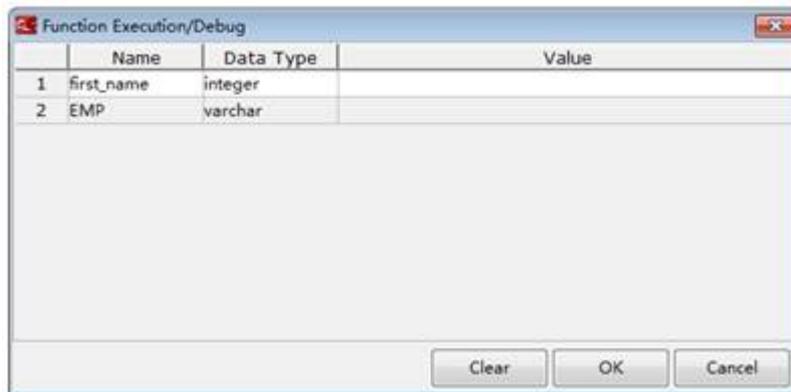
The button is ON state by default.

----End

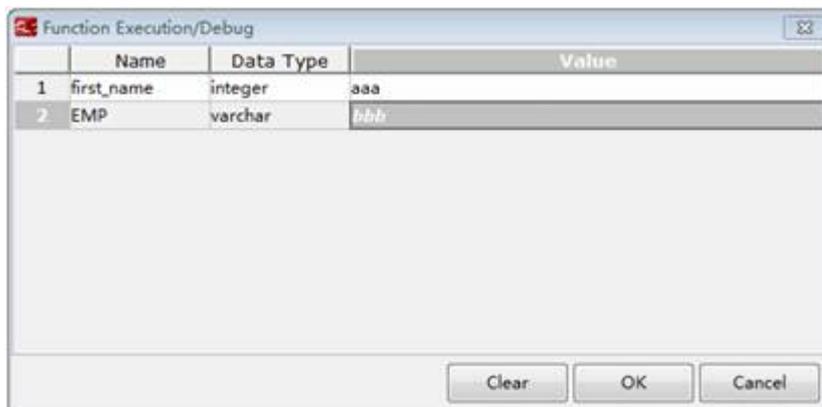
## Displaying Cached Parameter

During debug/execute a PL/SQL function or procedure, same parameter values are used for next time usage.

While executing PL/SQL object, following window is displayed.



For the first time, parameter values are empty. Enter the value as required.



and click OK. The parameter values will be cached. Next time during the query execution/ debug same parameter values will be displayed.

**NOTE**

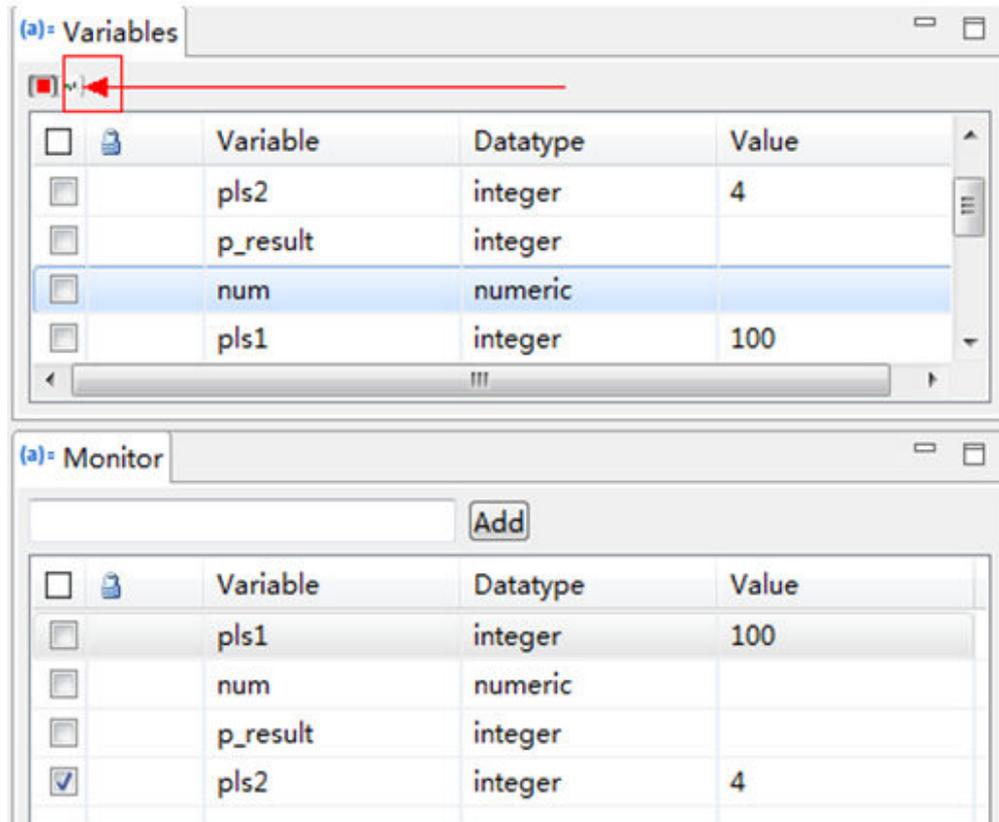
Once the specific connection is removed, all the parameter values in cache are cleared.

## Displaying Variable in Monitor Window

Data Studio displays the variables which are being monitored in the Monitor Window while debugging.

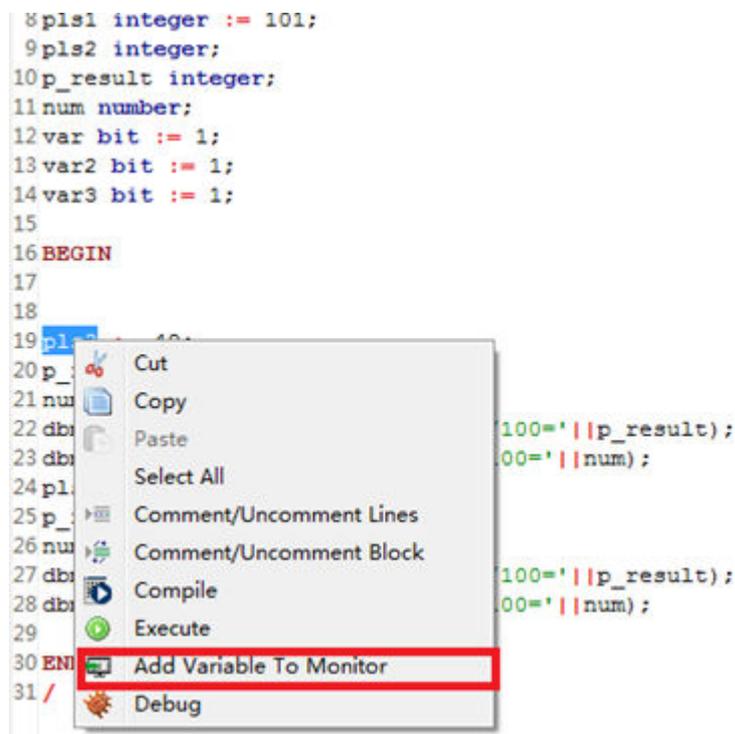
In the Monitor Window variables must be added in following ways:

- Adding selected variables from the Variable Window and right click .
- Selecting the variables from variable window and add by clicking the button in the variable window toolbar.



If value is changed in the variable window, the same would reflect in the monitor window if the variable is monitored and vice versa.

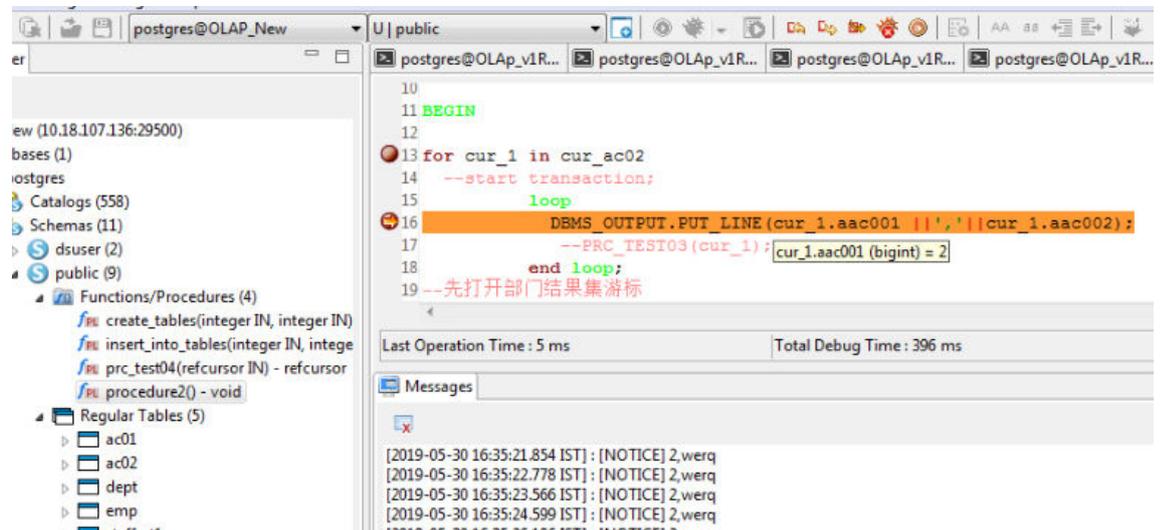
- Adding the variable by right click from the editor while debugging Function/Procedure.



The Monitor window can be dragged anywhere in the Data Studio window.

## Displaying Cursor Information For Variable During Debugging

In Data Studio, variable information is displayed if the cursor is hovered over that variable during debugging of PL/SQL function.



## Supporting Rollback/Commit During Debugging

Data Studio provides the option to commit/rollback the PL/SQL query execution result after debugging is finished.

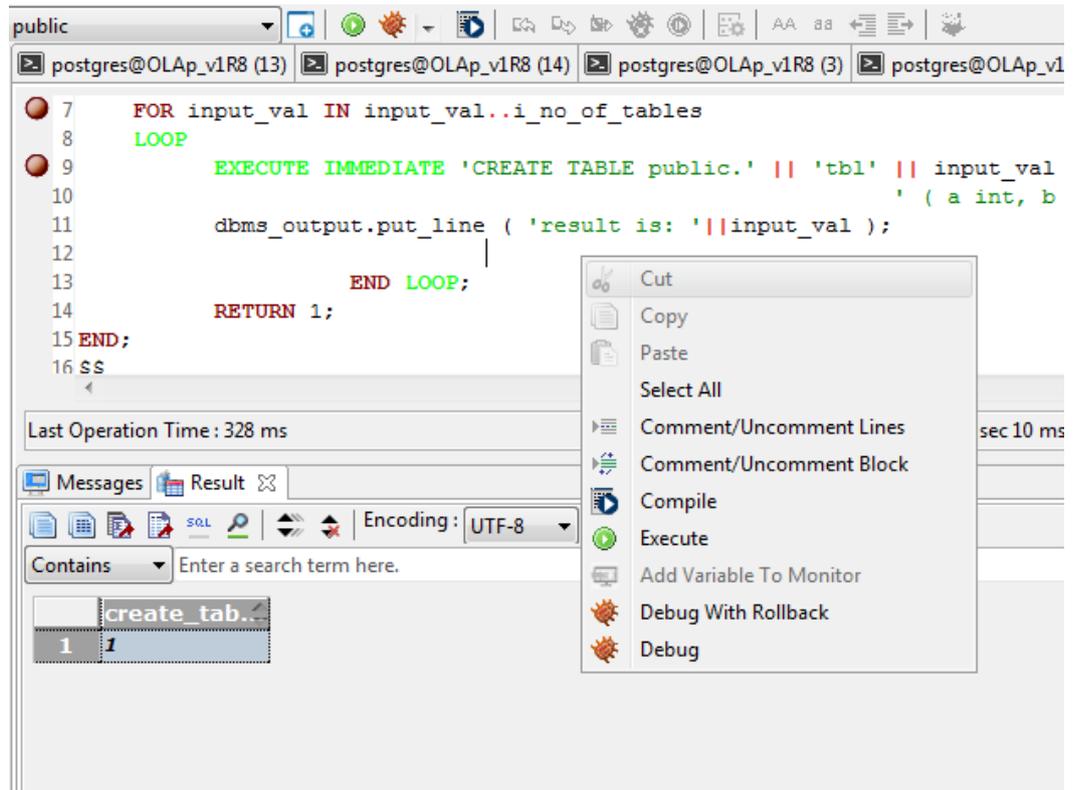
- If **Debug With Rollback** option is enabled, then after PL/SQL execution result after debugging is not saved in the database.
- If **Debug With Rollback** option is disabled, then after PL/SQL execution result after debugging is committed in the database.

Follow the steps to enable the rollback feature the feature:

**Step 1** Check the **Debug With Rollback** box to enable the rollback feature during PL/SQL debug

Or

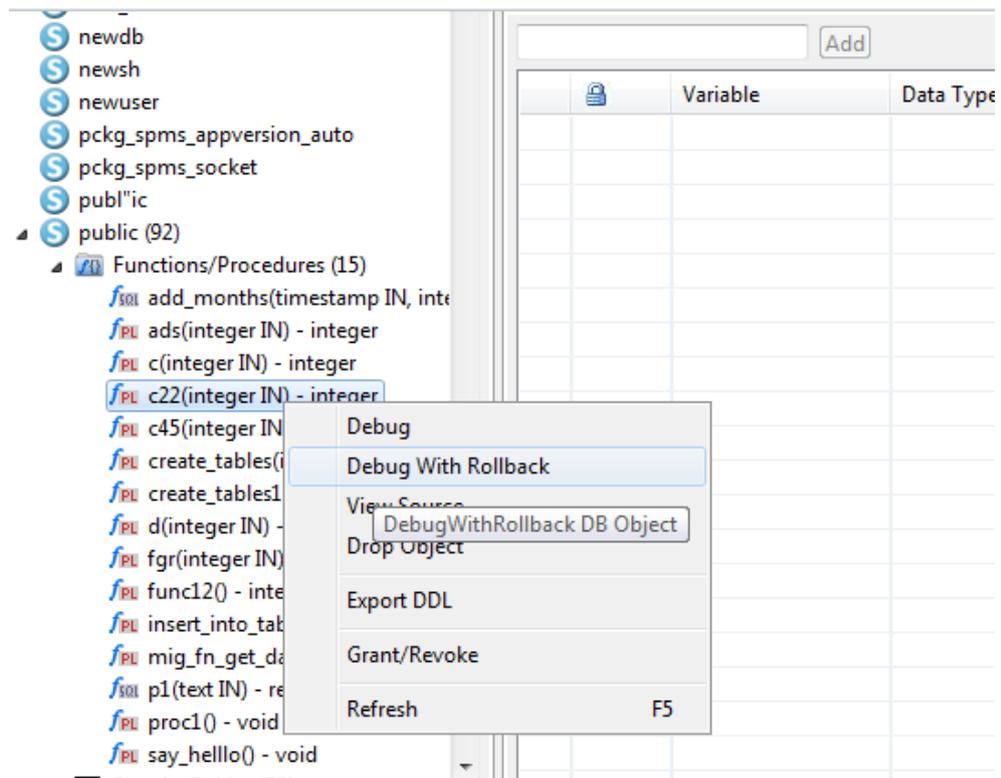
Right click on SQL Terminal window where is PL/SQL function is executed.



Select **Debug With Rollback** option to enable rollback feature after debug is finished.

Or

Right Click on any PL/SQL function under **Functions/Procedure** in Object Browser.



----End

### 6.5.4.2.3 Controlling Execution

This section contains the following topics:

- [Starting the Debug Process](#)
- [Stepping through a PL/SQL Function](#)
- [Continuing the Debug Execution](#)
- [Viewing Callstack](#)

## Starting the Debug Process

Select the function that you want to debug in the **Object Browser** pane. Start debugging by clicking  button on the toolbar (or any other method as mentioned in the earlier sections). If no breakpoint is set, or the set breakpoint is invalid, debug operation will not halt at any statement and simply execute the object and display the results (if any).

## Stepping through a PL/SQL Function

You can step through the debugging execution using the debug step commands from the toolbar. Step controls are used to step through the execution of the program line by line. If a breakpoint is encountered while performing a step operation, the execution will suspend at the breakpoint and the step operation is ended.

Stepping is the process of running one statement at a time. After stepping through a statement, you can see its effect in the other debugger tabs.

### NOTE

A maximum of 100 **PL/SQL Viewer** tabs can be displayed at a time. If a new tab beyond 100 is opened, the tab of the calling function is closed. For example, if 100 tabs are already opened and if one of the debug object calls a new debug object (other than already opened 100 tabs), then Data Studio will close the calling function, and open the new debug object.

## Step Into

To step through code one statement at a time, select **Step Into** from the **Debug** menu, or press  button, or press **F7**.

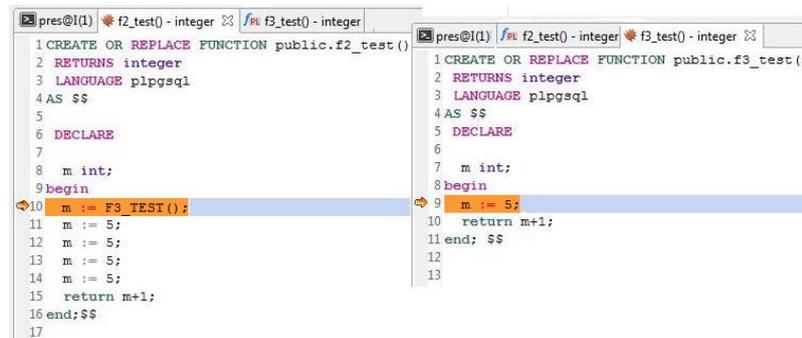
When stepping into a function, Data Studio executes the current statement and then enters the break mode. The debug position will be indicated by an arrow  on the left ruler pane. If the executed statement calls another function, Data Studio will step into that function. Once you have stepped through all the statements in that function, Data Studio will jump back to the next statement of the function it was called from.

To go into the next statement, press the **Step Into (F7)** button again. If you press the **Continue** button, PL/SQL code execution will continue as normal.

For example:

In the following example, when you step into *Line 10*, you will move to line 'm := F3\_TEST();', that is, *Line 9* in *f3\_test()*. You can step through all the statements in *f3\_test()* by stepping into each line by pressing the **Step Into (F7)** button repeatedly. Once you have stepped through all the statements in that function, Data Studio jumps to *Line 10* in *f2\_test()*.

The currently debugging object is marked with  symbol in the tab title with the function name.



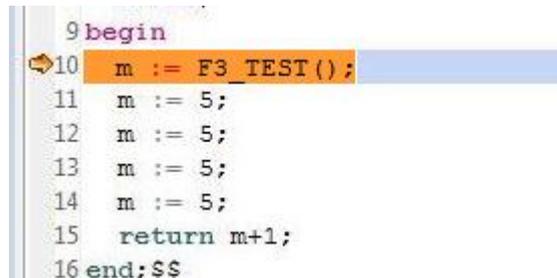
```
pres@[1] f2_test() - integer f3_test() - integer
1 CREATE OR REPLACE FUNCTION public.f2_test()
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5
6 DECLARE
7
8 m int;
9 begin
10 m := F3_TEST();
11 m := 5;
12 m := 5;
13 m := 5;
14 m := 5;
15 return m+1;
16 end;$$
17

pres@[1] f2_test() - integer f3_test() - integer
1 CREATE OR REPLACE FUNCTION public.f3_test()
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6
7 m int;
8 begin
9 m := 5;
10 return m+1;
11 end; $$
12
13
```

## Step Over

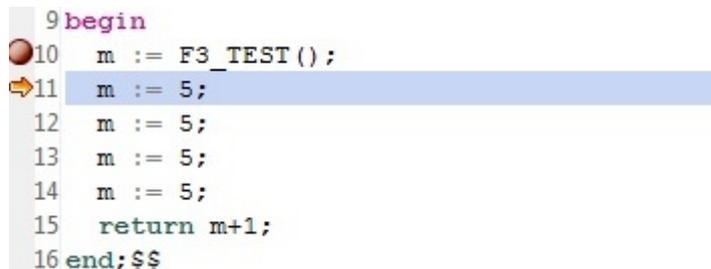
Step over is the same as step into, except that when it reaches a call for another function, it will not step into the function. The function will run, and you will be brought to the next statement in the current function. **F8** is the shortcut key for step over. However, if there is a breakpoint set inside the called function, step over will enter the function, and hit the set breakpoint.

In the below example, when you click **Step Over** in *Line 10*, Data Studio runs the *f3\_test()* function.



```
9 begin
10 m := F3_TEST();
11 m := 5;
12 m := 5;
13 m := 5;
14 m := 5;
15 return m+1;
16 end;$$
```

The cursor will be moved to the next statement in *f2\_test()*, that is, *Line 11* in *f2\_test()*.



```
9 begin
10 m := F3_TEST();
11 m := 5;
12 m := 5;
13 m := 5;
14 m := 5;
15 return m+1;
16 end;$$
```

You can step over a function when you are familiar with the way the function works and are sure that its execution will not affect the issue that you are investigating.

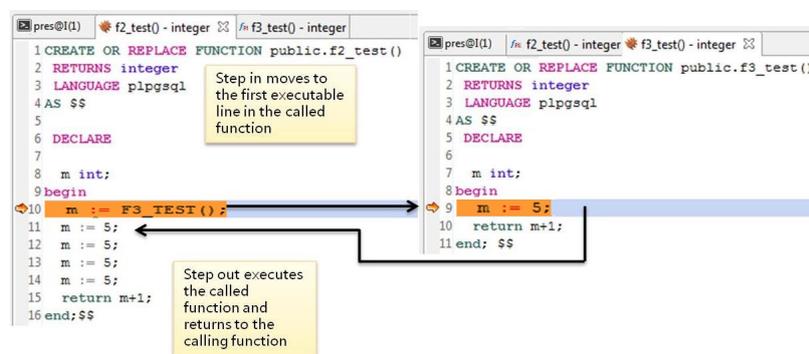
 **NOTE**

Stepping over a line of code that does not contain a function call executes the line just like stepping into the line.

## Step Out

Stepping out of a sub-program continues execution of the function and then suspends execution after the function returns to its calling function. You can step out of a long function when you have determined that the rest of the function is not significant to debug. However, if a breakpoint is set in the remaining part of the function, then that breakpoint will be hit before returning to the calling function.

Both stepping over and stepping out of a function will execute a function. The shortcut key for the step out operation is **Shift+F7**.



In the preceding example,

- Choose **Debug > Step Into** to step into *f3\_test()*.
- Choose **Debug > Step Out** to step out of *f3\_test()*

## Continuing the Debug Execution

When the debugging process stops at a specific location, you can select **Continue (F9)** from the **Debug Menu** or click  button from the toolbar to continue the PL/SQL function execution.

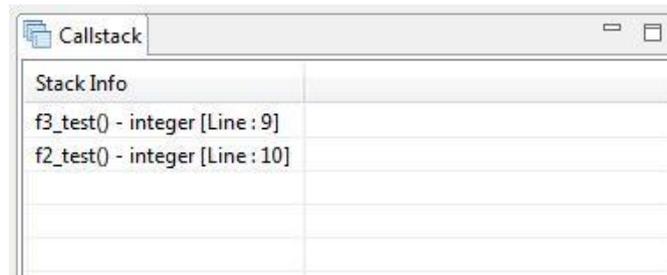
## Viewing Callstack

The **Callstack** pane displays the chain of functions as they are called. The Callstack pane can be opened from the minimized window panel. The most recent functions are listed on the top, and the least recent on the bottom. At the end of each function name is the current line number in that function.

You can navigate among multiple functions through the **Callstack** pane by double-clicking the function name in the **Callstack** pane.

For example, when *f2\_test()* calls *f3\_test()* at *Line 10*, the debug pointer will point to the first valid executable line (which is *Line 9*, in the above example) in the called function.

In this case, the **Callstack** pane will be as shown below:

**NOTE**

The content of the **Callstack** pane can be copied to the clipboard using **Alt+J**.

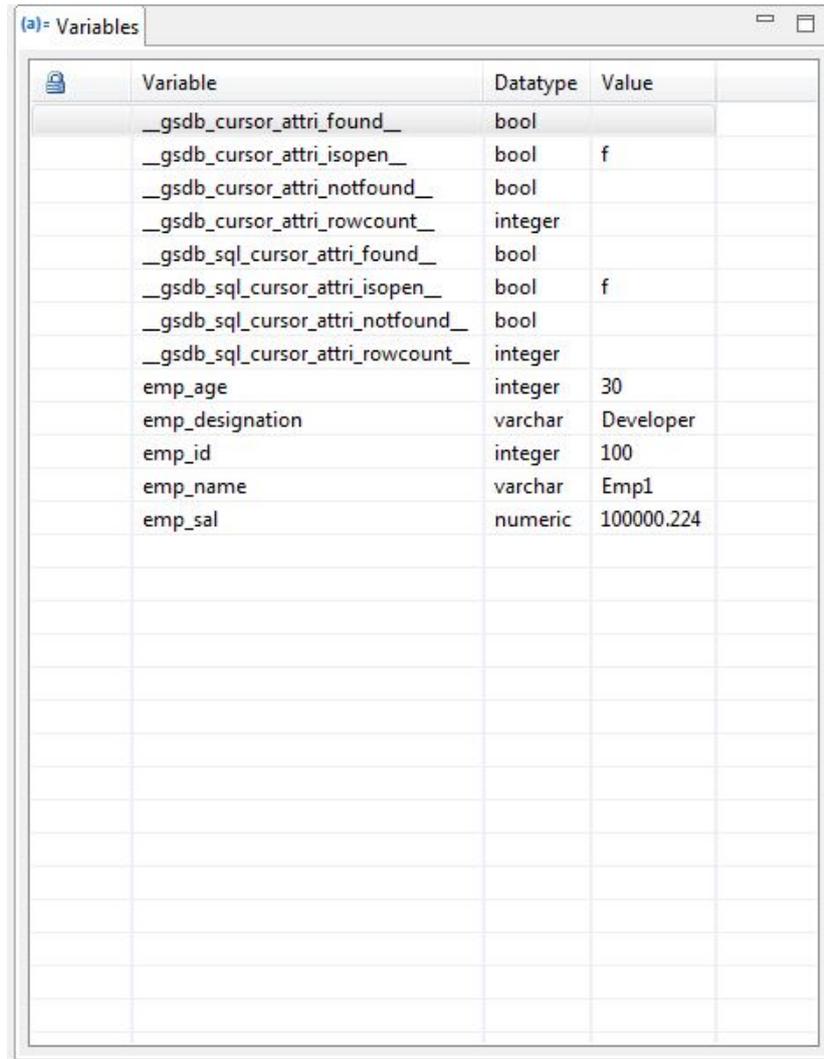
#### 6.5.4.2.4 Checking Debug Information

When you use Data Studio, you can examine debugging information through several debug tabs. This section describes the operations that can be performed to check the debug information:

- [Operating on Variables](#)
- [Viewing Results](#)

#### Operating on Variables

The **Variables** pane is used to monitor information or evaluate values. The Variables pane can be opened from the minimized window panel. Using this pane, you can evaluate or modify variables or arguments in a PL/SQL function. As you step through the code, the values of some local variables may change.



Variable	Datatype	Value
 _gsdb_cursor_attri_found_	bool	
 _gsdb_cursor_attri_isopen_	bool	f
 _gsdb_cursor_attri_notfound_	bool	
 _gsdb_cursor_attri_rowcount_	integer	
 _gsdb_sql_cursor_attri_found_	bool	
 _gsdb_sql_cursor_attri_isopen_	bool	f
 _gsdb_sql_cursor_attri_notfound_	bool	
 _gsdb_sql_cursor_attri_rowcount_	integer	
emp_age	integer	30
emp_designation	varchar	Developer
emp_id	integer	100
emp_name	varchar	Emp1
emp_sal	numeric	100000.224

 **NOTE**

The content of the **Variables** pane can be copied to the clipboard using **Alt+K**.

You can double-click the corresponding row of the variable and manually change variable values during run-time.

Click the **Variable**, **Datatype**, or **Value** column in the **Variables** pane to sort the values.

For example, to change the value of the percentage variable from 5 to 15, double-click the corresponding row in the **Variable** pane. The **Set Variable Value** dialog box will open, which prompts you to input the variable value. Provide the variable value and click **OK**.

To set NULL as the variable value, do not enter any value in text box.

If the variable is read-only, it will be indicated by  beside the corresponding variable.

A variable declared as a constant will not be shown as read-only in the **Variables** pane; however while updating it, an error will be seen.

 **NOTE**

- In the **Variables** pane, the parameter value will be displayed as *NULL*, if the input to the parameter value is string literal 'NULL'.
- When the value is set to a variable using Data Studio, then the value of the variable is same as the value returned by the statement "select expression" executed from **gsql**.

Setting/Displaying Variables	Description
Setting NULL Values	<ol style="list-style-type: none"> <li>1. Double click on a variable value in <b>Variables</b> pane. A dialog box is displayed.</li> <li>2. Make the value empty.</li> </ol>
Setting String Values	<ul style="list-style-type: none"> <li>● To set abc, enter abc.</li> <li>● To set string as Master's Degree enter Master's Degree.</li> <li>● To set variable as text(NULL), set NULL in <b>Variables</b> pane.</li> </ul>
Setting Boolean Values	<p>Enclose the boolean values <i>t</i> or <i>f</i> within single quotes.</p> <p>To set <i>t</i> to a boolean variable, enter '<i>t</i>' in the <b>Variables</b> pane.</p>
Displaying Variable Value	<p>If the variable value is NULL text, it will be displayed as <i>NULL</i>.</p> <p>If the variable value is NULL, it will be displayed as empty.</p> <p>If the variable value is a string, for example, abc, it will be displayed as abc.</p>

## Viewing Results

The **Result** tab displays the output for the PL/SQL debugging session, with the corresponding function/procedure name at the top of the tab. The **Result** tab will appear automatically, only if there is a result for the executed PL/SQL program.

You can copy the content of the **Result** tab, by clicking . Refer to [6.12.10 Working with the SQL Terminals](#) for more information.

 **NOTE**

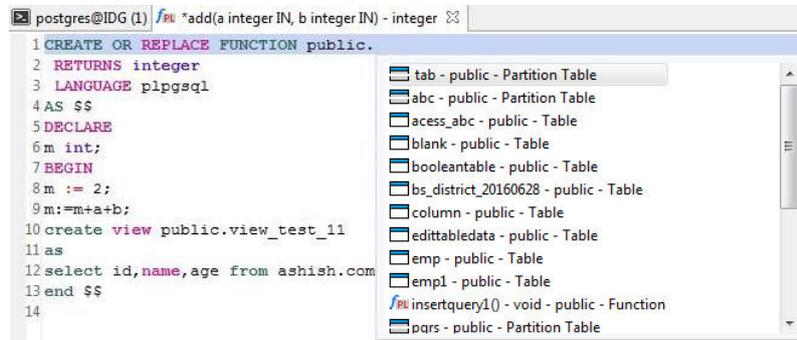
- The tool tip in the **Result** tab displays maximum of 10 lines, where each line contains maximum of 80 characters.
- If the result of the executed query is NULL, it will be displayed as *<NULL>*.
- Tab characters (ASCII 009) in table data will not be displayed in the **Results/View Table Data/Properties** window. Tab characters will be included correctly when copying/exporting the data. Tool tip will also display the tab characters correctly.

### 6.5.4.3 Selecting a DB Object in the PL/SQL Viewer

Data Studio suggests a list of possible schema names, table names, column names, views, sequences, and functions in the **PL/SQL Viewer**.

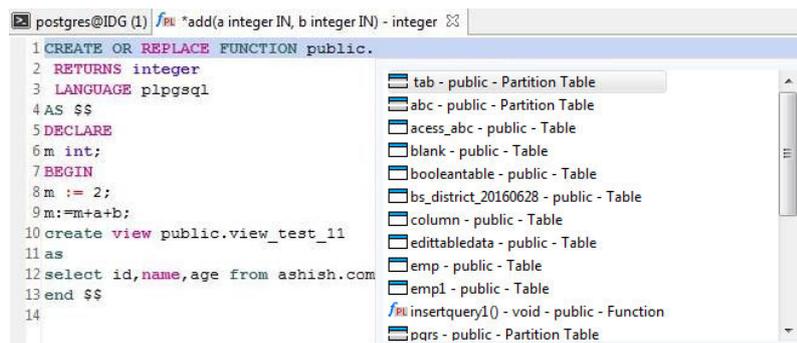
Follow the steps below to select a DB object:

- Step 1** Press **Ctrl+Space** and enter the required parent DB object name. The DB objects list is refined as you continue typing the DB object name. The DB objects list displays all DB objects of the database connected to the **SQL Terminal**.



- Step 2** To select the parent DB object, use the **Up** or **Down** arrow keys and press **Enter** on the keyboard, or double-click the parent DB object.

- Step 3** Enter . (period) to list all child DB objects.



- Step 4** To select the child DB object, use the **Up** or **Down** arrow keys and press **Enter** on the keyboard, or double-click the child DB object.

On selection, the child DB object will be appended to the parent DB object (with a period '.').

 **NOTE**

- Auto-suggest also works on schema names, table names, views, and table name aliases in the same way as shown above for all schema objects that you have access. Auto-suggest for column name alias is not supported.

Following is a sample query with alias objects:

```
SELECT
  table_alias.<auto-suggest>
FROM test.t1 AS table_alias
WHERE
  table_alias.<auto-suggest> = 5
GROUP BY table_alias.<auto-suggest>
HAVING table_alias.<auto-suggest> = 5
ORDER BY table alias.<auto-suggest>
```

- Auto-suggest may show "Loading" in Terminal for following scenarios:
  - The object is not loaded due to the value mentioned in the **Load Limit** field. Refer to [6.2.2 Adding a Connection](#) for more information.
  - The object is not loaded since it is added in the **Exclude** list option.
  - There is a delay in fetching the object from the server.
- If there are objects with the same name in different case, then auto-suggest will display child objects of both parent objects.

**Example:**

If there are two schemas with the name public and PUBLIC, then all child objects for both these schemas will be displayed.

---End

### 6.5.4.4 Exporting a Function/Procedure DDL

Follow the steps below to export the Function/Procedure DDL:

**Step 1** In the **Object Browser** pane, right-click the selected function/procedure and select **Export DDL**.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** In the **Save As** dialog box, select the location to save the DDL and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click .
- The exported file name will not be the same as function/procedure name, if the function/procedure name contains characters which are not supported by Windows.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.
- Multiple objects can be selected to export DDL. Refer to [Batch Export](#) section for list of objects not supported for export DDL operation.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes

Database Encoding	File Encoding	Supports Exporting DDL
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

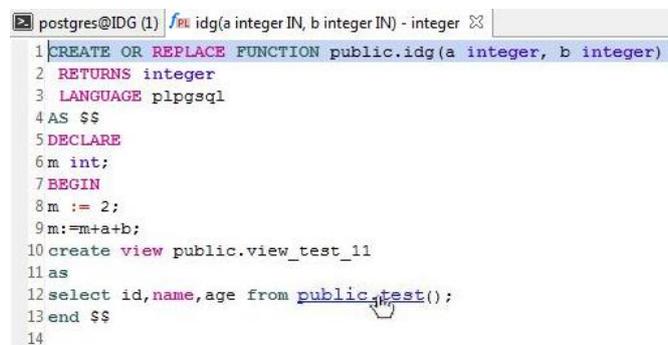
----End

### 6.5.4.5 Viewing Object Properties in the PL/SQL Viewer

Data Studio allows you to view table properties, procedures/functions and SQL functions.

Follow the steps below to view table properties:

**Step 1** Press **Ctrl** and point to the table name.



```

postgres@IDG (1) /pl idg(a integer IN, b integer IN) - integer ✕
1 CREATE OR REPLACE FUNCTION public.idg(a integer, b integer)
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6 m int;
7 BEGIN
8 m := 2;
9 m:=m+a+b;
10 create view public.view_test_11
11 as
12 select id,name,age from public.test();
13 end $$
14
    
```

**Step 2** Click the highlighted table name.

The properties of the selected table is displayed.

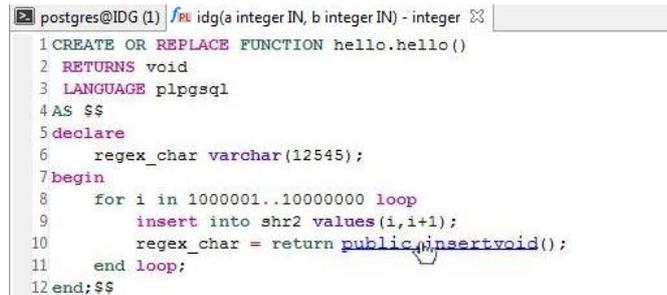
 **NOTE**

The table properties are read-only.

----End

Follow the steps below to view functions/procedures or SQL functions:

**Step 1** Press **Ctrl** and point to the procedure/function name or SQL function name.



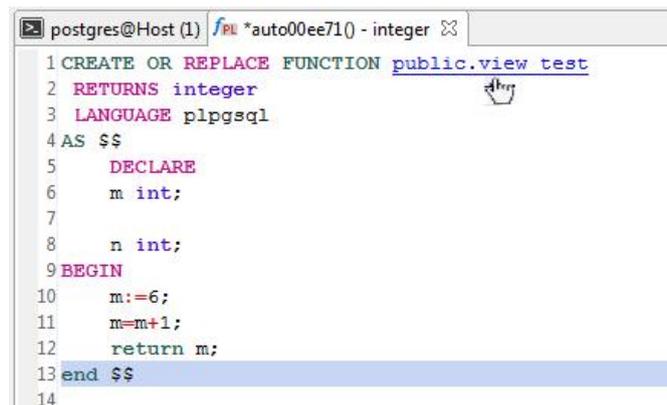
```
postgres@IDG (1) f idg(a integer IN, b integer IN) - integer
1 CREATE OR REPLACE FUNCTION hello.hello()
2 RETURNS void
3 LANGUAGE plpgsql
4 AS $$
5 declare
6     regex_char varchar(12545);
7 begin
8     for i in 1000001..10000000 loop
9         insert into shr2 values (i,i+1);
10        regex_char = return public.insertvoid();
11    end loop;
12 end;$$
```

**Step 2** Click the highlighted function/procedure name or SQL function name. The function/procedure or SQL function is displayed in a new **PL/SQL Viewer** tab based on your selection.

----End

Follow the steps below to View Object DDL:

**Step 1** Press **Ctrl** and point to the View Object DDL name.



```
postgres@Host (1) f *auto00ee710 - integer
1 CREATE OR REPLACE FUNCTION public.view test
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5     DECLARE
6         m int;
7
8         n int;
9 BEGIN
10    m:=6;
11    m=m+1;
12    return m;
13 end $$
14
```

**Step 2** Click the highlighted View Object DDL name. The View Object DDL is displayed in a new tab based on your selection.

----End

### 6.5.4.6 Dropping a Function/Procedure

Individual or batch drop can be performed on functions/procedures. Refer to [6.13.2 Dropping Batch of Objects](#) section for batch drop.

Follow the steps below to drop a function/procedure or SQL function object:

**Step 1** In the **Object Browser** pane, right-click the selected function/procedure object and select **Drop Object**.

**Step 2** In the confirmation dialog box, click **Yes** to complete the operation successfully.

The status bar displays the status of the completed operation.

----End

### 6.5.4.7 Executing a Function/Procedure

After you connect to the database, all the stored functions/procedures and tables will be automatically populated in the **Object Browser** pane. You can use Data Studio to execute PL/SQL programs or SQL functions.

#### NOTE

- Blank lines occurring above or below in a function/procedure will be trimmed by Data Studio before being sent to the server. Blank lines will also be trimmed when displaying the source received from the server.
- To execute a function/procedure, enter the same values in Data Studio and the **gsql** client. If you do not enter any value in Data Studio, then *NULL* is considered as the input.

For example:

- To execute the function/procedure with string, enter the value as *data*
- To execute the function/procedure with date, enter the value as *to\_date('2012-10-10', 'YYYY-MM-DD');*
- A function/procedure with OUT and INOUT parameter types cannot be executed directly.
- Data Studio will not execute any function/procedure with unknown data type parameters.

You can right-click the function/procedure in the **Object Browser** to perform the following operations:

- Refresh the program to get the latest program from the server
- Execute the function/procedure or SQL function
- Debug the PL/SQL function
- Drop the debug object

### How to execute a PL/SQL program or SQL function?

Follow the steps below to execute a PL/SQL program or SQL function:

**Step 1** Double-click and open the PL/SQL program or SQL function. Each debug object will open in a new tab. You can open a maximum of 100 tabs in Data Studio.

**Step 2** Click  in the toolbar, or choose **Run > Execute DB Object** from the main menu, or right-click the program name in the **Object Browser** and select **Execute**.

Alternatively, you can right-click in the **PL/SQL Viewer** tab and select **Execute**.

**Step 3** The **Execute Function/Procedure** dialog box is displayed prompting for your input.

#### NOTE

If there is no input parameter, then the **Execute Function/Procedure** dialog box will not appear. Instead, the PL/SQL program will execute and the result (if any) will be displayed in the **Result** tab.

**Step 4** Provide your input for the function/procedure in the **Execute PL/pgSQL** dialog box and click **OK**.

To set NULL as the parameter value, enter *NULL* or *null*.

- If you do not provide a value that starts with a single quote, then a single quote (') will be added by Data Studio before and after the value and typecasting is done.
- If you provide a value that starts with a single quote, an additional single quote will not be added by Data Studio, but data type typecasting is done.

For example:

For supported data types, the execution queries are as follows:

```
select func('1'::INTEGER);  
select func('1'::FLOAT);  
select func('xyz'::VARCHAR);
```

- If quotes are already provided, you need to take care of escaping the quotes.

For example:

If the input value is *ab'c*, then you need to enter *ab"'*.

The PL/SQL program is executed in the **SQL Terminal** tab and the result is displayed in the **Result** tab. You can copy the contents of the **Result** tab by clicking . Refer to [6.12.10 Working with the SQL Terminals](#) for more information on toolbar options.

Refer to [Execute SQL Queries](#) section for information on reconnect option in case connection is lost during execution.

----End

### 6.5.4.8 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click selected function/procedure and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Refer to [6.5.3 Grant/Revoke Privilege](#) section to grant/revoke privilege.

----End

## 6.6 Tables(GaussDB 200&GaussDB 300)

### 6.6.1 Overview

This section describes how to work with tables effectively.

#### NOTE

You need to fill all the mandatory parameters that are marked with asterisk (\*) to complete the operation successfully.

### 6.6.2 Creating Regular Table

#### 6.6.2.1 Overview

This section describes the steps to create a Regular table.

Tables are logical structures maintained by the database manager. Tables are made up of columns and rows. You can define tables as part of your data definitions in the data perspective. Before you can define a table, you must first have a database and a schema defined. This section shows you how to create new table using Data Studio.

Follow the steps below to define a table in your database:

- Step 1** In the **Object Browser** pane, right-click **Regular Tables**, and select **Create Regular Table**.
- Step 2** Provide basic table information such as table name, tablespace, table type and so on. For more details, refer to [Providing General Information](#). To create a tablespace, refer to [Working with Tablespaces](#).
- Step 3** Define column related information such as column name, data type schema, data type, and column constraints. For more details, refer to [Defining Columns](#).
- Step 4** Select the data distribution information for the table. For more details, refer to [Selecting Data Distribution](#).
- Step 5** Define the column constraints for different constraint types such as primary key, unique, and check. For more details, refer to [Defining Table Constraints](#).
- Step 6** Define the index information for the table such as index name, access method, tablespace and so on. For more details, refer to [Defining Indexes](#).

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided. For more details, refer to [SQL Preview](#).

----End

## Providing General Information

When you create a table within a schema, the current schema is used as the table's schema. There are several steps involved in creating a table.

Provide the following information to create a regular table:

- Step 1** Enter the table name in the **Table Name** field. It specifies the name of the table to be created.

### NOTE

Select **Case** check box to retain the capitalization of the text entered in **Table Name** field. For example, if the table name entered is "Employee", then the table name is created as "Employee".

The schema name under which the table is created is displayed in the **Schema** drop-down.

- Step 2** Select table orientation from **Table Orientation** drop-down.
- Step 3** Select the tablespace from **Tablespace** drop-down. It specifies the name of the tablespace in which the new table is to be created. If not provided, the default tablespace of the table's schema will be used.
- Step 4** Select the table type from **Table Type** drop-down. It specifies the type of the table.
  - **Normal**: If specified, the table is created as a normal table.
  - **Unlogged**: If specified, the table is created as an unlogged table. Data written to unlogged tables is not written to the write-ahead log, which makes it considerably faster than ordinary tables. However, it is not crash-safe. An unlogged table is automatically truncated after a crash or unclean shutdown. The contents of an unlogged table are also not replicated to standby servers. Any indexes created on an unlogged table are also automatically unlogged.
- Step 5** Select the required **Options**.
  - **IF NOT EXISTS** check box to create the table only if table with same name does not exist.

- **WITH OIDS** check box for the new table to have OIDs (object identifiers) assigned. If you need a new table with OIDs, choose this option.
- Select the **Fill Factor**. The fill factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default value.

When a smaller fill factor is specified, INSERT operations pack table pages only to the indicated percentage; the remaining space on each page is reserved for updating rows on that page. This gives UPDATE operation a chance to place the updated copy of a row on the same page as the original, which is more efficient than placing it on a different page. For a table whose entries are never updated, complete packing is the best choice, but in heavily updated tables, smaller fill factors are appropriate. This parameter cannot be set for TOAST tables.

**Step 6** Enter the description of the table in **Description of Table** box. It specifies a short note on the table.

**Step 7** After providing the general information about the table, click **Next** to define the columns information for the table.

----**End**

Table below lists the supported fields for each Regular type tables:

**Table 6-1** Supported Fields

Field Name	Row Table	Column Table	ORC Table
Tablespace	Normal	Normal	HDFS
Table Type	✓	✓	✗
If Not Exists	✓	✓	✓
With OIDS	✓	✗	✗
Fill Factor	✓	✗	✗

## Defining Columns

A column defines a unit of information within a table's row. Each row is an entry in the table. Each column is a category of information that applies to all rows. When you add a table to a database, you can define the columns that compose it. Columns determine the type of data that the table can hold.

After providing the general information about the table, click the **Columns** tab to define the list of table columns. Each column contains name, data type, and other optional properties.

You can perform the following operations on an existing column only for a Regular table:

- **Deleting a Column**
- **Editing a Column**

- **Moving a Column**

Follow the steps below to define column(s) for the table:

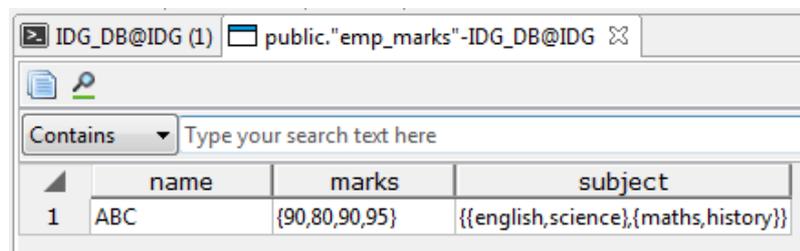
**Step 1** Enter the column name in **Column Name** field. It specifies the name of a column to be created in the new table. This must be a unique name in the table.

 **NOTE**

Select **Case** check box to retain the capitalization of the text entered in **Column Name** field. For example, if the column name entered is "Name", then the column name is created as "Name".

**Step 2** Select the **Array Dimensions**. It specifies the array dimensions for the column.

**Example:** If array dimension for a column is defined as integer [], then it will add the column data as single dimension array.



	name	marks	subject
1	ABC	{90,80,90,95}	{{english,science},{maths,history}}

The **marks** column in the above table was created as single dimension and **subject** column as two dimensions.

**Step 3** Select the data type of the column from **Data Type** drop-down. For example, **bigint** for integer values.

For complex data types,

- Select the required schema from the **Data type Schema** drop-down list.
- Select the corresponding data type from the **Data Type** drop-down list. This list displays the tables and views for the selected schema.

 **NOTE**

User defined data type will not be available for selection.

**Step 4** Enter the precision/size value of the datatype entered in the **Precision/Size** field. This option is available only if a data type can be defined with precision/size.

**Step 5** Select the scale of the data type entered in the **Scale** field.

**Step 6** Choose the following **Column Constraints** if required:

- **NOT NULL** - Specifies that this column is not allowed to contain null values.
- **UNIQUE** - Specifies that a column may contain only unique values.
- **DEFAULT** - Specifies the value that will be used for this column in case no value is defined.
- **CHECK** - Specifies an expression producing a boolean result which new or updated rows must satisfy for an INSERT or UPDATE operation to succeed.

**Step 7** After you enter all information for new column, click **Add**. You can also delete a column from a list or change the order of columns. After defining all columns, click **Next**.

----End

Table below lists the supported fields for each Regular type tables:

**Table 6-2** Supported Fields

Field Name	Row Table	Column Table	ORC Table
Array Dimensions	✓	✗	✗
Data type Schema	✓	✗	✗
Not Null	✓	✓	✓
Default	✓	✓	✓
Unique	✓	✗	✗
Check	✓	✗	✗

## Deleting a Column

Follow the steps below to delete a column:

**Step 1** Select the required column.

**Step 2** Click **Delete**.

----End

## Editing a Column

Follow the steps below to edit a column:

**Step 1** Select the required column.

**Step 2** Click **Edit**.

**Step 3** Edit the column details as required and click **Update** to save changes.

 **NOTE**

You must complete the edit operation and save the changes to continue with other operations.

----End

## Moving a Column

You can move a column to change the location of the column as required in the table. To move a column, select the required column and click **Up** or **Down**.

## Selecting Data Distribution

Data distribution specifies how the table is distributed or replicated among data nodes.

Select one of the following options for the distribution type:

Distribution Type	Description
DEFAULT DISTRIBUTION	Specifies that the default distribution type will be assigned for this table.
REPLICATION	Each row of the table will be replicated in all the data nodes of the database cluster.
HASH	Each row of the table will be placed based on the hash value of the specified column.

After selecting data distribution, click **Next**.

Table below lists the supported options for each Regular type tables:

**Table 6-3** Supported Options

Option Name	Row Table	Column Table	ORC Table
Default	✓	✓	✗
Hash	✓	✓	✓
Replication	✓	✓	✗

## Defining Table Constraints

Creating constraints is optional. A table can have one (and only one) primary key. Creating the primary key is a good practice.

You can select the following types of constraints from the **Constraint Type** drop-down list:

- **Primary Key**
- **Unique**
- **Check**

### Primary Key

The primary key is the unique identity of a row and consists of one or more columns.

Only one primary key can be specified for a table, either as a column constraint or as a table constraint. The primary key constraint must name a set of columns that is different from other sets of columns named by any unique constraint defined for the same table.

Select the constraint type as **PRIMARY KEY** in the combo box and enter the constraint name. Select the column from **Available Columns** list and click **Add**. If you need a multi-column primary key, repeat this step for another column.

Optionally, you can select **On Tablespace** in which the index associated with a **PRIMARY KEY** constraint will be created. If this parameter is not provided, the index will be created in the same tablespace as the table.

**Fillfactor** for a table is a percentage between 10 and 100. The default value is 100 (complete packing). When a smaller fill factor is specified, INSERT operations will pack table pages only up to the indicated percentage; the remaining space on each page is reserved for updating rows on that page. This gives UPDATE operation a chance to place the updated copy of a row on the same page as the original, which is more efficient than placing it on a different page.

For a table whose entries are never updated, complete packing is the best choice, but in heavily updated tables, smaller fill factors are appropriate. This parameter cannot be set for TOAST tables.

**DEFERRABLE**: Select this check box to defer this option.

**INITIALLY DEFERRED**: Select this check box to check the constraint at the set default time.

Click **Add** in the **Constraints** group box.

You can remove a primary key from the list using the **Delete** button.

Mandatory parameters are marked with asterisk (\*) in the corresponding field.

## Unique

Select the constraint type as **UNIQUE** in the combo box and enter the constraint name in the text box.

Select column in **Available Columns** list and click **Add**. If you need multi-column unique, repeat this step for another column. After adding the first column, the unique name is automatically filled from the table name. You can also change this name.

Optionally, you can select **Tablespace** in which the index associated with a unique constraint needs to be created. If this parameter is not provided, the index will be created in the same tablespace as the table.

**Fillfactor** - Refer to [Primary Key](#) section for fillfactor information.

**DEFERRABLE**: Refer to [Primary Key](#) section for deferrable information.

**INITIALLY DEFERRED**: Refer to [Primary Key](#) section for initially deferred information.

You can a remove unique from the list using **Delete** button.

Mandatory parameters are marked with asterisk (\*) in the corresponding filed.

## Check

Select the constraint type as **CHECK** in the combo box and enter the constraint name in the text box.

When the INSERT or UPDATE operation is performed, and if the check expression fails, then table data is not altered.

If you double-click on column in **Available Columns** list, it is inserted to **Check Expression** edit line to current cursor position.

Then, click **Add** in **Constraints** group box. You can also remove a check from the list using **Delete** button. Mandatory parameters are marked with asterisk (\*) in the corresponding field.

After defining all constraints, click **Next**.

Table below lists the supported options for each Regular type tables:

**Table 6-4** Supported Options

Option Name	Row Table	Column Table	ORC Table
Check	✓	✗	✗
Unique	✓	✗	✗
Primary Key	✓	✗	✗

## Defining Indexes

Creating indexes is optional. Indexes are primarily used to enhance database performance. This operation constructs an index on the specified column(s) of the specified table. Select the **Unique Index** check box to enable this option.

Choose the name of the index method from the **Access Method** list. The default method is B-tree. Select the **Tablespace** in which the index must be created. If not specified, the index is created in the default tablespace.

The **Fillfactor** for an index is a percentage that determines how full the index method will try to pack index pages. For B-trees, leaf pages are filled to this percentage during initial index build, and also when extending the index at the right (adding new largest key values). If pages subsequently become completely full, they will be split, leading to gradual degradation in the index's efficiency. B-trees use a default fill factor of 90, but any integer value from 10 to 100 can be selected. If the table is static, then a fill factor of 100 is best to minimize the index's physical size. For heavily updated tables, a explain plan smaller fill factor is better to minimize the need for page splits. Other index methods use fill factor in different but roughly analogous ways; the default fill factor varies between methods.

You can either enter a user-defined expression for the index or you can create the index using the **Available Columns** list. Select the column in the **Available Columns** list and click **Add**. If you need a multi-column index, repeat this step for other columns.

After entering the required information for the new index, click **Add**.

You can also delete an index from the list using the **Delete** button. After defining all indexes, click **Next**.

Table below lists the supported fields/options for each Regular type tables:

**Table 6-5** Supported Fields/Options

Field/Option Name	Row Table	Column Table	ORC Table
Unique Index	✓	✗	✗
btree	✓	✓	✗
gin	✓	✓	✗
gist	✓	✓	✗
hash	✓	✓	✗
psort	✓	✓	✗
spgist	✓	✓	✗
Tablespace	Normal	Normal	✗
Fill Factor	✓	✗	✗
User Defined Expression	✓	✗	✗
Partial Index	✓	✗	✗

## SQL Preview

Data Studio generates a DDL statement based on the inputs provided in **Create New table** wizard.

You can only view, select, and copy the query. You cannot edit the query.

- To select all queries, press **Ctrl+A** or right-click and select **Select All**.
- To copy the selected query, press **Ctrl+C** or right-click and select **Copy**.

Click **Finish** to create the table. On clicking the **Finish** button, the generated query will be sent to the server. Any errors are displayed in the dialog box and status bar.

### 6.6.2.2 Working with Columns

After creating a table, you can add new columns in that table. You can also perform the following operations on the existing column only for a Regular table:

- **Creating New Column**

- [Rename Column](#)
- [Toggle Not Null](#)
- [Drop Column](#)
- [Set Column Default](#)
- [Change Data Type](#)

## Creating New Column

Follow the steps below to add a new column to the existing table:

**Step 1** Right-click **Columns** and select **Create column**.

The **Add New Column** dialog box is displayed prompting you to add information about the new column.

**Step 2** Enter the details and click **Add**. You can view the added column in the corresponding table.

Data Studio displays the status of the operation in the status bar.

----End

## Rename Column

Follow the steps below to rename a column:

**Step 1** Right-click the selected column and select **Rename Column**.

A **Rename Column** dialog box is displayed prompting you to provide the new name.

**Step 2** Enter the name and click **OK**. Data Studio displays the status of the operation in the status bar.

----End

## Toggle Not Null

Follow the steps below to set or reset the Not Null option:

**Step 1** Right-click the selected column and select **Toggle Not Null**.

A **Toggle Not Null Property** dialog box is displayed prompting you to set or reset the Not Null option.

**Step 2** In the confirmation dialog box, click **OK** to complete the operation successfully. Data Studio displays the status of the operation in the status bar.

----End

## Drop Column

Follow the steps below to drop the column:

**Step 1** Right-click the selected column and select **Drop Column**. This operation deletes the column from the table.

A **Drop Column** dialog box is displayed.

**Step 2** Click **OK** to complete the operation successfully. Data Studio displays the status of the operation in the status bar.

----End

## Set Column Default

Follow the steps below to set the default value for the column:

**Step 1** Right-click the selected column and select **Set Column Default Value**.

A dialog box with the current default value (if it is set) and prompting you to provide the default value is displayed.

**Step 2** Enter the value and click **OK**. Data Studio displays the status of the operation in the status bar.

----End

## Change Data Type

Follow the steps below to change the data type of the column:

**Step 1** Right-click the selected column and select **Change Data Type**.

**Change Data Type** dialog box is displayed.

### NOTE

The existing data type will show as Unknown while modifying complex data types.

**Step 2** Select the **Data type Schema** and **Data Type**. If the **Precision/Size** spin box is enabled, enter the required details and click **OK**. Data Studio displays the status of the operation in the status bar.

----End

### 6.6.2.3 Working with Constraints

You can perform the following operations after a table is created only for a Regular table:

- [Creating a Constraint](#)
- [Renaming a Constraint](#)
- [Dropping a Constraint](#)

## Creating a Constraint

Follow the steps below to add a new constraint to the existing table:

**Step 1** Right-click the selected constraint of the table and select **Create constraint**.

An **Add New Constraint** dialog box is displayed prompting you to add information about the new constraint.

**Step 2** Enter the **Constraint Name**, **Check Expression**, and click **Add**. You can view the added constraint in the corresponding table.

Data Studio displays the status of the operation in the status bar.

 **NOTE**

The status bar will show the name of the constraint if it has been provided in the **Constraint Name** field, else the constraint name will not be displayed as it is created by database server.

---End

## Renaming a Constraint

Follow the steps below to rename a constraint:

**Step 1** Right-click the selected constraint and select **Rename Constraint**.

A **Rename Constraint** dialog box is displayed prompting you to provide the new name.

**Step 2** Enter the constraint name and click **OK**. Data Studio displays the status of the operation in the status bar.

---End

## Dropping a Constraint

Follow the steps below to drop the constraint:

**Step 1** Right-click the selected constraint and select **Drop Constraint**.

A **Drop Constraint** dialog box is displayed.

**Step 2** Click **OK** to complete the operation successfully. Data Studio displays the status of the operation in the status bar.

---End

### 6.6.2.4 Working with Indexes

An index can be created in a table to find data quickly and efficiently.

After creating a table, you can add a new index to that table. You can also perform the following operations on an existing index only for a Regular table:

- [Creating a New Index](#)
- [Renaming an Index](#)
- [Changing the Tablespace](#)
- [Changing the Fill Factor](#)
- [Dropping an Index](#)

## Creating a New Index

Follow the steps below to add a new index to the existing table:

**Step 1** Right-click **Indexes** and select **Create index**.

A **Create Index** dialog box is displayed prompting you to add information about the index.

**Step 2** Enter the details and click **Create**. You can also view the SQL statement by clicking the **Preview Query** button. Items in **Available Columns** are not sorted. Items moved back from **Index Columns** to **Available Columns** are unsorted, and is not related to the column order in the table. You can set the order of the **Index Columns** using the arrow buttons. Data Studio displays the status of the operation in the status bar.

---End

## Renaming an Index

Follow the steps below to rename an index:

**Step 1** Right-click the selected index and select **Rename Index**.

A **Rename Index** dialog box is displayed prompting you to provide the new name.

**Step 2** Enter the name and click **OK**. Data Studio displays the status of the operation in the status bar.

---End

## Changing the Tablespace

Follow the steps below to change the tablespace:

**Step 1** Right-click the selected index and select **Change Tablespace**.

A **Change Tablespace** dialog box is displayed prompting you to select the tablespace details.

**Step 2** Select the tablespace and click **OK**. Data Studio displays the status of the operation in the status bar.

---End

## Changing the Fill Factor

Follow the steps below to change the fill factor:

**Step 1** Right-click the selected index and select **Change Fill Factor**.

A **Change Fill Factor** dialog box is displayed prompting you to select the fill factor details.

**Step 2** Select the fill factor and click **OK**. Data Studio displays the status of the operation in the status bar.

---End

## Dropping an Index

Follow the steps below to drop an index:

**Step 1** Right-click the selected index and select **Drop Index**. Data Studio prompts you to confirm this operation.

The **Drop Index** dialog box is displayed.

**Step 2** Click **OK** to complete the operation successfully. Data Studio displays the status of the operation in the status bar. This operation deletes the index from the table.

 **NOTE**

When the last index of a table is dropped and if the table properties are checked, then **Has Index** may reflect the value "TRUE", though the table has no index. This value is updated to "FALSE" when a vacuum operation on the table is performed.

----End

## 6.6.3 Creating Foreign Table

Foreign tables created using query execution in SQL Terminal or any other tool can be viewed in the Object browser after refresh.

**Step 1** To view the newly created foreign table, right-click and select **Refresh** either at database, schema and foreign table group level.

 **NOTE**

- GDS Foreign table is denoted with  icon before the table name.
- HDFS Foreign table is denoted with  icon before the table name.
- HDFS Foreign table with partition is denoted with  icon before the table name.

----End

## 6.6.4 Creating Partition Table

### 6.6.4.1 Overview

Partitioning refers to splitting what is logically one large table into smaller physical pieces based on specific schemes. The table based on the logic is called a partition table, and a physical piece is called a partition. Data is stored on these smaller physical pieces, namely, partitions, instead of the larger logical partition table.

Follow the steps below to define a table in your database:

- Step 1** In the **Object Browser** pane, right-click **Regular Tables**, and select **Create Partition Table**.
- Step 2** Provide basic table information such as table name, tablespace, table type and so on. For more details, refer to [Providing General Information](#).
- Step 3** Define column related information such as column name, data type schema, data type, and column constraints. For more details, refer to [Defining Columns](#).
- Step 4** Select the data distribution information for the table. For more details, refer to [Selecting Data Distribution](#).
- Step 5** Define the column constraints for different constraint types such as primary key, unique, and check. For more details, refer to [Defining Table Constraints](#).
- Step 6** Define the index information for the table such as index name, access method, tablespace and so on. For more details, refer to [Defining Indexes](#).
- Step 7** Define the partition information for the table such as partition name, partition column, partition value and so on. For more details, refer to [Defining Partitions](#).

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided. For more details, refer to [SQL Preview](#).

---End

## Providing General Information

Provide the following information to create a table:

For information on completing the below fields refer to [Providing General Information](#).

- Table Name
- Schema
- Tablespace
- Options
- Description of Table

For completing all other fields refer below:

**Step 1** Select table orientation from **Table Orientation** drop-down.

 **NOTE**

If table orientation is selected as ORC, then a HDFS Partition table is created.

**Step 2** Enter the ORC version number in the **ORC Version** field. This is applicable only for HDFS Partition table.

**Step 3** After providing the general information about the table, click **Next** to define the columns information for the table.

Table below lists the supported fields for each Partition type tables:

**Table 6-6** Supported Fields

Field Name	Row Partition	Column Partition	ORC Partition
Tablespace	Normal	Normal	HDFS
Table Type	✘	✘	✘
If Not Exists	✔	✔	✔
With OIDS	✘	✘	✘
Fill Factor	✔	✘	✘

---End

## Defining Columns

Refer to [Defining Columns](#) to define column(s) for the table.

Table below lists the supported fields for each Partition type tables:

**Table 6-7** Supported Fields

Field Name	Row Partition	Column Partition	ORC Partition
Array Dimensions	✓	✗	✗
Data type Schema	✓	✗	✗
Not Null	✓	✓	✓
Default	✓	✓	✓
Unique	✓	✗	✗
Check	✓	✗	✗

## Change Order of Partition

You can change the order of partition as required in the table. To change the order, select the required partition and click **Up** or **Down**.

## SQL Preview

Refer to [SQL Preview](#)

## Editing a Partition

Follow the steps below to edit a partition:

- Step 1** Select the required partition.
- Step 2** Click **Edit**.
- Step 3** Edit the partition details as required and click **Update** to save changes.

### NOTE

You must complete the edit operation and save the changes to continue with other operations

----End

## Deleting a Partition

Follow the steps below to delete a partition:

- Step 1** Select the required partition.
- Step 2** Click **Delete**.

----End

**Table 6-8** Supported Fields/Options

Field/Option Name	Row Partition	Column Partition	ORC Partition
Partition Type	By Range	By Range	By Values
Partition Name	✓	✓	✗
Partition Value	✓	✓	✗
Tablespace	Normal	Normal	✗

Table below lists the supported fields/options for each Partition type tables:

**Step 1** If **Row** or **Column** is selected as **Table Orientation** in the **General** tab, then **By Range** is displayed in the **Partition Type** section. If **ORC** is selected as **Table Orientation** in the **General** tab, then **By Value** is displayed in the **Partition Type** section.

**Step 2** Select the column based on which partition needs be defined from the **Available Column**

section and click .

The column moves to the **Partition Column** section.

 **NOTE**

- If **Table Orientation** is selected as **Row** or **Column**, then only one column can be selected for partition.
- If **Table Orientation** is selected as **ORC**, then maximum of four columns can be selected for partition.
- A maximum of 4 columns can be selected to define partition.

**Step 3** Enter a name for the partition in **Partition Name** field.

**Step 4** Click  next to the **Partition Value** field.

1. Enter the value by which you want to partition the table in **Value** column.
2. Click **OK**.

**Step 5** Select the tablespace name from the **Tablespace** drop-down.

 **NOTE**

**Partition Name**, **Partition Value**, **Tablespace** and **Partitions** section are disabled for ORC Partition tables.

**Step 6** After you enter all information for partition, click **Add**.

**Step 7** After defining all partitions, click **Next**.

----**End**

Follow the steps below to define partition(s) for the table:

- [Deleting a Partition](#)
- [Editing a Partition](#)
- [Editing a Partition](#)

You can perform the following operations on an existing partition for Row or Column Partition table. Below operations are not applicable for ORC Partition table:

## Defining Partitions

After defining the indexes, click the **Partitions** tab to define the partitions.

**Table 6-9** Supported Fields/Options

Field/Option Name	Row Partition	Column Partition	ORC Partition
Unique Index	✓	✗	✗
btree	✓	✓	✗
gin	✓	✓	✗
gist	✓	✓	✗
hash	✓	✓	✗
psort	✓	✓	✗
spgist	✓	✓	✗
Tablespace	Normal	Normal	✗
Fill Factor	✓	✗	✗
User Defined Expression	✓	✗	✗
Partial Index	✓	✗	✗

Table below lists the supported fields/options for each Partition type tables:

## Defining Indexes

Refer to [Defining Indexes](#) to define table indexes.

**Table 6-10** Supported Options

Option Name	Row Partition	Column Partition	ORC Partition
Check	✓	✗	✗
Unique	✓	✗	✗
Primary Key	✓	✗	✗

Table below lists the supported options for each Partition type tables:

## Defining Table Constraints

Refer to [Defining Table Constraints](#) to define table constraint(s).

**Table 6-11** Supported Options

Option Name	Row Partition	Column Partition	ORC Partition
Default	✓	✓	✗
Hash	✓	✓	✓
Replication	✓	✓	✗

Table below lists the supported options for each Partition type tables:

## Selecting Data Distribution

Refer to [Selecting Data Distribution](#) for selecting the distribution type.

### 6.6.4.2 Working with Partitions

After creating a table, you can add/modify partitions. You can also perform the following operations on an existing partition:

[Rename a Partition](#)

[Drop a Partition](#)

#### Rename a Partition

Follow the steps below to rename a partition:

**Step 1** Right-click the selected partition and select **Rename Partition**.

**Rename Partition Table** dialog box is displayed prompting you to provide the new name for the partition.

**Step 2** Enter new name and click **OK**.

Data Studio displays the status of the operation in the status bar.

----End

## Drop a Partition

Follow the steps below to drop a partition:

**Step 1** Right-click the selected index and select **Drop Partition**.

**Drop Partition Table** dialog box is displayed.

**Step 2** Click **OK**.

The partition is dropped from the table. Data Studio displays the status of the operation in the status bar.

----End

## 6.6.5 Grant/Revoke Privilege - Regular/Partition Table

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click regular tables group and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Select the objects to grant/revoke privilege from **Object Selection** tab and click **Next**.

**Step 3** Select the role from **Role** drop-down in **Privilege Selection** tab.

**Step 4** Select **Grant/Revoke** in **Privilege Selection** tab.

**Step 5** Select/unselect the required privileges in **Privilege Selection** tab.

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided.

**Step 6** Click **Finish**.

----End

## 6.6.6 Managing Table

### 6.6.6.1 Overview

This section describes how to manage tables effectively.

#### NOTE

- You need to fill all the mandatory parameters, that are marked with asterisk (\*) to complete the operation successfully.
- Refresh is the only operation supported for foreign table.

After creating the table, you can perform operations on the existing table. Right-click the selected table and select the required operation.

## Context Menu

Additional options for table operations are available in the table context menu.

The context menu options available for table operations are:

**Table 6-12** Table Context Menu Options

Option	Description
View Table Data	Opens the table data information. Refer to <a href="#">6.6.7.7 Viewing Table Data</a>
Edit Table Data	Opens the edit table data window. Refer to <a href="#">6.6.7.8 Editing Table Data</a>
Reindex Table	Performs reindexing a table operation. Refer to <a href="#">6.6.6.4 Reindexing a Table</a>
Analyze Table	Analyzes a table. Refer to <a href="#">6.6.6.5 Analyzing a Table</a>
Truncate Table	Truncates table data. Refer to <a href="#">6.6.6.3 Truncating a Table</a>
Vacuum Table	Vacuums table data. Refer to <a href="#">6.6.6.6 Vacuuming a Table</a>
Set table Description	Sets the table description for the table. Refer to <a href="#">6.6.6.7 Setting the Table Description</a>
Set Tablespace	Sets the tablespace for the table. Refer to <a href="#">6.6.6.8 Setting the Tablespace</a>
Set Schema	Sets the schema for the table. Refer to <a href="#">6.6.6.9 Setting the Schema</a>
Export Table Data	Exports the table data. Refer to <a href="#">6.6.7.4 Exporting Table Data</a>
Import Table Data	Imports the table data. Refer to <a href="#">6.6.7.6 Importing Table Data</a>
Show DDL	Shows the DDL of the table. Refer to <a href="#">6.6.7.5 Showing DDL</a>
Export DDL	Exports the DDL of the table. Refer to <a href="#">6.6.7.2 Exporting Table DDL</a>
Export DDL and Data	Exports the DDL and data of the table. Refer to <a href="#">6.6.7.3 Exporting Table DDL and Data</a>
Rename Table	Renames the table. Refer to <a href="#">6.6.6.2 Renaming a Table</a>
Drop Table	Drops (deletes) the table. Refer to <a href="#">6.6.6.10 Dropping a Table</a>
Properties	Displays the properties of the table. Refer to <a href="#">6.6.6.11 Viewing Table Properties</a>

### 6.6.6.2 Renaming a Table

Follow the steps below to rename the table:

**Step 1** Right-click the selected table and select **Rename Table**.

A **Rename Table** dialog box is displayed prompting you to provide the new name.

**Step 2** Enter the table name and click **OK**. You can view the updated table name in the **Object Browser**.

Data Studio displays the status of the operation in the status bar.

 **NOTE**

This operation is not supported for Partition ORC tables.

---End

### 6.6.6.3 Truncating a Table

Follow the steps below to truncate the table:

**Step 1** Right-click the selected table and select **Truncate Table**. This operation deletes the data from an existing table.

Data Studio prompts you to confirm this operation.

**Step 2** In the confirmation dialog box, click **OK** to complete the operation successfully.

A popup message and status bar displays the status of the completed operation.

---End

### 6.6.6.4 Reindexing a Table

Index helps with faster lookup of records. You need to reindex tables in the following scenarios:

- An index is corrupted and no longer contains valid data. Although in theory this must never happen, in practice, indexes can become corrupted due to software bugs or hardware failures. Reindex provides a recovery method.
- An index has become "bloated", that is, it contains many empty or nearly-empty pages. This can occur with B-tree indexes in PostgreSQL under certain uncommon access patterns. Reindex provides a way to reduce the space consumption of the index by writing a new version of the index without the dead pages.
- You have altered a storage parameter (such as fill factor) for an index, and wish to ensure that the change has taken full effect.

Follow the steps below to reindex a table:

**Step 1** Right-click the selected table and select **Reindex Table**.

A popup message and status bar displays the status of the completed operation.



This operation is not supported for Partition ORC tables.

----End

### 6.6.6.5 Analyzing a Table

The analyze table operation gathers statistics about tables and indices of that table and stores the collected information in internal tables of the database where the query optimizer can access the information and use it to help make better query planning choices.

Follow the steps below to analyze a table:

**Step 1** Right-click the selected table and select **Analyze Table**.

The **Analyze Table** message and status bar displays the status of the completed operation.

----End

### 6.6.6.6 Vacuuming a Table

Vacuum table operation reclaims space and makes it available for re-use.

Follow the steps below to vacuum the table:

**Step 1** Right-click the selected table and select **Vacuum Table**.

The **Vacuum Table** message and status bar displays the status of the completed operation.

----End

### 6.6.6.7 Setting the Table Description

Follow the steps below to set the description of the table:

**Step 1** Right-click the selected table and select **Set Table Description**.

An **Update Table Description** dialog box is displayed. It prompts you to provide the table description.

**Step 2** Enter the description and click **OK**.

The status bar displays the status of the completed operation.

**Step 3** To view the table description, right-click selected the table and select **Properties**.

----End

### 6.6.6.8 Setting the Tablespace

Follow the steps below to set the tablespace:

**Step 1** Right-click the selected table and select **Set Tablespace**.

**Set Tablespace** dialog box is displayed that prompts you to select the new tablespace.

**Step 2** Select the tablespace from the drop-down list and click **OK**. The selected table will be moved to the new tablespace. To create a new tablespace, refer to [6.10.2 Working with Tablespaces](#).

The status bar displays the status of the completed operation.

 **NOTE**

- This operation will not be successful if you do not have the required access.
- This operation is not available for Partition table.

----End

### 6.6.6.9 Setting the Schema

Follow the steps below to set the schema:

**Step 1** Right-click the selected table and select **Set Schema**.

**Set Schema** dialog box is displayed that prompts you to select the new schema for the selected table.

**Step 2** Select the schema name from the drop-down list and click **OK**. The selected table will be moved to the new schema.

The status bar displays the status of the completed operation.

 **NOTE**

- This operation is not supported for Partition ORC tables.
- If the required schema contains a table with the same name as the current table, then Data Studio does not allow setting the schema for the table.

----End

### 6.6.6.10 Dropping a Table

Individual or batch drop can be performed on tables. Refer to [6.13.2 Dropping Batch of Objects](#) section for batch drop.

This operation removes the complete table structure (including the table definition and index information) from the database and you have to re-create this table once again to store data.

Follow the steps below to drop the table:

**Step 1** Right-click the selected table and select **Drop Table**.

Data Studio prompts you to confirm this operation.

**Step 2** In the confirmation dialog box, click **OK** to complete the operation successfully.

The status bar displays the status of the completed operation.

----End

### 6.6.6.11 Viewing Table Properties

Follow the steps below to view the properties of the table:

**Step 1** Right-click the selected table and select **Properties**.

Data Studio displays the properties (General, Columns, Constraints, and Index) of the selected table in different tabs.

The following table lists the operations that can be performed on each tab along with edit and refresh of data operation. Edit operation is performed by double-clicking on the cell.

Tab Name	Operations Allowed
General	Save, Cancel, and Copy <b>NOTE</b> Only Table Description field can be modified.
Columns	Add, Delete, Save, Cancel, and Copy
Constraints	Add, Delete, Save, Cancel, and Copy
Index	Add, Delete, Save, Cancel, and Copy

Refer to [6.6.7.8 Editing Table Data](#) section for more information on edit, save, cancel, copy, paste, refresh operations.

**NOTICE**

When viewing table data, Data Studio automatically adjusts the column widths for an optimal table view. Users can resize the columns as needed. If the text contents of a cell exceeds the total available display area, then resizing the cell column may cause DS to become unresponsive.

 **NOTE**

- Individual property window is displayed for each table.
- If the property of a table is modified for the table that is already opened, then refresh and open the properties of the table again to view the updated information on the same opened window.
- If the content of the column have spaces between the words, then word wrap is applied to fit the column within the display area. Word wrap is not applied if the content does not have any spaces between the words.
- The size of the column is determined by the maximum content length column.
- Any change made to the table properties from Object Browser will be reflected after refreshing  the **Properties** tab.
- Paste operation is not allowed in **Data Type** column.

----End

### 6.6.6.12 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click selected regular/partition table and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Refer to [6.6.5 Grant/Revoke Privilege - Regular/Partition Table](#) section to grant/revoke privilege.

----End

## 6.6.7 Managing Table Data

### 6.6.7.1 Overview

This section describes how to manage table data.

### 6.6.7.2 Exporting Table DDL

Follow the steps below to export the table DDL:

**Step 1** In the **Object Browser** pane, right-click the selected table and select **Export DDL**.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** In the **Save As** dialog box, select the location to save the DDL and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click .
- The exported file name will not be the same as table name, if the table name contains characters which are not supported by Windows.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

 **NOTE**

Multiple objects can be selected to export DDL on regular and partition tables. Refer to [Batch Export](#) section for list of objects not supported for export DDL operation.

----End

### 6.6.7.3 Exporting Table DDL and Data

Exporting the table DDL and data exports the following:

- DDL of the table.
- Columns and rows of the table.

Follow the steps below to export the table DDL:

**Step 1** In the **Object Browser** pane, right-click the selected table and select **Export DDL and Data**.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** In the **Save As** dialog box, select the location to save the DDL and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click .
- The exported file name will not be the same as table name, if the table name contains characters which are not supported by Windows.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

 **NOTE**

Multiple objects can be selected to export DDL and data on regular and partition tables. It exports columns, rows, indexes, constraints, and partitions. Refer to [Batch Export](#) section for list of objects not supported for export DDL and Data operation.

----End

## 6.6.7.4 Exporting Table Data

Follow the steps below to export table data:

**Step 1** Right-click the selected table and select **Export Table Data**.

The **Export Table Data** dialog box is displayed with the following options:

- **Format** - Table data can be exported either as excel (xlsx/xls), CSV, Text or binary format. By default Excel (xlsx) is selected.
- **Include Header** - This option is available for CSV and Text files. If selected, it will include the column headers. By default, this option is selected when exporting to CSV or Text file, although it is not a mandatory field. This field will be disabled for excel (xlsx/xls) and binary format.
- **Quotes** - Use this option to define the quote character. You should enter only single byte character for this field. Quote character should not be same as delimiter. For CSV and Text format, by default this field is enabled, although it is not a mandatory field. This field will be disabled for excel (xlsx/xls) and binary format.
  - If table data value has delimiter in their values, then it will use the character mentioned in this field.
  - If the Quote character is present in value, then that character will be escaped with same quoted character.
  - If result value has multiline values, then it will be quoted with quoted character.
- **Escape** - Use this option to define the escape value. You should enter only single byte character for this field. Escape value should not be same as quote character. For CSV and Text format, by default this field is enabled, although it is not a mandatory field. This field will be disabled for excel (xlsx/xls) and binary format.
- **Replace NULL with** - Use this option to replace null value in the table with string. New line or carriage return characters are non-acceptable values for this field. Maximum of 100 characters can only be entered in this field. This field value must be different from delimiter and quote values. For CSV and Text format, by default this field is enabled, although it is not a mandatory field. This field will be disabled for excel (xlsx/xls) and binary format.
- **Encoding** - The **Encoding** field will be pre-populated with the encoding selection made in **Preferences > Session Setting** tab. This is not a mandatory field.
- **Delimiter** - Use this option to define delimiter. You can select the available delimiter or mention customized delimiter in the **Other** field in the Delimiter section. For CSV and Text format ",", will be the default delimiter. Maximum of 10 bytes can only be entered in the **Other** field. For CSV and Text format, by default this field is enabled, although it is not a mandatory field. This field will be disabled for excel (xlsx/xls) and binary format. It is mandatory to enter a value when **Other** field is selected.
- **All Columns** - Use this option to quick select all columns. By default this is checked. To manually select columns, uncheck this and select columns from the **Available Columns** list.
  - **Available Columns** - You can use this column to select specific columns to export.
  - **Selected Columns** - This field displays the selected columns that will be exported. The columns can be re-ordered. By default all columns display in this field.

 **NOTE**

Refer to [Column/Row Size](#) in FAQ section for row and column size supported by xlsx and xls.

- **File Name** - Use this option to specify the name to save the exported file. By default, the table name is displayed in this field.

 **NOTE**

File name follows Windows file naming convention.

- **Output Path** - Use this option to select the location to save the exported file. The **Output Path** field is auto-populated with the selected path.
- **Security Disclaimer** - The security disclaimer is mentioned in this section, and you should read and agree, to continue with the export operation.
  - **I Agree** - By default this field is selected. You cannot proceed further if this field is not checked.
  - **Do not show again** - You can select this field to hide the Security Disclaimer for subsequent export table data operation for current logged instance of Data Studio.

 **NOTE**

- String, double, date, calendar, and boolean datatype will be stored as is in excel. All other datatypes will be converted into string and stored in excel.
- For excel export if the cell size is beyond 32767, then exported cell data will be truncated.

**Step 2** Complete the required fields and click **OK**.

The **Save As** dialog box is displayed.

**Step 3** Click **Save** to save the exported data in the selected format. The status bar displays the progress of the operation.

The **Data Exported Successfully** dialog box and status bar displays the status of the completed operation.

 **NOTE**

- If the disk is full while exporting the table, then Data Studio displays an I/O error. Perform the following operations to resolve this error:
  1. Click **OK** to close the connection profile.
  2. Clean the disk.
  3. Re-establish the connection and export the table data.
- The exported file name will not be the same as table name, if the table name contains characters which are not supported by Windows.

----End

## Canceling the export table data operation

Follow the steps below to cancel the export table data operation:

**Step 1** Double-click the status bar to open the **Progress View** tab.

**Step 2** In the **Progress View** tab, click .

**Step 3** In the **Cancel Operation** dialog box, click **Yes**.

The **Messages** tab and status bar displays the status of the cancelled operation.

----End

### 6.6.7.5 Showing DDL

Follow the steps below to show DDL query of the table:

**Step 1** Right-click the selected table and select **Show DDL**.

The DDL of the selected table is displayed.

 **NOTE**

- A new terminal is opened each time the **Show DDL** operation is executed.
- Microsoft Visual C runtime file (msvcr100.dll) is required to complete this operation. Refer to [Troubleshooting](#) section for more information.

Database Encoding	File Encoding	Supports Show DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

---End

### 6.6.7.6 Importing Table Data

Prerequisites to import table data are:

- If the source import file does not match with the destination import table definition, then you must modify the properties of the destination table in the **Import Table Data** dialog box. Additional columns will be inserted with default value.
- You should know the export properties of the file that you are importing like delimiter, quote, and escape character and so on. Export properties saved during export operation cannot be changed while importing the file.

Follow the steps below to import table data:

**Step 1** Right-click the selected table and select **Import Table Data**.

Data Studio displays the **Import Table Data** dialog box with the following options:

- **Import Data File** - This field displays the file path of the imported file. Use the **Browse** button to select different file.
- **Format** - Table data can be imported as CSV, Text or binary format. By default CSV is selected.

- **Include Header** - Use this option if the import file has column header. For CSV and Text format, by default this field is selected, although it is not a mandatory field. This field will be disabled for binary format.
- **Quotes** - You should enter only single byte character for this field. Quote character should not be same as delimiter and null parameter. For CSV and Text format, by default this field is selected, although it is not a mandatory field. This field will be disabled for binary format.
- **Escape** - You should enter only single byte character for this field. If escape value is same as quote value, then escape value will be replaced with '\0'. For CSV and Text format, by default this field is selected with value as double quotation mark, although it is not a mandatory field. This field will be disabled for binary format.
- **Replace with Null** - You can use this field to replace null value in the table with string. The same null string used while exporting should be used while importing data and this need to be explicitly mentioned. For CSV and Text format, by default this field is selected, although it is not a mandatory field. This field will be disabled for binary format.
- **Encoding** - The **Encoding** field will be pre-populated with the encoding selection made in **Preferences > Session Setting** tab. This is not a mandatory field.
- **Delimiter** - You can select the available delimiter or mention customized delimiter in the **Other** field in the Delimiter section. For CSV and Text format "," will be the default delimiter. This field value should not be same as Quote and Replace Null with field values. For CSV and Text format, by default this field is selected, although it is not a mandatory field. This field will be disabled for binary format. It is mandatory to enter a value when **Other** field is selected.
- **All Columns** - Use this option to quick select all columns. By default this field is selected. To manually select columns, uncheck this and unselect columns from the **Selected Columns** list.
  - **Available Columns** - You can use this column to select specific columns to import.
  - **Selected Columns** - This field displays the selected columns that will be imported. By default all columns display in this field.

**Step 2** Click the **Browse** button from the **Import Data File** field.

The **Open** dialog box is displayed.

**Step 3** In the **Open** dialog box, select the file to import and click **Open**.

**Step 4** Complete the required fields and click **OK**.

The status bar displays the progress of the operation. The imported data is appended to the existing table data.

The **Data Imported Successfully** dialog box and status bar displays the status of the completed operation.

---End

## Canceling the import table data operation

Follow the steps below to cancel the import table data operation:

**Step 1** Double-click the status bar to open the **Progress View** tab.

**Step 2** In the **Progress View** tab, click .

**Step 3** In the **Cancel Operation** dialog box, click **Yes**.

The **Messages** tab and status bar displays the status of the canceled operation.

----End

### 6.6.7.7 Viewing Table Data

Follow the steps to view table data:

**Step 1** Right-click the selected table and select **View Table Data**.

The **View Table Data** tab is displayed where you can view the table data information.

Toolbar menu in the **View Table Data** window:

Toolbar Name	Toolbar Icon	Description
Copy		Click the icon to copy selected content from <b>View Table Data</b> window to clipboard. Shortcut key - <b>Ctrl+C</b> .
Advanced Copy		Click the icon to copy content from result window to clipboard. Results can be copied to include the row number and/or column header. Refer to <b>View Query Results</b> to set this preference. Shortcut key - <b>Ctrl+Shift+C</b> .
Show/Hide Search bar		Click the icon to display/hide the search text field. This is a toggle button.
Encoding	-	Refer to <b>Execute SQL Queries</b> section for information on encoding selection.

Icons in Search field:

Icon Name	Icon	Description
Search		Click the icon to search the table data displayed based on the criteria defined. Search text are case insensitive.
Clear Search Text		Click the icon to clear the search text entered in the search field.

Refer to **Execute SQL Queries** section for column reordering and sort option.

- **Query Submit Time** - Provides the query submitted time.
- Number of rows fetched with execution time is displayed. The default number of rows is displayed. If there are additional rows to be fetched, then it will be denoted with the word "more". You can scroll to the bottom of the table to fetch and display all rows.

### NOTICE

- When viewing table data, Data Studio automatically adjusts the column widths for an optimal table view. Users can resize the columns as needed. If the text contents of a cell exceeds the total available display area, then resizing the cell column may cause DS to become unresponsive.
- When the data in a table cell is more than 1000 characters, it will appear trimmed upto 1000 characters with "..." at the end.
  - If the user copies the data from a cell in a table or Result tab and pastes it on any editor (such as SQL terminal/PLSQL source editor, notepad or any other external editor application), the entire data is pasted.
  - If the user copies the data from a cell in a table or Result tab and pastes it on an editable cell (same or different), the cell shows only the first 1000 characters with "..." in the end.
  - When the table/Result tab data is exported, the exported file contains the whole data.

### NOTE

- Individual table data window is displayed for each table.
- If the data of the table that is already opened is modified, then refresh and open the table data again to view the updated information on the same opened window.
- While the data is loading a message displays at the bottom stating "fetching".
- If the content of the column have spaces between the words, then word wrap is applied to fit the column within the display area. Word wrap is not applied if the content does not have any spaces between the words.
- Select part of cell content and press **Ctrl+C** or click  to copy selected text from a cell.
- The size of the column is determined by the maximum content length column.
- You can save preference to define:
  - Number of records to be fetched.
  - Column width
  - Copy option from result set.Refer to [Query Results](#) for more information.

----End

## 6.6.7.8 Editing Table Data

Follow the steps below to edit table data:

**Step 1** Right-click the selected table and select **Edit Table Data**.

The **Edit Table data** tab is displayed.

Refer to [6.6.7.7 Viewing Table Data](#) section for description on copy and search toolbar options.

----End

Data Studio validates only the following data types entered into cells:

Bigint, bit, boolean, char, date, decimal, double, float, integer, numeric, real, smallint, time, time with time zone, time stamp, time stamp with time zone, tinyint, and varchar.

Editing of array type data type is not supported.

Any related errors during this operation reported by database will be displayed in Data Studio. Time with time zone and timestamp with time zone columns are non-editable columns.

You can perform the following operations in the **Edit Table Data** tab:

- **Insert**
- **Delete**
- **Update**
- **Copy**
- **Paste**

## Insert

Follow the steps below to insert a row:

- Step 1** Click  to insert a row.
- Step 2** Double-click the cell to modify and enter the required details in the row.
- Step 3** Click  to save changes.

The **Edit Table Data** tab status bar shows the **Query Submit Time**, **Number of rows fetched**, **Execution time** and **Status** of the operation.

---

### NOTICE

Data Studio updates rows identified by the unique key. If a unique key is not identified for a table and there are identical rows, then an update operation made on one of the rows will affect all identical rows. Refresh the **Edit Table Data** tab to view the updated rows.

---

### NOTE

- Changes to cells in a row that are not saved are highlighted in green. Once saved the color resets to default color.
- Unsaved records are highlighted in red. The number of successful and failed records are displayed in the status bar of the **Edit Table Data** tab.
- Clicking **Save** either saves all the valid changes or does not save anything if there are invalid changes. Refer to **Edit Table Data** to set the behavior of save operation.

- Step 4** Click  to roll back the changes that are not saved.
- Step 5** Set the preference to define:
- Number of records to be fetched
  - Column width

- Copy option from result set  
Refer to [Query Results](#) for more information.

----End

Data Studio allows you to edit the distribution key column only for a new row.

## Delete

Follow the steps to delete a row:

**Step 1** Click the row header of the row to be deleted.

**Step 2** Click  to delete a row.

**Step 3** Click  to save changes.

Define unique key dialog box is displayed.

**Step 4** Click the required option:

- **Use All Columns**  
Click **Use All Columns** to define all columns as unique key.
- **Custom Unique Key**
  - a. Click **Custom Unique Key** to define selected columns as unique key.
  - b. **Define Unique Key** dialogue box is displayed.
  - c. Select the required columns and click **OK**.
- **Cancel**

Click **Cancel** to modify the information in **Edit Table Data** tab.

The **Edit Table Data** tab status bar shows the **Query Submit Time**, **Number of rows fetched**, **Execution time** and **Status** of the operation.

Select **Remember the selection for this window** option to hide the unique definition

window from displaying while continuing with the edit table data operation. Click  from **Edit Table Data** toolbar to clear previously selected unique key definition and display unique definition window again.

### NOTE

- Deleted rows that are not saved are highlighted in red. Once saved the color resets to default color.
- Unsaved records are highlighted in red. The number of successful and failed records are displayed in the status bar of the **Edit Table Data** tab.
- Clicking **Save** either saves all the valid changes or does not save anything if there are invalid changes. Refer to [Edit Table Data](#) to set the behavior of save operation.

**Step 5** Click  to roll back the changes that are not saved.

**Step 6** Refresh the table data to view deleted duplicate rows.

----End

## Update

Follow the steps to update cell data:

**Step 1** Double-click the cell to update the contents of the cell.

**Step 2** Click  to save changes.

Define unique key dialog box is displayed.

**Step 3** Click the required option:

- **Use All Columns**  
Click **Use All Columns** to define all columns as unique key.
- **Custom Unique Key**
  - a. Click **Custom Unique Key** to define selected columns as unique key.
  - b. **Define Unique Key** dialogue box is displayed.
  - c. Select the required columns and click **OK**.
- **Cancel**

Click **Cancel** to modify the information in **Edit Table Data** tab.

The status bar shows the **Execution Time** and **Status** of the operation.

Select **Remember the selection for this window** option to hide the unique definition

window from displaying while continuing with the edit table data operation. Click  from **Edit Table Data** toolbar to clear previously selected unique key definition and display unique definition window again.

### **NOTE**

- Changes to cells in a row that are not saved is highlighted in green. Once the record is saved the color resets to default color.
- Unsaved records are highlighted in red. The number of successful and failed records are displayed in the status bar of the **Edit Table Data** tab.
- Clicking **Save** either saves all the valid changes or does not save anything if there are invalid changes. Refer to [Edit Table Data](#) to set the behavior of save operation.

**Step 4** Click  to roll back the changes that are not saved.

**Step 5** Refresh the table data to view deleted duplicate rows.

---End

During edit operation, Data Studio does not allow you to edit the distribution key column as it is used by the DB to locate data in the database cluster.

## Copy

You can copy data from the **Edit Table Data** tab.

Follow the steps below to copy data:

**Step 1** Select the cell(s) and click  (Copy) or  (Advanced Copy).

Refer to [Execute SQL Queries](#) section to understand the difference between copy and advanced copy.

 **NOTE**

- Data can be copied to include the row number and/or column header. Refer to [Query Results](#) to set this preference.
- Select part of cell content and press **Ctrl+C** or click  to copy selected text from a cell.

----End

## Paste

You can copy data from a CSV file and paste it into cells in the **Edit Table Data** tab to insert and update records. If you paste onto existing cell data, the data is overwritten with the new data from the CSV file.

Follow the steps below to paste data into a cell:

**Step 1** Copy data from CSV file.

**Step 2** Select the cell(s) and click .

**Step 3** Click  to save changes.

Define unique key dialogue box is displayed.

**Step 4** Click the required option:

- **Use All Columns**  
Click **Use All Columns** to define all columns as unique key.
- **Custom Unique Key**
  - a. Click **Custom Unique Key** to define selected columns as unique key.
  - b. **Define Unique Key** dialogue box is displayed.
  - c. Select the required columns and click **OK**.
- **Cancel**

Click **Cancel** to modify the information in **Edit Table Data** tab.

The status bar shows the **Execution Time** and **Status** of the operation.

Select **Remember the selection for this window** option to hide the unique definition

window from displaying while continuing with the edit table data operation. Click  from **Edit Table Data** toolbar to clear previously selected unique key definition and display unique definition window again

 **NOTE**

- The number of copied cells from CSV must match the number of cells selected in the Edit Table Data tab to paste the data.
- Use the  to roll back the changes that are not saved.
- Changes to cells in a row that are not saved is highlighted in green. Once saved the color resets to default color.
- Failed unsaved records are highlighted in red. The number of successful and failed records are displayed in the status bar of the **Edit Table Data** tab.
- Clicking **Save** either saves all the valid changes or does not save anything if there are invalid changes. Refer to [Edit Table Data](#) to set the behavior of save operation.

**----End**

During paste operation, Data Studio does not allow you to edit the distribution key column as it is used by the DB to locate data in the database cluster.

 **NOTE**

Empty cells are shown as [NULL]. Empty cell in **Edit Table Data** tab can be searched using the **Null Values** search drop-down.

Refer to [Execute SQL Queries](#) section for information on show/hide search bar, sort, column reorder, and encoding options.

## 6.6.8 Editing Temporary Tables

Data Studio allows you to edit temporary tables. Temporary tables are deleted automatically when you close the connection that was used to create the table.

**NOTICE**

Ensure that connection reuse is enabled when you use the SQL Terminal to edit temporary tables. Refer to [6.12.12 Managing SQL Terminal Connections](#) for information about enabling SQL Terminal Connection reuse.

Follow the steps to edit a temporary table:

**Step 1** Execute a query on the temporary table.

The **Result** tab displays the results of the SQL query along with the query statement executed.

**Step 2** Edit the temporary table from the **Result** tab. Refer to the [Execute SQL Queries](#) section for information on editing the resultset.

**----End**

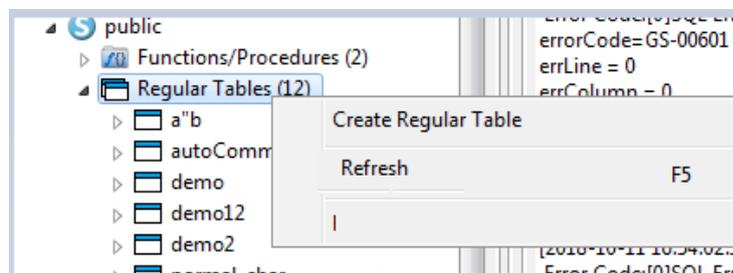
## 6.7 Tables(GaussDB 100)

### Creating Table

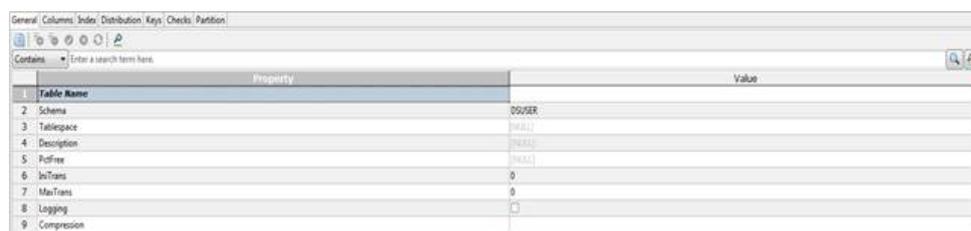
Data Studio allows the user to create the regular table and partition table from object browser in GaussDB 100.

Follow the steps to create table:

**Step 1** Right click Regular Tables on Object browser. Following menu is displayed:



**Step 2** Click **Create Regular Table**. Following window is displayed :



**Step 3** Click **General**. Provide **Value** based on **Property**.

**Table Name**- Provide the table name.

**Schema**- You are not able to change the value as it is already chosen.

**Tablespace**- Choose the tablespace where you want to create the table.

**Description**-Description of the function of the table.

**PctFree**-The proportion of the smallest space that must be preserved.

**IniTrans**- Initial transaction count.

**MaxTrans**- Enter the maximum value for the transaction count.

**Logging**- Check the box if you want to log.

**Compression**- There are two options: 1. **FOR ALL OPERATIONS** 2. **FOR DIRECT\_LOAD OPERATIONS**.

**Step 4** Click **Columns** tab.

**Step 5** Click on .Provide the value for the attributes.

**Name**- Provide the name of the column.

**Data Type**-Choose the data type from dialog box.

**Is Nullable**- Whether you can make the column's value null.

**Default**- The default value of a column.

**Is Default Expression**- This field is relevant only when **Default** field has some value. Check the box means **Default** value is displayed as an expression given. On the other hand unchecking the box means **Default** value is displayed based on the function.

**Comment**-Description of a column.

1. Click **Index**.

**Step 6** Click on  .Provide the value for the attributes.

**Name-** Provide the name of the index.

**Columns-** Choose the Column from dialog box, where datatype of the column is displayed.

**Is Unique-** Wheather this index is unique or not.

**IniTrans-** Initial transaction count.

**Tablespace-** Chosse the tablespace you want

**Step 7** Click **Distribution** tab.

**Step 8** Click on  .Provide the value for the attributes.

**Type-** Provide the type of the distribution from dropdown.

**Columns-** Select the columns from available columns in the dialog, where datatype of the column is displayed.

**GroupID-**Enter the Group ID.

**Values-** Enter the index value.

**Step 9** Click **Keys** tab.

**Step 10** Click on  .Provide the value for the attributes.

**Name-** Provide the name of the key.

**Type-** Seclect the type of the key from the dropdown.

**Columns-** Select the columns from the dropdown where datatype of the column is displayed in the dialog box.

**Enabled-** It refers if the key enabled or not.

**Referencing table-** Select the table you want to refer from the dialog box. This option will only be available when you choose **FOREIGN KEY** in **Type**.

**Referencing column-** Select the column you want to refer.This option will only be available when you choose **FOREIGN KEY** in **Type**.

**On delete -** There are two options here. **1. Cascade and 2.Not Null**. This option will only be available when you choose **FOREIGN KEY** in **Type**. **Deferrable-** Check the box to make it deferrable.

**Deferred-** Check the box to make it deferred

**Validated-** Check the box to validate existing data on the table when a constraint is created.

**Index-**Enter the index value.

**Step 11** Click **Checks** tab.

**Step 12** Click on  .Provide the value of the attributes.

**Name-**Provide the name .

**Condition**-Provide the check expression.

**Enabled**- Check the box to enable it.

**Deferrable**- Check the box to make it deferrable.

**Validated**- Check the box to validate existing data on the table when a constraint is created.

**Step 13** Click **Partition** tab.

**Step 14** Click on  . Provide the value of the attributes.

**Partition Type**-Select the partition type from the drop down.

**partition Keys**- Provide the partition key.

**Partition Name**-Enter the partition name.

**Partition Value**- Provide the partition value.

**Tablespace**-Select the tablespace

**IniTrans**-Initial transaction count.

**PctFree**-The proportion of the smallest space that must be preserved.

**Step 15** Click **DDL** tab. You can view the SQL query automatically generated for the inputs provided.

----End

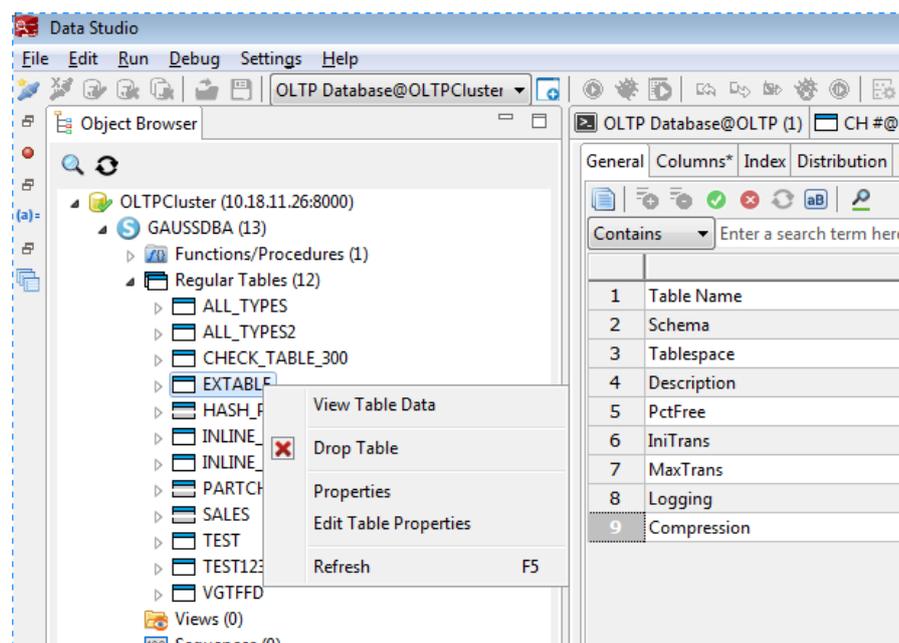
## Editing Table Properties

Data Studio allows the user to edit the structure of the database objects, especially the tables after it is created.

Follow the steps to edit the property of a table:

**Step 1** Select the table you want to edit.

**Step 2** Right Click on the table. Following menu options are displayed.



**Step 3** Click **Edit Table Properties**. A window is displayed.

**General** - Following are the behaviors of the buttons for this tab:

- **Add**  - This button is disabled in the General tab.
- **Delete**  - This button is disabled in the General tab.
- **Save**  - Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.
- **Cancel**  - This option allows you to cancel all the modifications and resets the tabs.
- **Refresh**  - Click the refresh button to update the property information.

Folloing are the editable fields of this tab:

- Editable Tab- **Table Name, Description,PctFree.**

**Columns** -Following are the behaviors of the buttons for this tab:

- **Add**  - Click the button to add new columns.
- **Save**  - Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.
- **Delete**  - On clicking the row header, this button will be enabled. Clicking on the delete button the row will be marked for deletion. After you click save, the column will be dropped.
- **Cancel**  - Cancel the modification done in the columns tab by selecting Cancel button.
- **Refresh**  -Click the refresh button to update the columns information.

Folloing are the editable fields of this tab:

- Editable Tab- **Column Name, Data type, Comments.**

**Index** -Following are the behaviors of the buttons for this tab:

- **Add**  -Click on add to add a new index.
- **Save**  - Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.
- **Delete**  - On clicking the row header, this button will be enabled. Clicking on the delete button the row will be marked for deletion. After you click save, the column will be dropped.
- **Cancel**  -Cancel the modification done in the index tab by selecting cancel button.

- **Refresh**  -Click the refresh button to update the index information.

This tab does not support modification of the existing fields..

**Distribution**-This tab does not support addition and editing of fields.

**Keys**- Following are the behaviors of the buttons for this tab:

- **Add**  - Click add button to add a new constraint.
- **Save**  -Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.
- **Delete**  -On clicking the row header, this button will be enabled. Clicking on the delete button the row will be marked for deletion. After you click save, the column will be dropped.
- **Cancel**  -Cancel the modification done in the keys tab by selecting cancel button.
- **Refresh**  - Click the refresh button to update chnages.

Folloing are the editable of this

- Editable Tab- **Enabled** (Only for Foreign Key), **Validated**.

**Checks**-Following are the behaviors of the buttons for this tab:

- **Add**  -Click add button to add new information.
- **Save**  - Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.
- **Delete**  - On clicking the row header, this button will be enabled. Clicking on the delete button the row will be marked for deletion. After you click save, the column will be dropped.
- **Cancel**  - Cancel the modification done in the Checks tab by selecting cancel button.
- **Refresh**  -Click the refresh button to update the Checks tab information

Folloing are the editable fields of this

Editable Tab- **Enabled, Validated**.

**Partition** -This tab does not support the editing of existing fields.

- **Add**  - Click add button a new row will be added and user will be able to fill the fields related to partition.
- **Save**  - Once you modify anything in editable tab the Save button will be enabled, the user has to select this option to save the modifications.

- **Delete** - On clicking the row header, this button will be enabled. Clicking on the delete button the row will be marked for deletion. After you click save, the column will be dropped.
- **Cancel**  -Cancel the modification done in the Partition tab by selecting cancel button.
- **Refresh**  - Click the refresh button to update the Partition tab information.

----End

#### NOTE

- Cancelling an ongoing task for alter queries is not supported from Terminal/Edit table properties in cluster environment.
- In a cluster environment, if you cancel the ongoing task for multiple create table queries and in that case if it throws an error message displaying: errMsg="current operation canceled"

sql=create table <table name><columns>distributed by<function>

Then, there is a possible reason that table with the same name is already created in few nodes of the same cluster. So performing operation on the same table might throw an error. In this case it is recommended to drop the table and create the table again to proceed further.

## 6.8 Sequences

### 6.8.1 Creating Sequence

Follow the steps below to create a sequence:

**Step 1** In the **Object Browser** pane, right-click **Sequences** under the particular schema where you want to create the sequence and select **Create Sequence**.

The **Create New Sequence** dialog box is displayed.

**Step 2** Provide information to create a sequence:

1. Enter a name in **Sequence Name** field.

#### NOTE

Select **Case** check box to retain the capitalization of the text entered in **Sequence Name** field. For example, if the sequence name entered is "Employee", then the sequence name is created as "Employee".

2. Enter minimum value in **Minimum Value** field.
3. Enter the increase step value in **Increment By** field.
4. Enter maximum value in **Maximum Value** field.

#### NOTE

Minimum and Maximum value should be between -9223372036854775808 and 9223372036854775807.

5. Enter start value of the sequence in **Start Value** field.
6. Enter cache information in **Cache** field. The cache value denotes the number of sequences stored in the memory for quick access.
7. Select **Cycle** field to recycle sequences after the number of sequences reaches either the maximum or minimum value.



The schema name auto-populates in the **Schema** field.

8. Select table from **Table** drop-down.
9. Select column from **Column** drop-down.

**Step 3** Click **Finish**.

The status bar displays the status of the completed operation.



In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided.

----End

## 6.8.2 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click sequences group and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Select the objects to grant/revoke privilege from **Object Selection** tab and click **Next**.

**Step 3** Select the role from **Role** drop-down in **Privilege Selection** tab.

**Step 4** Select **Grant/Revoke** in **Privilege Selection** tab.

**Step 5** Select/unselect the required privileges in **Privilege Selection** tab.

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided.

**Step 6** Click **Finish**.

----End

## 6.8.3 Working with Sequences

You can perform the following operations on an existing sequence:

- [Grant/Revoke Privilege](#)
- [Dropping a Sequence](#)
- [Dropping a Sequence Cascade](#)

### Dropping a Sequence

Individual or batch drop can be performed on sequences. Refer to [6.13.2 Dropping Batch of Objects](#) section for batch drop.

Follow the steps below to drop a sequence:

**Step 1** Right-click the selected sequence and select **Drop Sequence**.

The **Drop Sequence** dialog box is displayed.

**Step 2** Click **Yes** to drop the sequence.

The status bar displays the status of the completed operation.

----End

## Dropping a Sequence Cascade

Follow the steps below to drop the sequence cascade:

**Step 1** Right-click the selected sequence and select **Drop Sequence Cascade**.

The **Drop Sequence Cascade** dialog box is displayed.

**Step 2** Click **Yes** to drop the sequence cascade.

The status bar displays the status of the completed operation.

----End



This is only available for OLAP, not for OLTP.

## Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click selected sequence and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Refer to [6.8.2 Grant/Revoke Privilege](#) section to grant/revoke privilege.

----End

## 6.9 Views

### 6.9.1 Creating a View

Follow the steps below to create a new view:

**Step 1** Right-click **Views** and select **Create View**.

The DDL template for the view is displayed in the SQL Terminal tab.

**Step 2** Edit the DDL as required.

**Step 3** Click  to execute the DDL.

**Step 4** Press **F5** to refresh the **Object Browser**.

You can view the new view in the **Object Browser**.



The status bar will not display message on completion of this operation.

----End

## 6.9.2 Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click views group and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Select the objects to grant/revoke privilege from **Object Selection** tab and click **Next**.

**Step 3** Select the role from **Role** drop-down in **Privilege Selection** tab.

**Step 4** Select **Grant/Revoke** in **Privilege Selection** tab.

**Step 5** Select/unselect the required privileges in **Privilege Selection** tab.

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided.

**Step 6** Click **Finish**.

---End

## 6.9.3 Working with Views

Views can be created to restrict access to specific rows or columns of a table. A view can be created from one or more tables and is determined by the query used to create the view.

You can perform the following operations on an existing view:

- [Exporting the View DDL](#)
- [Dropping a View](#)
- [Dropping a View Cascade](#)
- [Renaming a View](#)
- [Setting the Schema for a View](#)
- [Viewing the Show DDL](#)
- [Setting the Default Value for the View Column](#)
- [Viewing the Properties of a View](#)
- [Grant/Revoke Privilege](#)

### Exporting the View DDL

Follow the steps below to export view DDL:

**Step 1** Right-click the selected view and select **Export DDL**.

**Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** In the **Save As** dialog box, select the location to save the DDL and click **Save**. The status bar displays the progress of the operation.

 **NOTE**

- To cancel the export operation, double-click the status to open the **Progress View** tab and click .
- The exported file name will not be the same as view name, if the view name contains characters which are not supported by Windows.
- Multiple objects can be selected to export the view DDL. Refer to **Batch Export** section for list of objects not supported for export view DDL operation.

The **Export** message and status bar displays the status of the completed operation.

Database Encoding	File Encoding	Supports Exporting DDL
UTF-8	UTF-8	Yes
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

----End

## Dropping a View

Individual or batch drop can be performed on views. Refer to **6.13.2 Dropping Batch of Objects** section for batch drop.

Follow the steps below to drop the view:

**Step 1** Right-click the selected view and select **Drop View**.

The **Drop View** dialog box is displayed.

**Step 2** Click **Yes** to drop the view.

The status bar displays the status of the completed operation.

----End

## Dropping a View Cascade

Follow the steps below to drop the view and its dependent database objects:

**Step 1** Right-click the selected view and select **Drop View Cascade**.

The **Drop View** dialog box is displayed.

**Step 2** Click **Yes** to drop the view and its dependent database objects.

The status bar displays the status of the completed operation.

----End

## Renaming a View

Follow the steps below to rename the view:

**Step 1** Right-click the selected view and select **Rename View**.

The **Rename View** dialog box is displayed.

**Step 2** Enter the required name for the view and click **OK**. You can view the renamed view in the **Object Browser**.

The status bar displays the status of the completed operation.

----End

## Setting the Schema for a View

Follow the steps below to set the schema for the view:

**Step 1** Right-click the selected view and select **Set Schema**.

The **Set Schema** dialog box is displayed.

**Step 2** Select the required schema from the drop-down list and click **OK**.

The status bar displays the status of the completed operation.

If the required schema contains a view with the same name as the current view, then Data Studio does not allow setting the schema for the view.

----End

## Viewing the Show DDL

Follow the steps below to view the DDL of the view:

**Step 1** Right-click the selected view and select **Show DDL**.

The DDL is displayed in a new **SQL Terminal** tab. You must refresh the **Object Browser** to view the latest DDL.

----End

## Setting the Default Value for the View Column

Follow the steps below to set the default value for a column in the view:

**Step 1** Right-click the selected column name under the view and select **Set View Column Default Value**.

A dialog box with the current default value (if it is set) is displayed which prompts you to provide the default value.

**Step 2** Enter the value and click **OK**.

Data Studio displays the status of the operation in the status bar.

----End

## Viewing the Properties of a View

Follow the steps below to view the properties of the View:

**Step 1** Right-click the selected View and select **Properties**.

The properties (General and Columns) of the selected View is displayed in different tabs.

### NOTE

If the property of a View is modified that is already opened, then refresh and open the properties of the View again to view the updated information on the same opened window.

----End

## Grant/Revoke Privilege

Follow the steps below to grant/revoke privilege:

**Step 1** Right-click selected view and select **Grant/Revoke**.

The **Grant/Revoke** dialog is displayed.

**Step 2** Refer to [6.9.2 Grant/Revoke Privilege](#) section to grant/revoke privilege.

----End

## 6.10 Tablespaces

### 6.10.1 Creating a Tablespace

You can create tablespaces to optimize performance of database objects.

After creating a tablespace, it will be available in the **Create New table** wizard. For more information, refer to [6.6.2 Creating Regular Table](#).

Follow the steps below to create a new tablespace for GaussDB 200, GaussDB 300, and DWS database:

**Step 1** Right-click **Tablespaces** and select **Create Tablespace**. The **Create Tablespace** dialog box is displayed.

**Step 2** Enter the following information to create the tablespace:

### NOTE

You need to fill all the mandatory parameters that are marked with asterisk (\*) to complete the operation successfully.

- **Name:** Enter the name of the tablespace. For example, `New_Tablespace`.
- **Location:** Enter the path to store the tablespace on the database. For example, `\home\user1\`  
One path is limited to only one tablespace. Access permissions to the path must be set by the user.
- **Unlimited Size:** Select this check box to set unlimited maximum size of the tablespace.  
 **NOTE**  
Once the **Unlimited Size** field is checked, the **Max Size** field becomes non-editable.
- **Max Size:** Enter the maximum size of the tablespace. The supported range is 1 KB - 9,007,199,254,740,991 KB. The only acceptable value in this field is positive whole number.
- Select an option from the **Max Size** drop-down list. The options available are:
  - **KB:** Specifies the **Max Size** in kilobytes.
  - **MB:** Specifies the **Max Size** in megabytes.
  - **GB:** Specifies the **Max Size** in gigabytes.
  - **TB:** Specifies the **Max Size** in terabytes.
  - **PB:** Specifies the **Max Size** in petabytes.
- **File System:** Select the type of file system.
  - Select **General** to create the tabular type of tablespace.
  - Select **HDFS** to create the Hadoop Distributed File System (HDFS) type of tablespace.
- **Address:** Enter the NameNode IP address (IPv4) of the HDFS cluster and port number of the active and standby nodes.  
For example, `xx.xx.xx.xx:xxxx`
- **HDFS Configuration File Path:** Enter the path of the HDFS cluster configuration file. This is a mandatory field for HDFS file system.
- **HDFS Data Store Path:** Enter the HDFS data storage path. This is a mandatory field for HDFS file system.
- **Sequential Page Cost:** Sets the optimizer's estimated cost of a disk page fetch that is part of a series of sequential fetches. Enter the sequential read page overhead. The supported range is 0 - 1.79769e+308 (double byte). The default value is 1 which is also the recommended value. The acceptable values are either positive whole number or positive decimals with one decimal point.
- **Random Page Cost:** Sets the optimizer's estimated cost of a non-sequentially-fetched disk page. Enter the random read page overhead. The supported range is 0 - 1.79769e+308 (double byte). The default value is 4 which is also the recommended value. The acceptable values are either positive whole number or positive decimals with one decimal point.

**NOTICE**

Although the server allows to set the value of **Random Page Cost** to less than that of **Sequential Page Cost**, it is not physically sensitive to do so. However, setting them equal makes sense if the database is entirely cached in RAM, because in that case there is no penalty for fetching pages out of sequence. Also, in a heavily-cached database you must lower both values relative to the CPU parameters, since the cost of fetching a page already in RAM is much smaller than it would normally be.

 **NOTE**

If the server does not support HDFS file system, then Address, HDFS Configuration File Path, and HDFS Data Store Path, fields becomes non-editable.

**Step 3** Click **OK**. You can view the new tablespace in the **Object Browser**.+

The status bar displays the status of the completed operation.

 **NOTE**

When OK is clicked, the Run in Background option is enabled. This option helps in continuing with other operations on the database while the tablespace is being created in the background. Once the tablespace is created a pop-up message is displayed with success or failure notification. Alternatively clicking  or pressing Esc runs the tablespace creation operation in background.

---End

Follow the steps below to create a new tablespace for GaussDB 100 database:

**Step 1** Right-click **Tablespaces** and select **Create Tablespace**. The **Create Tablespace** tab box is displayed.

**Step 2** Enter the following information in **General** tab:

- **Tablespace Name:** Enter the name of the tablespace. For example, New\_Tablespace.

**Step 3** Enter the following information in **Data File** tab.

- **File Name:** Enter the path of the new data file in the operating system and the file name of the new data file. If the file name is a relative path, then the file is saved in the data directory by default.
- **File Size:** Enter the size of the data file.
- **Autoextend:** true/false. By default, this parameter is set to false.

**Autoextend - true**

- You can manually specify the size of each extension.
- You can set the following attributes:
  - **Autoextend Size:** Specifies the size of the automatic extension. If this parameter is not specified, the default value 16MB is used.
  - **Maxsize:** Specifies the maximum size the file can be automatically expanded. If you do not specify or specify UNLIMITED, it takes the value as 8TB. If the upper limit is specified, then the upper limit cannot be greater than 8TB. If the upper limit and **Autoextend Size** is specified, then the specified upper limit cannot be lesser than the value in **Autoextend Size** and **File Size**.

**Autoextend - false**

If this field is set to false, then the system does not automatically extend the value.

----End

## 6.10.2 Working with Tablespaces

You can perform the following operations on an existing tablespace:

- [Adding a Data File](#)
- [Renaming a Tablespace](#)
- [Renaming a Data File](#)
- [Setting Tablespace Options](#)
- [Setting Maximum Tablespace Size](#)
- [Dropping a Tablespace](#)
- [Dropping a Data File](#)
- [Viewing the Tablespace DDL](#)
- [Viewing/Editing Tablespace Properties](#)

### Adding a Data File

Follow the steps below to add a data file for GaussDB 100 database:

**Step 1** Right-click the selected tablespace and select **Properties**.

The properties of the tablespace is displayed.

**Step 2** Add the required fields. Refer to [6.10.1 Creating a Tablespace](#) section for different field definition.

**Step 3** Click  to save changes.

----End

### Renaming a Tablespace

Follow the steps below to rename a tablespace for GaussDB 200, GaussDB300, DWS database:

**Step 1** Right-click the selected tablespace and select **Rename Tablespace**.

The **Rename Tablespace** dialog box is displayed.

**Step 2** Enter the new name for the tablespace and click **OK**.

The status bar displays the status of the completed operation.

#### NOTE

System tablespaces are displayed in the **Object Browser** by default.

You can view the renamed tablespace in the **Object Browser**.

----End

Follow the steps below to rename a tablespace for GaussDB 100 database:

**Step 1** Right-click the selected tablespace and select **Properties**.

The properties of the tablespace is displayed.

**Step 2** In the **General** tab, enter the new name for the tablespace in **Tablespace Name**.

**Step 3** Click  to save changes.

You can view the renamed tablespace in the **Object Browser**.

----End

## Renaming a Data File

Follow the steps below to rename a data file for GaussDB 100 database:

**Step 1** Right-click the selected tablespace and select **Properties**.

The properties of the tablespace is displayed.

**Step 2** In the **Data File** tab, enter the new name for the data file in **File Name**.

**Step 3** Click  to save changes.

----End

## Setting Tablespace Options

Follow the steps below to set tablespace options:

**Step 1** Right-click the selected tablespace and select **Set Tablespace Option**.

The **Set Tablespace Option** dialog box is displayed.

**Step 2** Enter the **Random Page Cost** and **Sequential Page Cost** and click **OK**.

The status bar displays the status of the completed operation.

----End

## Setting Maximum Tablespace Size

Follow the steps below to set the maximum size of the tablespace:

**Step 1** Right-click the selected tablespace and select **Set Tablespace MaxSize**.

The **Set Tablespace Maxsize** dialog box is displayed.

**Step 2** To set an unlimited value for the maximum size of the tablespace, select the **Unlimited Size** check box.

To set a custom value for the maximum size of the tablespace, enter the maximum size of the tablespace in the **Max Size** text box. The supported range is 1 KB - 9,007,199,254,740,991 KB. Select an option from the **Max Size** drop-down list. The options available are:

- **KB**: Specifies the **Max Size** in kilobytes.
- **MB**: Specifies the **Max Size** in megabytes.

- **GB**: Specifies the **Max Size** in gigabytes.
- **TB**: Specifies the **Max Size** in terabytes.
- **PB**: Specifies the **Max Size** in petabytes.

**Step 3** Click **OK**.

The status bar displays the status of the completed operation.

----End

## Dropping a Tablespace

Follow the steps below to drop the tablespace for GaussDB 200, GaussDB300, DWS database:

### **NOTE**

- Data Studio does not allow dropping of system tablespace(s).
- Data studio does not allow batch drop of tablespaces.

**Step 1** Right-click the selected tablespace and select **Drop Tablespace**.

The **Drop Tablespace** dialog box is displayed.

**Step 2** Click **OK** to drop the tablespace.

The status bar displays the status of the completed operation.

----End

## Dropping a Data File

Follow the steps below to drop a data file for GaussDB 100 database:

**Step 1** Right-click the selected tablespace and select **Properties**.

The properties of the tablespace is displayed.

**Step 2** Select the data file to be dropped and click  to delete a row.

**Step 3** Click  to save changes.

----End

## Viewing the Tablespace DDL

Follow the steps below to view the DDL of the tablespace:

**Step 1** Right-click the selected tablespace and select **Show DDL**.

The tablespace DDL is displayed in a new **SQL Terminal** tab with the terminal name format as 'tablespacename@servername'. You must refresh the **Object Browser** to view the latest DDL.

----End

## Viewing/Editing Tablespace Properties

Follow the steps below to view tablespace properties for GaussDB 100 database:

**Step 1** Right-click the selected tablespace and select **Properties**.

The properties of the tablespace is displayed.

**Step 2** Modify the required fields. Refer to [6.10.1 Creating a Tablespace](#) section for different field definition.

**Step 3** Click  to save changes.

----End

## 6.11 Users/Roles

### 6.11.1 Create User/Role

A database is used by many users, and the users are grouped for management convenience. A database role can be one or a group of database users.

Users and roles have similar concepts in databases. In practice, you are advised to use a role to manage permissions rather than to access databases.

**Users** - They are set of database users. These users are different from operating system users. These users can assign privileges to other users to access database objects.

**Role** - This can be considered as a user or group based on the usage. Roles are at cluster level, and hence applicable to all databases in the cluster.

Follow the steps below to create user/role:

**Step 1** Right-click **Users/Roles** and select **Create User/Role**. The **Create User/Role** dialog box is displayed.

**Step 2** Provide basic table information such as name, password, and privileges. For more details, refer to [Providing General Information](#).

**Step 3** Define advanced information such as connect limit, validity, resource pool, role, and administrator group. For more details, refer to [Defining Advanced Options](#).

In **SQL Preview** tab, you can view the SQL query automatically generated for the inputs provided. For more details, refer to [SQL Preview](#).

----End

## Providing General Information

Provide the following information to create a user/role:

**Step 1** Enter the user/role name in the **Name** field. It specifies the name of the user/role to be created.

- Step 2** Enter password for user/role in **Password** field.
- Step 3** Reconfirm the password in **Re-enter Password** field.
- Step 4** Select the privileges to set for user/role from **Privileges** section.
- Step 5** Click **Next** to define the advanced columns information for user/role.

----End

## Defining Advanced Options

Provide the following information to define advanced options:

- Step 1** Input the number of concurrent connections the role can make in the Connection Limit field.  
Set the connection to -1 for the unlimited connections.
- Step 2** Select a date and time when the role's password becomes valid Start Date field. If this field is not filled, then the password will be valid all time.
- Step 3** Select a date and time after which the role's password is no longer valid. If this field is not filled, then the password will be valid all time.
- Step 4** Select the resource pool from Resource Pool drop-down.
- Step 5** Select the role group from Role Group drop-down.
- Step 6** Select the administrator group from Administrator Group drop-down.
- Step 7** Click Next.

----End

## SQL Preview

Data Studio generates a DDL statement based on the inputs provided in **Create User/Role** wizard.

You can only view, select, and copy the query. You cannot edit the query.

- To select all queries, press **Ctrl+A** or right-click and select **Select All**.
- To copy the selected query, press **Ctrl+C** or right-click and select **Copy**.

Click **Finish** to create the user/role. On clicking the **Finish** button, the generated query will be sent to the server. Any errors are displayed in the dialog box and status bar.



- Indicates user/role has login permission.



- Indicates user/role does not have login permission.

## 6.11.2 Working with Users/Roles

You can perform the following operations on an existing user/role:

- **Dropping a User/Role**

- [Viewing/Editing User/Role Properties](#)
- [Viewing the User/Role DDL](#)

## Dropping a User/Role

Follow the steps below to drop a user/role:

**Step 1** Right-click the selected user/role and select **Drop User/Role**.

The **Drop User/Role** dialog box is displayed.

**Step 2** Click **Yes** to drop the user/role.

The status bar displays the status of the completed operation.

----End

## Viewing/Editing User/Role Properties

Follow the steps below to view the properties of user/role:

**Step 1** Right-click the selected user/role and select **Properties**.

Data Studio displays the properties (General, Privilege, and Membership) of the selected user/role in different tabs.

Editing of properties can be performed. OID is non-editable field.

Refer to [6.6.7.8 Editing Table Data](#) section for information on edit, save, cancel, copy, and refresh operations.

----End

## Viewing the User/Role DDL

Follow the steps below to view the DDL of the user/role:

**Step 1** Right-click the selected user/role and select **Show DDL**.

The user/role DDL is displayed in a new **SQL Terminal** tab. You must refresh the **Object Browser** to view the latest DDL.

----End

# 6.12 SQL Terminal

## 6.12.1 Opening SQL Terminals

You can open multiple **SQL Terminal** tabs in Data Studio. You can use this feature to work with SQL queries when the current **SQL Terminal** is executing a query.

Follow the steps below to open a new SQL Terminal:

You can also open multiple SQL terminals on different connection profiles.

**Step 1** In the **Object Browser** pane, right-click the selected database and select **Open Terminal** or click  on the toolbar or Ctrl+T shortcut key to open new SQL terminal.

The new **SQL Terminal** tab is displayed.

----End

 **NOTE**

- Data Studio supports a maximum of 100 SQL terminals and tabs in total. Each **SQL Terminal** will have multiple **Result** and one **Messages** tab based on the number of times a query is executed. If the connection with the database is lost, then the corresponding **SQL Terminals** are not disabled.
- Restoring individual SQL Terminal or tabs is not possible. The restore operation restores the complete set of minimized SQL Terminals and tabs.
- Data Studio resets the numbering counter of SQL Terminal after all terminals are closed.
- Data Studio resets the numbering counter of **Resultset** after all the tabs are closed.
- Data Studio resets the numbering counter for **show DDL Tablespace**, **show DDL Users/Roles**, **Batch Drop** tab, **Result** tab and **Execution Plan** tab.

Errors and warnings are displayed which do not have accompanying results in the status bar. Results of successful executions are displayed in the **Result** tab.

Follow the steps below to open a new SQL Terminal on a different connection profile:

**Step 1** On the toolbar, select the required connection from the connection profile drop-down list.

**Step 2** In the **Object Browser** pane, right-click the selected database in the connection profile and select **Open Terminal** or click  on the toolbar. The new **SQL Terminal** tab is displayed.

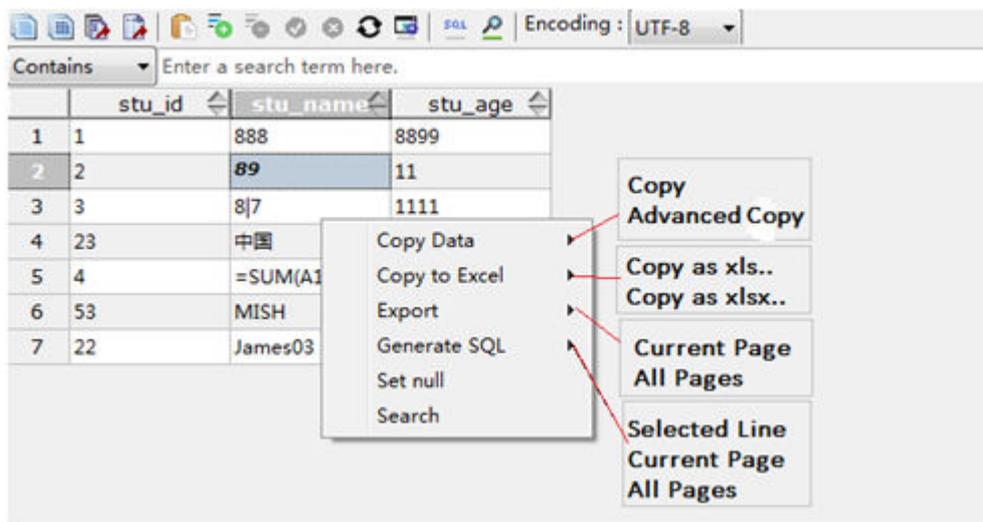
The new **SQL Terminal** tab is named as `<database_name>@<connection_profile>(<tab_number>)`. For example, **postgres@IDG\_1(2)**. The tab number is updated for each new **SQL Terminal** tab of the connection profile.

----End

### Managing Right Click Option On Result Window

This feature allows to copy , export cell data to excel files and generate SQL files of queries as well.

Right click on the result window after the result of the SQL query is shown. Right Click Menu is displayed as following:



Follow The Steps For Including Row Number And Column Header in Result Set:

- Step 1** Click **Settings** on Menu bar of Data Studio.
- Step 2** Select **Preferences**.
- Step 3** Expand **Result Management** and select **Query Results**.
- Step 4** Under **Result Advanced Copy** option check **Include column header** and **Include row number** boxes.

----End

Feature description of the of the menu is as following:

**Table 6-13** Right Click Menu

Menu Option	Sub Menu	Description
Copy Data	Copy	Copies the selected cell data.
	Advanced Copy	Copies the selected cell data with row number and column header as per the preference setting.
Copy to Excel	Copy as xls	Export the selected cell data in xls format. Maximum 64k Rows and 256 columns can be exported.
	Copy as xlsx	Export the selected cell data in xlsx format. Maximum 1M Rows can be exported.
Export	Current Page	Exports the table data of the current page.
	All Pages	Exports the entire table content.

Menu Option	Sub Menu	Description
Generate SQL	Selected Line	Generate the SQL file of the selected data in logical insert statement.
	Current Page	Generate a SQL file of the current page data in logical insert statement.
	All Pages	Generate a SQL file of entire table data in logical insert statement.
Set Null	-	Sets a cell data to null.
Search	-	Searches the selected cell data and displays all the data that matches the search condition.

 **NOTE**

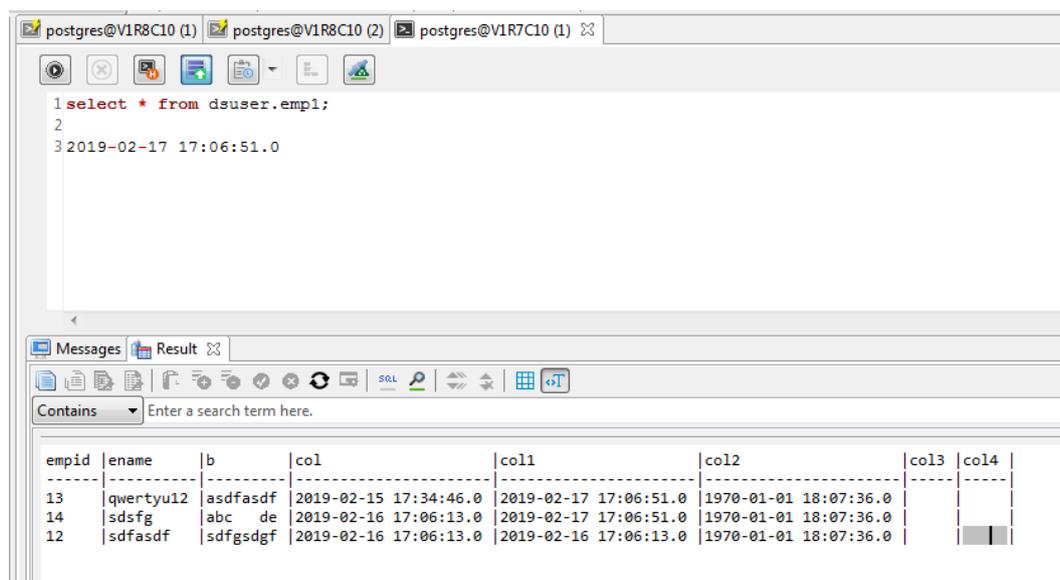
Generated SQLs are not valid for result sets derived from queries with JOINS, Expressions, Views, Set operators, Aggregate functions, GROUP BY clause and column aliases.

## Viewing Text Mode In Result Set Tab

This new feature in Data Studio enables you to view the data in text mode in resultset tab.

Apart from having the grid view, the text mode view provides you with two features: copy and search.

**Step 1** Click  icon to obtain the result in text mode.



----End

 NOTE

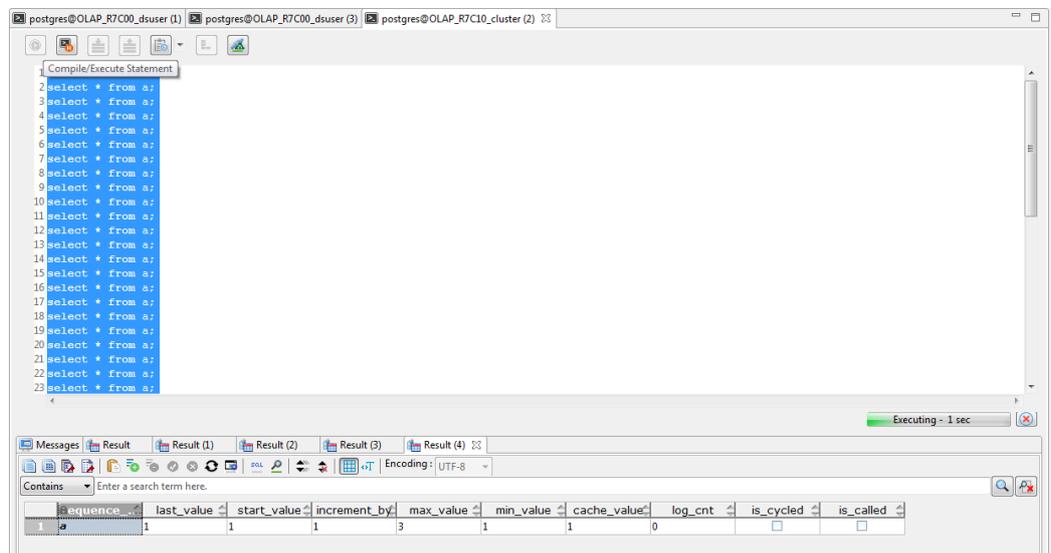
Selecting multiple cell data and searching may show some incorrect results in text mode as all the information is copied in plain text to the search window.

## Displaying Execution Progress Bar

When a query is being executed from SQL Terminal, A progress bar is shown associated with particular terminal with dynamic elapsed time . The progress bar disappears as the query execution finishes. The time information alongside the bar displays the duration of the query execution on completion.

An option to cancel the query execution is available alongside the progress bar if required.

Refer to the below image:

 NOTE

- The Cancel button now has been removed from the toolbar.
- Execution Progress Bar is also shown in compiling/debugging of function/procedures in PL/SQL editor .
- The time format shown in the progress bar will be as: w hrs x min y sec z ms
- For batch execution in SQL Terminal, the progress bar is shown with total elapsed time on completion.

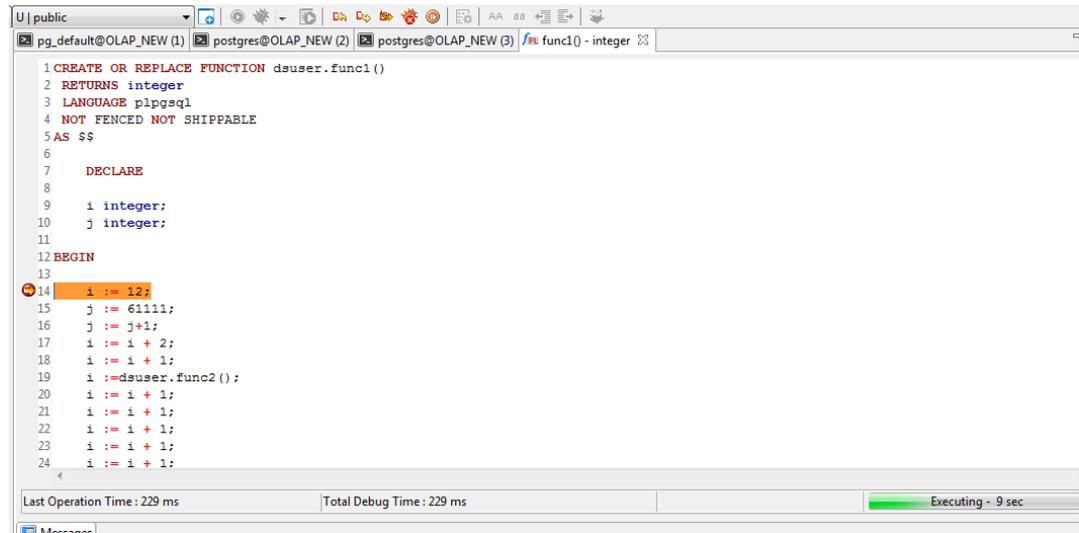
## Debug Time Consumption

Data Studio displays a status bar during debugging, that shows the **Last Operation Time** consumed for executing statement in each debug operation and also the **Total Debug Time**.

While Debugging, based on the each operation performed , the **Last Operation Time** and **Total Debug Time** provided in terminal status bar gets updated. **Total Debug Time** is the cumulative sum of **Last Operation Time**.

This provides an easy way to find out huge time consuming statement of debug object.

Refer to the following picture:



```
1 CREATE OR REPLACE FUNCTION dsuser.func1 ()
2 RETURNS integer
3 LANGUAGE plpgsql
4 NOT FENCED NOT SHIPPABLE
5 AS $$
6
7 DECLARE
8
9 i integer;
10 j integer;
11
12 BEGIN
13
14 i := 12;
15 j := 61111;
16 j := j+1;
17 i := i + 2;
18 i := i + 1;
19 i :=dsuser.func2 ();
20 i := i + 1;
21 i := i + 1;
22 i := i + 1;
23 i := i + 1;
24 i := i + 1;
```

Last Operation Time : 229 ms      Total Debug Time : 229 ms      Executing - 9 sec

**NOTE**

- Debug is applicable for both functions and procedures.
- Its only applicable for OLAP.

## 6.12.2 Managing SQL Query Execution History

Data Studio allows viewing and managing frequently used SQL queries. The history of executed SQL queries is maintained only for the **SQL Terminal**.

Follow the steps below to view the SQL history:

**Step 1** Click  in the **SQL Terminal** tab.

The **SQL History** dialog box is displayed.

----**End**

**NOTE**

SQL history scripts are not encrypted.

The number of queries saved in the **SQL History** dialog box is based on the value defined in **Preferences > Editor > SQL History** pane. Refer to the **SQL History** section to modify the SQL History count. Data Studio overwrites the older queries into the SQL history after the list is full. The executed query is automatically stored in the list.

The **SQL History** dialog box has the following columns:

- **Pin Status** - Displays the pinned status of the queries. Pinned queries will always show on the top and it will not be deleted from the history even when the list is full.
- **SQL Statement** - Displays the SQL query. The number of characters for an SQL query displayed in the **SQL Statement** column is based on the number defined in **Preferences > Editor > SQL History** pane. Refer to the **SQL History** section to modify the number of characters for a query.
- **Number of Records** - Displays the amount of records fetched by the SQL query.
- **Start Time** - Displays the time the query execution was started.
- **Execution Time** - Displays the time taken to execute the query.

- **Database Name** - Displays the name of the database.
- **Execution Status** - Displays the execution status of the query as **Success** or **Failure**.

Deleting the connection profile deletes the history. If the **SQL History** dialog box is closed, the query is not removed from the list.

You can perform the following operations in the **SQL History** dialog box:

- **Loading an SQL query into the SQL terminal**
- **Loading multiple SQL queries into the SQL terminal**
- **Deleting an SQL query**
- **Deleting all SQL queries**
- **Pinning an SQL query**
- **Unpinning an SQL query**

## Loading an SQL query into the SQL terminal

Follow the steps below to load the SQL query into the SQL terminal:

**Step 1** Select the required query and click .

The query is appended to the cursor position in the **SQL Terminal**.

---End

## Loading multiple SQL queries into the SQL terminal

The **Load in SQL Terminal and close History** button loads selected queries into the **SQL Terminal** and closes the **SQL History** dialog box.

Follow the steps below to load selected SQL queries into the SQL terminal:

**Step 1** Select the required queries.

**Step 2** Click .

The queries are appended to the cursor position in the **SQL Terminal**.

---End

### **NOTE**

If you continue the execution on error, then each statement in the terminal will be running as a scheduled job and runs one after the other. The execution status is updated in the console and job is listed in the progress bar. When the time difference between Job Execution, Progress Bar Update and Console Update is very minimal, you will not be able to open the progress bar and stop the execution. In such scenarios you have to close the SQL Terminal to come out of execution.

## Deleting an SQL query

Follow the steps below to delete a SQL query from the SQL History list:

**Step 1** Select the required query and click .

A confirmation pop up window is displayed.

**Step 2** Click **OK**.

----End

## Deleting all SQL queries

Follow the steps below to delete all SQL queries from the SQL History list:

**Step 1** Click .

A confirmation pop up window is displayed.

**Step 2** Click **OK**.

----End

## Pinning an SQL query

You can pin queries that you do not want Data Studio to delete automatically from the **SQL History**. You can pin a maximum of 50 queries. Pinned queries are displayed at the top of the list. The value set in SQL History count does not affect the pinned queries. Refer to [SQL History](#) section for additional information on SQL History count.

### NOTE

- The pinned queries appear on top once the **SQL History** window is closed and re-opened.

Follow the steps below to pin a SQL query:

**Step 1** Select the required SQL query and click .

The **Pin Status** column displays the pinned status of the query.

----End

## Unpinning an SQL query

Follow the steps below to unpin a SQL query:

**Step 1** Select the required SQL query and click .

The **Pin Status** column displays the unpinned status of the query.

----End

## 6.12.3 Opening And Saving SQL Scripts

### Opening an SQL Script

Follow the steps below to open an SQL script:

**Step 1** Choose **File > Open** from the main menu. Alternatively, click **Open** on the toolbar or right-click the **SQL Terminal** and select **Open**.

If the SQL Terminal has existing content, then there will be an option to override the existing content or append content to it.

**Step 2** The **Open** dialog box is displayed.

**Step 3** In **Open** dialog box, select the SQL file to import and click **Open**.

The selected SQL script is opened as a **File Terminal**.

Icon is different. On mouse over the source file and corresponding database connection will be displayed on File Terminal.

----End



- The encoding type of the SQL file must match the encoding type specified in [preferences](#).
- Label of the file terminal will start with \* if any of its content is edited. Dirty flag is removed once the file terminal is saved.
- File Terminals cannot be renamed, one terminal is always mapped to one Source Script File, but one script can be opened in multiple terminals.
- You can open SQL scripts only on SQL Terminals.

Data Studio allows you to save and open SQL scripts in the **SQL Terminal**. After saving the changes, SQL Terminal will be changed to a File Terminal.

## Saving an SQL Script

Save option saves the File Terminal content to the associated file. ,

Follow the steps below to save an SQL script:

**Step 1** Choose **File > Save** from the main menu. Alternatively press "Ctrl + s" to save the SQL terminal content. Alternatively, click **Save** on the toolbar or right-click the **SQL Terminal** and select **Save**.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

Data Studio displays the status of the operation in the status bar.



- The script is saved as an SQL file. Data Studio sets the read/write permission for the saved SQL file. To ensure security, you must set the read/write permissions for folders.
- When a change is made in a file and if that associated file is unavailable, it will trigger *Save As* option.
  - In any case, if saving of the source file is failed due to some reason, then user is prompted with *Save As* option to save the content as a new source file. If you choose not to save (i.e. cancel *Save As*), then File Terminal gets converted into an SQL Terminal.
  - The changes made to File Terminals are not Auto Saved.

----End

## Saving an SQL Script in New File

Save As option saves the terminal content to a new file.

Follow the steps below to save an SQL script:

**Step 1** Choose **File > Save As** from the main menu. Alternatively click "ctrl +Alt+ s" key to save SQL Terminal or File Terminal content in new file.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 2** Click **OK**.

The **Save As** dialog box is displayed.

**Step 3** Select the location to save the script and click **Save**.

----End

 **NOTE**

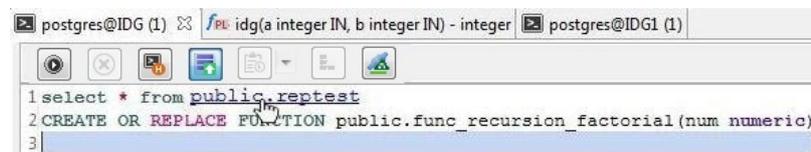
When there are unsaved changes in File Terminals , then user will be given an option to save or cancel on graceful exit of data studio.

## 6.12.4 Viewing Object Properties in the SQL Terminal

Data Studio allows you to view table properties and functions/procedures.

Follow the steps below to view table properties:

**Step 1** Press **Ctrl** and point to the table name.



**Step 2** Click the highlighted table name. The properties of the selected table is displayed.

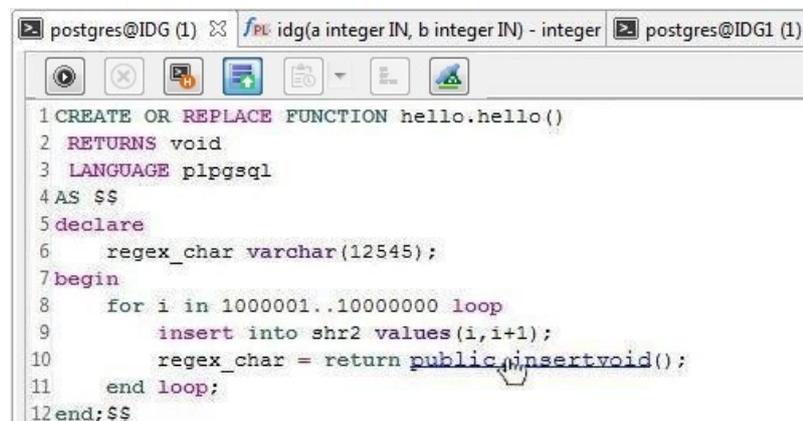
 **NOTE**

The table properties are read-only.

----End

Follow the steps below to view functions/procedures:

**Step 1** Press **Ctrl** and point to the function/procedure name.



**Step 2** Click the highlighted function/procedure name. The function/procedure is displayed in a new **PL/SQL Viewer** tab.

----End

Follow the steps below to view the properties of a View:

**Step 1** Press **Ctrl** and point to the view name.

**Step 2** Click the highlighted view name. The properties of the selected view is displayed.

----End

## Saving a Terminal Content Before Exiting Application

Data Studio allows you to save the unsaved content of the the terminal before exiting the application.

Follow the steps below to save the content of the terminal.

**Step 1** Click on close button of the application. **Exit Application** dialog box will appear.

**Step 2** Click **Graceful Exit**.

1. **Saving File Terminal** dialog box appears.Unsaved dirty file terminal is displayed.

**Step 3** Select the terminal to save.

**Step 4** Click **OK**.

----End

 **NOTE**

- Saving File Terminal dialog box will not appear in case of Force Exit.

## 6.12.5 Canceling Execution of SQL Queries

Data Studio allows you to cancel execution of an SQL query executing in the **SQL Terminal**.

Follow the steps below to cancel execution of an SQL query:

**Step 1** Execute the SQL query in the **SQL Terminal**.

**Step 2** Click  in the **SQL Terminal** or press **Shift+Esc**.

Alternatively, you can choose **Run > Cancel** from the main menu or right-click in the **SQL Terminal** and select **Cancel**, or from **Progress View** tab select **Cancel**.

----End

When you cancel the query, the execution stops at the currently executing SQL statement.

Database changes made by the canceled query are rolled back and the queries following the canceled query are not executed.

A query is not canceled and the **Result** tab shows the result when:

1. The server has finished execution of the query and is preparing the result.

2. The result of the executed query is being transferred from the server to the Data Studio client.

A query cannot be canceled while viewing the query **Execution Plan**. For more details, refer to [6.12.8 Viewing the Query Execution Plan and Cost](#).

The **Messages** tab shows the query cancelation message.

 **NOTE**

The **Cancel** button is enabled only during query execution.

## 6.12.6 Formatting of SQL Queries

Data Studio supports formatting and highlighting of SQL queries and PL/SQL statements.

### PL/SQL Formatting

Follow the steps below to format PL/SQL statements:

**Step 1** Select the PL/SQL statement to be formatted.

**Step 2** Click  on the toolbar to format the query.

Alternatively, use the key combination **Ctrl+Shift+F** or choose **Edit > Format** from the main menu.

The PL/SQL statements are formatted.

----**End**

### SQL Formatting

Data Studio supports formatting of simple SQL SELECT, INSERT, UPDATE, DELETE statements which are syntactically correct. Following are some of the statements for which formatting is supported.

1. The SELECT statement must be made of following clauses:

- Target list
- From clause (includes join)
- Where clause
- Group by clause
- Having clause
- Order by clause
- Common table expression

SELECT statement without SET operations like UNION, UNION ALL, MINUS, INTERSECT and so on.

SELECT statements without sub-queries.

2. The INSERT statement is made of below clauses only

- Insert Into Table name
- Values clause
- Values Column List

- RETURNING
- 3. The UPDATE statement is made of below clauses only
  - Update Table name
  - SET Clause
  - From Clause(Including Join)
  - Where Clause
  - RETURNING
- 4. The DELETE statement is made of below clauses only
  - Delete From Table name
  - Using Clause(Including Join)
  - Where Clause
  - RETURNING

Follow the steps below to format SQL queries:

**Step 1** Select the SQL query statements to be formatted.

**Step 2** Click  on the toolbar to format the query.

Alternatively, use the key combination **Ctrl+Shift+F** or choose **Edit > Format** from the main menu.

The query is formatted.

Refer following table for query formatting rules.

**Table 6-14** Query Formatting Rules

Statement	Clauses	Format Rules
SELECT	SELECT list	Line break before first column
		Indent column list
	FROM	Line break before FROM
		Line break after FROM
		Indent FROM list
		Stack FROM list
	JOIN (FROM clause)	Line break before JOIN
		Line break after JOIN
		Line break before ON
		Line break after ON
		Indent table after JOIN
		Indent ON condition

Statement	Clauses	Format Rules
	WHERE	Line break before WHERE
		Line break after WHERE
		Indent WHERE condition
		Place WHERE condition on single line
	GROUP BY	Line break before GROUP
		Line break before GROUP BY expression
		Indent column list
		Stack column list
	HAVING	Line break before HAVING
		Line break after HAVING
		Indent HAVING condition
	ORDER BY	Line break before ORDER
		Line break after BY
		Indent column list
		Stack column list
	CTE	Indent subquery braces
		Each CTE in a new line
INSERT	INSERT INFO	Line break before opening brace
		Line break after opening brace
		Line break before closing brace
		Indent column list braces
		Indent column list
		Line break before VALUES
		Stack column list
		Line break before VALUES
		Line break before opening brace
		Line break after opening brace
		Line break before closing brace
		Indent VALUES expressions list braces
		Indent VALUES expressions list

Statement	Clauses	Format Rules	
		Stack VALUES expressions list	
	DEFAULT	Line break before DEFAULT	
		Indent DEFAULT keyword	
	CTE	Each CTE in a new line	
	RETURNING	Line break before RETURNING	
		Line break after RETURNING	
		Indent RETURNING column list	
		Place RETURNING column List on single line	
	UPDATE	UPDATE Table	Line break before table
			Indent table
SET Clause		Line break before SET	
		Indent column assignments list	
		Indent column assignments list	
FROM CLAUSE		Line break before FROM	
		Line break after FROM	
		Indent FROM list	
		Stack FROM list	
JOIN CLAUSE(FROM CLAUSE)		Line break before JOIN	
		Line break after JOIN	
		Line break before ON	
		Line break after ON	
		Indent table after JOIN	
		Indent ON condition	
WHERE CLAUSE		Line break before WHERE	
		Line break after WHERE	
		Indent WHERE condition	
		Indent WHERE condition	
CTE		Each CTE in a new line	
RETURNING	Line break before RETURNING		

Statement	Clauses	Format Rules
DELETE	USING CLAUSE	Line break after RETURNING
		Indent RETURNING column list
		Line break before FROM
		Line break after FROM
		Indent USING list
		Stack FROM list
	JOIN CLAUSE	Line break before JOIN
		Line break after JOIN
		Line break before ON
		Line break after ON
		Indent table after JOIN
		Indent ON condition List
	WHERE CLAUSE	Line break before WHERE
		Line break after WHERE
		Indent WHERE condition
		Stack WHERE condition list
	CTE	Each CTE in a new line
	RETURNING	Line break before RETURNING
		Line break after RETURNING
		Indent RETURNING column list

**---End**

Data Studio supports automatic highlighting of following punctuation mark's pair when cursor is placed before or after the punctuation mark or selection of the punctuation mark.

- Brackets - ( )
- Square brackets - [ ]
- Braces - { }
- Single-quoted string literals - ''
- Double-quoted string literals - " "

Follow the steps below to change case for SQL queries and PL/SQL statements:

Text case can be changed in the SQL Terminal using one of the following methods:

**Method 1:**

**Step 1** Select the text, choose **Edit > Upper Case/Lower Case**.

The text changes to the case selected.

----End

**Method 2:**

**Step 1** Select the text, choose **AA** or **aa** from toolbar.

The text changes to the case selected.

----End

**Method 3:**

**Step 1** Select the text, press **Ctrl+Shift+U** to change to upper case or **Ctrl+Shift+L** to change to lower case.

The text changes to the case selected.

----End

## SQL Highlight

Keywords are highlighted automatically as you enter them (according to the default color scheme) as shown below:

```

1 CREATE OR REPLACE FUNCTION pg_catalog.login_audit_messages(flag boolean)
2 RETURNS TABLE(username text, database text, logintime timestamp with time zone, type text, result text, client_conninfo text)
3 LANGUAGE plpgsql
4 SECURITY DEFINER
5 AS $$
6 DECLARE
7 user_name text;
8 db_name text;
9 success_time1 timestamp with time zone;
10 success_time2 timestamp with time zone;
11 success_count integer;
12 sql_stmt VARCHAR(4096);
13 BEGIN
14 SELECT SESSION_USER INTO user_name;
15 SELECT CURRENT_DATABASE() INTO db_name;
16 IF flag = true THEN
17 --get the last login success info.
18 sql_stmt := 'SELECT username,database,time,type,result,client_conninfo FROM pg_query_audit(''1970-1-1'', ''9999-12-31'')
19
20 --get the count of success login before.
21 EXECUTE 'SELECT count(*) FROM pg_query_audit(''1970-1-1'', ''9999-12-31'') WHERE type = ''login_success'' AND client_con
22
23 --if success count is more than 1, calculate the failed login record between success_time1 and success_time2.
24 IF success_count > 1 THEN
25 --get the last 1 login success info.
26 EXECUTE 'SELECT time FROM pg_query_audit(''1970-1-1'', ''9999-12-31'') WHERE type = ''login_success'' AND client_con

```

The following figure shows the default color scheme for the specified type of syntax:

Syntax Category	Color Value	HEX Value	Shade
DEFAULT	RGB(0,0,0)	000000	
UNRESERVED_KEYWORD	RGB(198,0,134)	C60086	
TYPE	RGB(64,0,200)	4000C8	
PREDICATES	RGB(224,5,11)	E0050B	
RESERVED	RGB(35,90,80)	235A50	
CONSTANTS	RGB(35,90,80)	235A50	
SINGLE_LINE_COMMENT	RGB(64,128,128)	408080	
STRING	RGB(0,0,255)	0000FF	

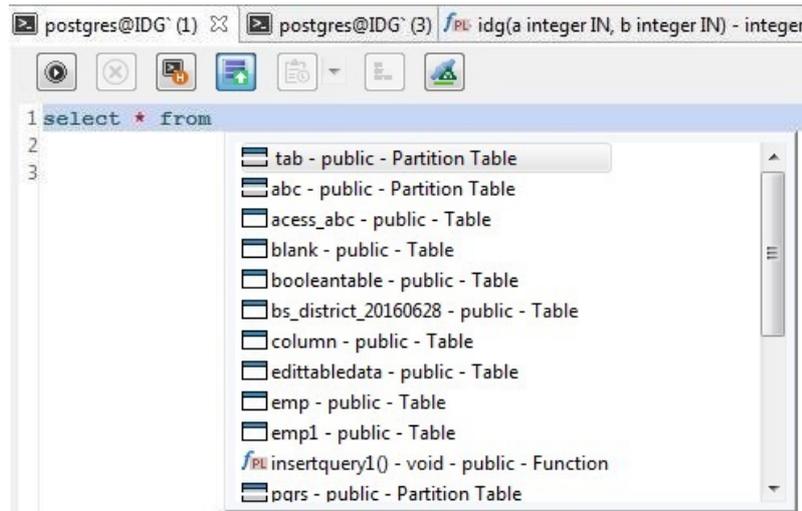
Refer to [Syntax Coloring](#) to customize the SQL highlight color scheme for the specific type of syntax.

## 6.12.7 Selecting a DB Object in the SQL Terminal

Data Studio suggests a list of possible schema names, table names and column names, and views in the **SQL Terminal**.

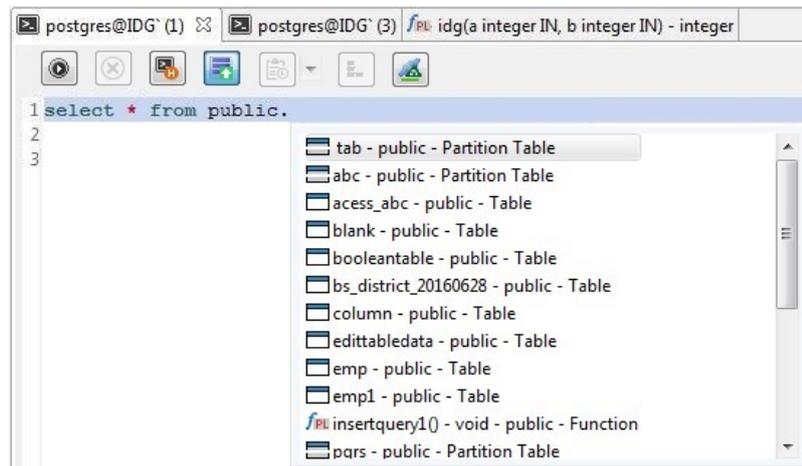
Follow the steps below to select a DB object:

- Step 1** Press **Ctrl+Space** and enter the required parent DB object name. The DB objects list is refined as you continue typing the DB object name. The DB objects list displays all DB objects of the database connected to the **SQL Terminal**.



- Step 2** To select the parent DB object, use the **Up** or **Down** arrow keys and press **Enter** on the keyboard, or double-click the parent DB object.

- Step 3** Enter **.** (period) to list all child DB objects.



- Step 4** To select the child DB object, use the **Up** or **Down** arrow keys and press **Enter** on the keyboard, or double-click the child DB object.

On selection, the child DB object will be appended to the parent DB object (with a period '.').

 **NOTE**

- Auto-suggest also works on schema names, table names, views, and table name aliases in the same way as shown above for all schema objects that you have access. Auto-suggest for column name alias is not supported.

Following is a sample query with alias objects:

```
SELECT
  table_alias.<auto-suggest>
FROM test.t1 AS table_alias
WHERE
  table_alias.<auto-suggest> = 5
GROUP BY table_alias.<auto-suggest>
HAVING table_alias.<auto-suggest> = 5
ORDER BY table alias.<auto-suggest>
```

- Auto-suggest may show "Loading" in Terminal for following scenarios:
  - The object is not loaded due to the value mentioned in the **Load Limit** field. Refer to [6.2.2 Adding a Connection](#) for more information.
  - The object is not loaded since it is added in the **Exclude** list option. Refer to [6.2.2 Adding a Connection](#) for more information.
  - There is a delay in fetching the object from the server.
- If there are objects with the same name in different case, then auto-suggest will display child objects of both parent objects.

**Example:**

If there are two schemas with the name public and PUBLIC, then all child objects for both these schemas will be displayed.

---End

## 6.12.8 Viewing the Query Execution Plan and Cost

The execution plan shows how the table(s) referenced by the SQL statement will be scanned (plain sequential scan and index scan).

The SQL statement execution cost is the estimate at how long it will take to run the statement (measured in cost units that are arbitrary, but conventionally mean disk page fetches).

Follow the steps below to view the plan and cost for a required SQL query:

- Step 1** Enter the query or use an existing query in the **SQL Terminal** and click  on the SQL Terminal toolbar to view explain plan.

To view explain analyze, click the drop-down from , select **Include Analyze**, and click .

The **Execution Plan** opens in tree view format as a new tab at the bottom by default. The display mode has a tree shape and text style.

 **NOTE**

The data shown in tree explain plan and visual explain may vary, since the execution parameters considered by both are not the same.

Following are the parameters selected for explain plan with/without analyze and the columns displayed:

**Table 6-15** Explain Plan Options

Explain Plan Type	Parameters Selected	Columns
Include Analyze unchecked (default setting)	Verbose, Costs	Node type, startup cost, total cost, rows, width, and additional Info
Include Analyze checked	Analyze, Verbose, Costs, Buffers, Timing	Node type, startup cost, total cost, rows, width, Actual startup time, Actual total time, Actual Rows, Actual loops, and Additional Info

Additional Info column includes, predicate information (filter predicate, hash condition), distribution key and output information along with the node type information.

The tree view of plan categorizes nodes into 16 types. In tree view, each node will be preceded with corresponding type of icon. Following is the list of node categories with icons:

**Table 6-16** Node Categories with Icon

Node Category	Icon
Aggregate	
Group Aggregate	
Function	
Hash	
Hash Join	
Nested Loop	
Nested Loop Join	
Modify Table	
Partition Iterator	
Row Adapter	
Seq Scan on	

Node Category	Icon
Set Operator	
Sort	
Stream	
Union	
Unknown	

Hover over the highlighted cells to identify the heaviest, costliest, and slowest node. Cells will be highlighted only for tree view.

If multiple queries are selected, explain plan with/without analyze will be displayed only for last query selected.

Each time execution plan is executed, the plan opens in a new tab.

If the connection is lost and the database is still connected in Object Browser, then **Connection Error** dialog box is displayed:

- **Yes** - The connection is reestablished and retrieves explain plan and cost.
- **No** - Disconnects database in Object Browser.

Toolbar menu in the **Execution Plan** window:

Toolbar Name	Toolbar Icon	Description
Tree Format		This icon is used view explain plan in tree format.
Text Format		This icon is used view explain plan in text format.
Copy		This icon is used to copy selected content from result window to clipboard. Shortcut key - <b>Ctrl+C</b> .
Save		This icon is used to save the explain plan in text format.

Refer to [Execute SQL Queries](#) for information refresh, SQL preview, and search bar.

Refresh operation re-executes the explain/analyze query and refreshes the plan in the existing tab.

The result is displayed in the **Messages** tab.



- 3 - **Properties - General** tab - Provides the execution time of the query in ms.
- 4 - **Properties - All Nodes** tab - Provides all node information.

Column Name	Description
Node Name	Name of the node
Analysis	Node analysis information is provided.
RowsOutput	Number of rows output by the plan node
RowsOutput Deviation (%)	Deviation % between estimated rows output and actual rows output by the plan node
Execution Time (ms)	Execution time taken by the plan node
Contribution (%)	Percentage of the execution time taken by plan node against the overall query execution time.
Self Cost	<b>Total Cost</b> of the plan node - Total Cost of all child nodes
Total Cost	Total cost of the plan node

- 5 - **Properties - Exec. Plan** tab - Provides the execution information of all nodes.

Column Name	Description
Node Name	Name of the node
Entity Name	Name of the object
Cost	Execution time taken by the plan node
Rows	Number of rows output by the plan node
Loops	Number of loops of execution performed by each node.
Width	The estimated average width of rows output by the plan node in bytes
Actual Rows	Number of estimated rows output by the plan node
Actual Time	Actual execution time taken by the plan node

- 6 - **Plan Node - General** tab - Provides the node information for each node.

Row Name	Description
Output	Provides the column information returned by the plan node

Row Name	Description
Analysis	Provides analysis of the plan node like costliest, slowest, and heaviest.
RowsOutput Deviation (%)	Deviation % between estimated rows output and actual rows output by the plan node
Row Width (bytes)	The estimated average width of rows output by the plan node in bytes
Plan Output Rows	Number of rows output by the plan node
Actual Output Rows	Number of estimated rows output by the plan node
Actual Startup Time	The actual execution time taken by the plan node to output the first record
Actual Total Time	Actual execution time taken by the plan node
Startup Cost	The execution time taken by the plan node to output the first record
Total Cost	Execution time taken by the plan node
Is Column Store	This field represents the orientation of the table (column or row store)
Node Type	Represents the type of node
Parent Relationship	Represents the relationship with the parent node

Based on the plan node type additional information may display. Few examples:

Plan Node	Additional Information
Partitioned CStore Scan	Table Name, Table Alias, Schema Name
Vector Sort	Sort keys
Vector Hash Aggregate	Group By Key
Vector Has Join	Join Type, Hash Condition
Vector Streaming	Distribution key, Spawn On

- **7 - Plan Node - DN Details** tab - Provides detailed data node information for each node. DN Details are available only if data is being collected from data node.

Refer to [6.6.7.7 Viewing Table Data](#) section for description on copy and search toolbar options.

---End

## 6.12.10 Working with the SQL Terminals

In the **SQL Terminal**, you can

- [Auto Commit](#)
- [Execute SQL Queries](#)
- [Backup Unsaved Queries/Functions/Procedures](#)
- [Locate Error](#)
- [Search in PL/SQL Viewer or SQL Terminal](#)
- [Go to Line in PL/SQL Viewer or SQL Terminal](#)
- [Comment/Uncomment](#)
- [Indent/Unindent Lines](#)
- [Insert Space](#)
- [Execute Multiple Functions/Procedures or Queries](#)
- [Rename SQL Terminal](#)
- [SQL Assistant](#)
- [Using Templates](#)

### Auto Commit

Auto Commit option can be switched on or off based on the **Preferences** settings. Refer [Transaction](#) section for further details on how to enable and disable **Auto Commit** option.

- If **Auto Commit** option is enabled, **Commit** and **Rollback** buttons are disabled. Transactions are committed automatically.
- If **Auto Commit** option is disabled, **Commit** and **Rollback** buttons are enabled. You can use the buttons manually to commit or revert the changes.

#### NOTE

- For OLAP, server will open a transaction for all the SQL statements.(For Example: select statement, explain select statement, set parameter)
- For OLTP ,Server will open transaction for only DML statements.(For Example: INSERT statement, UPDATE statement, DELETE statement)

### Reuse Connection

It enables the user to choose the same SQL terminal connection or new connection for the result set. The choice affects the record visibility as per the isolation levels defined in the database server.

- When **Reuse Connection** is **ON**, terminal connection will be used for data manipulation and refresh of the result window.

For some data base temp tables that are created or used by the terminal can be edited from the result window.

- When **Reuse Connection** is **OFF**, new connection will be used for data manipulation and refresh of the result window.

For some data base temp tables cannot be edited from the result window.



Icon is displayed when Reuse Connection is **ON**.



Icon is displayed when Reuse Connection is **OFF**.



Icon is displayed when Reuse Connection is **disabled**.

**Follow the steps to turn off Reuse Connection:**

**Step 1** Click  on the **SQL Terminal** toolbar.

Reuse Connection is disabled for the terminal. 

 **NOTE**

- Auto Commit ON- Reuse Connection is enabled and ON by default. You can switch it OFF if needed.
- Auto Commit OFF- Reuse Connection is disabled and ON.

----**End**

Refer to [Table 7-1](#) for more details about the behavior of **Auto Commit** and **Reuse Connection**.

## Execute SQL Queries

**Follow the steps below to execute function/procedure(s) or SQL queries**

Enter a function/procedure(s) or SQL query(s) in the **SQL Terminal** tab and click  in the **SQL Terminal** tab, or press **Ctrl+Enter**, or choose **Run > Compile/Execute Statement** from the main menu.

Alternatively, you can right-click in the **SQL Terminal** tab and select **Execute Statement**.

 **NOTE**

You can check the status bar to view the status of a query being executed.

The **Result** tab displays the results after executing the function/procedure(s) or SQL queries along with the query statement executed.

If the connection is lost during execution and the database is still connected in Object Browser, then **Connection Error** dialog box is displayed:

- **Reconnect** - The connection is reestablished.
- **Reconnect and Execute** - With Auto commit on, execution will continue from failure statement. With Auto commit off, execution will continue from position of cursor.
- **Cancel** - Disconnects database in Object Browser.

Failure to reconnect after three attempts will disconnect the database in Object Browser. Connect to the database in Object Browser and retry execution.

 **NOTE**

- For long running queries, result set can be edited only after the complete results are fetched.
- Editing of query results are only allowed in following scenarios:
  - Select is from a single table
  - Either select all columns or subset of columns [No aliases, aggregate functions, expressions on columns]
  - All WHERE condition
  - All ORDER BY clause
  - On regular, partition, and temporary tables.
- Committing an empty row assigns Null to all columns.
- Only result set of queries executed on tables available in Object Browser is editable.
- Editing of query results is allowed only for queries executed in SQL Terminal.

The column width definition can be set using **Settings > Preferences** option. Refer to [Query Results](#) to set this parameter.

**Column Reorder**

Column reordering can be performed by clicking and dragging the selected column header to the desired position.

**Multi-Column Sort**

This feature allows users to sort table data of some pages by multiple columns. In addition, you can set the priority of columns for sorting.

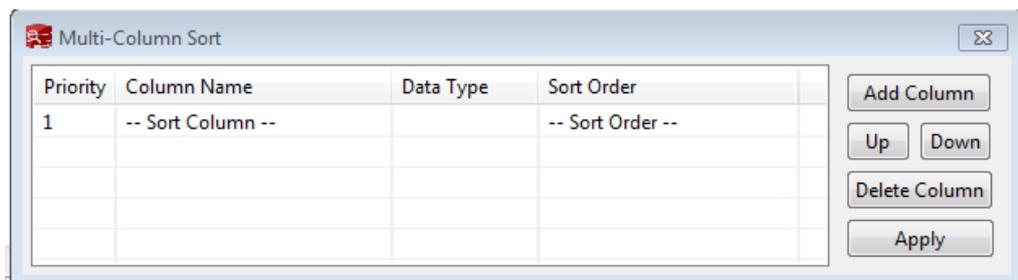
The feature is available for the following pages:

- Result Set Tab
- Edit Table Data Window
- View Table Data Window
- Batch Drop Result Window

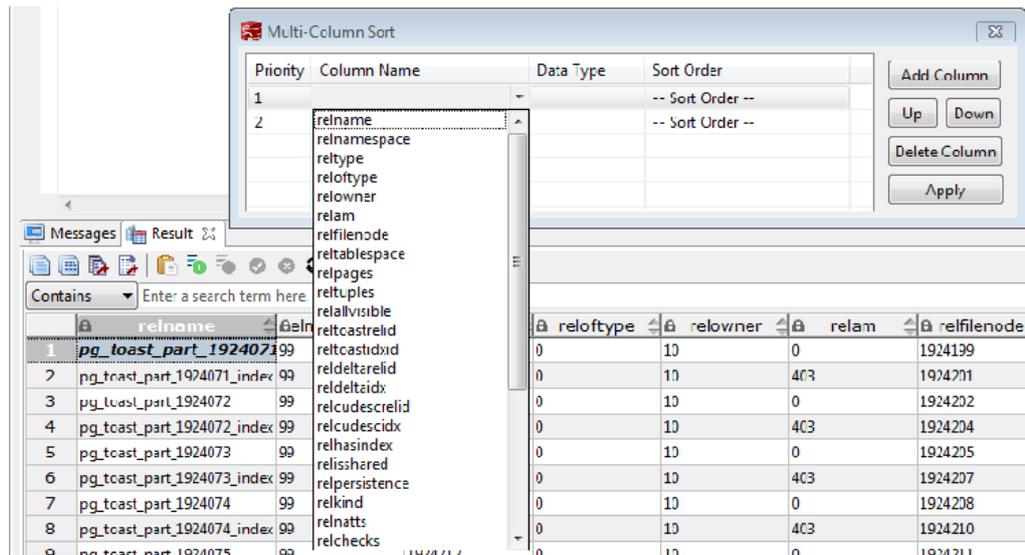
Follow the Step to access Multi-column sort:

**Step 1** Click  button on the toolbar.

**Multi-Column Sort** pop-up is displayed.



**Step 2** Click **Add Column**. Choose the column you want to sort from the drop down.



**Step 3** Select the required sort order.

**Step 4** Click on **Apply**.

----End

Multi-sort pop up has following elements:

**Table 6-17** Elements Of Multi-Column Pop-up:

Attribute Name	UI Element Type	Description/Action
Priority	Read only text field	Shows column priority in multi sort.
Column Name	Combo field having all column names of the table as its value set	Column name of the column added for sorting.
Data Type	Read only text field	Shows data type of the column selected.
Sort Order	Combo field having values {sort_ascending, sort_descending}	Sort order of the column.
Add Column	Button	Adds new row to multi-sort table.
Delete Column	Button	Deletes selected column from multi-sort table.
Up	Button	Moves selected column up by 1 step, thus changing sort priority.

Attribute Name	UI Element Type	Description/Action
Down	Button	Moves selected column down by 1 step, this changing sort priority.
Apply	Button	Apply prepared sort configuration.

 **NOTE**

Except following data types, all the other data types will be sorted by their string value (Alphabetical order):

TINYINT, SMALLINT, INTEGER, BIGINT, FLOAT, REAL, DOUBLE, NUMERIC, BIT, BOOLEAN, DATE, TIME, TIME\_WITH\_TIMEZONE, TIMESTAMP, TIMESTAMP\_WITH\_TIMEZONE.

Following icons are provided in Multi-Column sort:

Elements Of Multi-Column Pop-up:

**Table 6-18** Icons Of Multi-Column Sort:

Icon	Description	Action
	Not Sorted	This icon in column header indicates that the column is not sorted. If you click on this icon the column will be sorted in ascending order. Alternatively click <b>Alt +column header</b>
	Ascending Sort	This icon in column header indicates that the column is sorted in ascending order. If you click on this icon, the column will be sorted in descending order. Alternatively click <b>Alt +column header</b>
	Descending Sort	This icon in column header indicates that the column is sorted in descending order. If you click on this icon the column will be in no sort order. Alternatively click <b>Alt +column header</b>

Icons for the sort priority are as folloing:

 : Icons having three dots have the highest priority.

 : Icons having two dots have the second highest priority.

 : Icons having three dots have the third and onwards priority.

**Table 6-19** Toolbar Menus

Toolbar Name	Toolbar Icon	Description
Copy		This button is used to copy selected content from result window to clipboard. Shortcut key - <b>Ctrl+C</b> .
Advanced Copy		This button is used to copy content from result window to clipboard. Results can be copied to include column header. Refer to <a href="#">View Query Results</a> to set this preference. Shortcut key - <b>Ctrl+Shift+C</b> .
Export all data		This icon is used to export all data either in excel (xlsx/xls), CSV, Text or Binary format. Refer to <a href="#">6.6.7.4 Exporting Table Data</a> . <b>NOTE</b> <ul style="list-style-type: none"> <li>Columns mentioned in the query is auto-populated in the <b>Selected Columns</b> section with <b>Available Columns</b> section empty.</li> <li>To export the query results, the query is re-executed using a new connection. The exported results may differ from the data in the results tab.</li> <li>Disabled for explain/analyze queries. To export explain/analyze queries use the <b>Export current page data</b> option.</li> </ul>
Export current page data		This button is used to export current page data to excel (xlsx/xls) or CSV.
Paste		This button is used to paste copied information. Refer to <a href="#">Paste</a> section for more information.
Add		This button is used to add a row to the result set. Refer to <a href="#">Insert</a> section for more information.
Delete		This button is used to delete a row from the result set. Refer to <a href="#">Delete</a> section for more information.
Save		This button is used to save the changes made in the result set. Refer to <a href="#">6.6.7.8 Editing Table Data</a> section for more information.
Rollback		This button is used to roll back the changes made to the result set. Refer to <a href="#">6.6.7.8 Editing Table Data</a> section for more information.

Toolbar Name	Toolbar Icon	Description
Refresh		This button is used to refresh information in the result set. If multiple result sets are open for the same table, then changes made to one result set will reflect on the other post refresh. Similarly if the same table is edited, then the result set will be updated post refresh.
Clear Unique Key selection		This button is used to clear the previous unique key selection. Refer to <a href="#">6.6.7.8 Editing Table Data</a> section for more information.
Show/Hide Query bar		This button is used to display/hide the query executed for that particular result set. This is a toggle button.
Show/Hide Search bar		This button is used to display/hide the search text field. This is a toggle button.
Encoding		This field will be available based on the <b>Preference &gt; Result Management &gt; Query Result &gt; Result Advanced Copysettings</b> . This drop-down is used to select the appropriate encoding to view the data accurately. The default encoding is UTF-8. Refer to <a href="#">Result Data Encoding</a> section to set the encoding preference. <b>NOTE</b> <ul style="list-style-type: none"> <li>● Data editing except for data insertion is restricted once the default encoding is modified.</li> <li>● Only UTF-8 and GBK encoding are supported by GaussDB 100 database.</li> </ul>
Multi Sort		This button brings up multi-sort pop up.
Clear Sort		This button is used to reset all the sorted column.

Icons in Search field:

Icon Name	Icon	Description
Search		This icon is used to search the result set based on the criteria defined. Search text are case insensitive.
Clear Search Text		This icon is used to clear the search text entered in the search field.

Right-click options in the **Result** window:

Option	Description
Close	Closes only the active result window.
Close Others	Closes all other result windows except for the active result window.
Close All	Closes all result windows including the active result window.

Status information displayed in the **Result** window:

- **Query Submit Time** - Provides the query submitted time.
- Number of rows fetched with execution time is displayed. The default number of rows is displayed. If there are additional rows to be fetched, then it will be denoted with the word "more". You can scroll to the bottom of the table to fetch and display all rows.

#### NOTICE

When viewing table data, Data Studio automatically adjusts the column widths for an optimal table view. Users can resize the columns as needed. If the text contents of a cell exceeds the total available display area, then resizing the cell column may cause DS to become unresponsive.

#### NOTE

- Each time a query is run in **SQL Terminal** tab, a new result window opens. To view the results in the new window, you must select the newly opened window.
- Set the **focusOnFirstResult** configuration parameter to **false** to automatically set focus to the newly opened **Result** window. Refer to [4.1 Installing and Configuring Data Studio](#) for details.
- Each row, column and selected cells can be copied from the result set.
- Export all data operation will be successful even after the connection is removed.
- If the content of the column have spaces between the words, then word wrap is applied to fit the column within the display area. Word wrap is not applied if the content does not have any spaces between the words.
- Select part of cell content and press **Ctrl+C** or click  to copy selected text from a cell.
- The size of the column is determined by the maximum content length column.
- You can save preference to define:
  - Number of records to be fetched.
  - Column width
  - Copy option from result set.Refer to [Query Results](#) for more information.

## Backup Unsaved Queries/Functions/Procedures

Data Studio creates back up of unsaved data in SQL Terminal and PL/SQL Viewer periodically based on the time interval defined in the **Preferences** tab. The data can be encrypted and saved based on **Preference** settings. Refer to [Query/Function/Procedure Backup](#) section to turn on/off backup, define time interval to save the data, and encrypt the saved data.

Unsaved changes of each SQL Terminal/PL/SQL Viewer are taken as backup and stored in *DataStudio\UserData\\Autosave folder*. Backup files saved before unexpected shutdown of Data Studio will be available at next login.

In case there are unsaved data in SQL Terminal/PL/SQL Viewer, during graceful exit, Data Studio will wait for backup to complete before closing.

## Error Locator

During execution of query/function/procedure in case of an error the error locator message is displayed.

**Yes** - Click **Yes** to continue with the execution.

**No** - Click **No** to stop the execution.

Select **Do not show additional errors for this execution** option to hide the error message popup from displaying while continuing with the current execution.

Line number and position of error displays in **Messages** tab. The corresponding line number is marked with  icon along with red underline at the position of the error in the Terminal/PL/SQL Viewer. Hovering over  displays the error message. Refer to [11 FAQs](#) section to understand in certain scenarios why the line number and error detail does not match.

### NOTE

If the query/function/procedure is modified while execution is in progress, then error locator may not display the correct line and position number.

## Search in PL/SQL Viewer or SQL Terminal

Follow the steps below to search in PL/SQL Viewer or SQL Terminal:

**F3** key is used to search next word and **Shift+F3** key is used to search previous word. These shortcut keys will be enabled only after **Ctrl+F** is used to search a text. These keys will be active with the current search word until a new word is searched. The value searched using **Ctrl+F** and **F3/Shift+F3** will be applicable only for the current instance.

**Step 1** Choose **Edit > Find and Replace** from the main menu.

Alternatively press **Ctrl+F**.

**Find and Replace** dialog box is displayed.

**Step 2** Enter the text to be searched in **Find what** field, and click the **Find Next** button.

The searched text is highlighted.

**F3** and **Shift+F3** key will now be enabled for forward and backward search.

### NOTE

Select **Wrap around** option to continue the search after reaching the last line in the SQL queries or PL/SQL statements.

----End

## Go to Line in PL/SQL Viewer or SQL Terminal

Go to line option is used to skip to a specific line in the terminal.

Follow the steps below to go to a line in PL/SQL Viewer or SQL Terminal:

**Step 1** Choose **Edit > Go To Line** from the main menu or

Alternatively press **Ctrl+G**.

**Go to Line** dialog box is displayed.

**Step 2** Enter the desired number in the **Enter the line number** field, and then click the **OK** button.

The cursor moves to the beginning of the line entered in the **Go to Line** dialog box.

### **NOTE**

Below are invalid inputs to this field.

- Non-numeric value
- Special characters
- Line number entered does not exist in the editor.
- More than 10 digits is entered.

---End

## Comment/Uncomment

Comment/uncomment option is used to comment/uncomment lines or block of lines.

Follow the steps below to comment/uncomment lines in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to comment/uncomment.

**Step 2** Choose **Edit > Comment/Uncomment Lines** from the main menu, or alternatively press **Ctrl +/,** or right-click, and select **Comment/Uncomment Lines**, the selected lines are commented/uncommented.

---End

Follow the steps below to comment/uncomment block of lines/content in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines/content to comment/uncomment.

**Step 2** Choose **Edit > Comment/Uncomment Block** from the main menu, or alternatively press **Ctrl+Shift+/,** or right-click and select **Comment/Uncomment Block**, the selected block of lines/content are commented/uncommented.

---End

## Indent/Unindent Lines

The indent/unindent option is used to shift lines as per the indent size defined in the **Preferences** tab.

Follow the steps below to indent lines in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to indent.

**Step 2** Press **Tab** or click .

Shifts the selected line as per the indent size defined in the **Preferences** tab. Refer to **Formatter** section to modify the indent size.

----End

Follow the steps below to unindent lines in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to unindent.

**Step 2** Press **Shift+Tab** or click .

Shifts the selected line as per the indent size defined in the **Preferences** tab. Refer to **Formatter** section to modify the indent size.

 **NOTE**

Only selected lines that have available tab space will be unindented. For example, if multiple lines are selected, and one of the selected line starts at position 1, then pressing **Shift+Tab** will unindent all the lines except for the one starting at position 1.

----End

## Insert Space

The **Insert Space** option is used to replace a tab with spaces based on the indent size defined in the **Preferences** tab.

Follow the steps below to replace a tab with spaces in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to replace tab with spaces.

**Step 2** Press **Tab** or **Shift+Tab**.

Replaces the tab with spaces as per the indent size defined in the **Preferences** tab. Refer to **Formatter** section to modify the indent size.

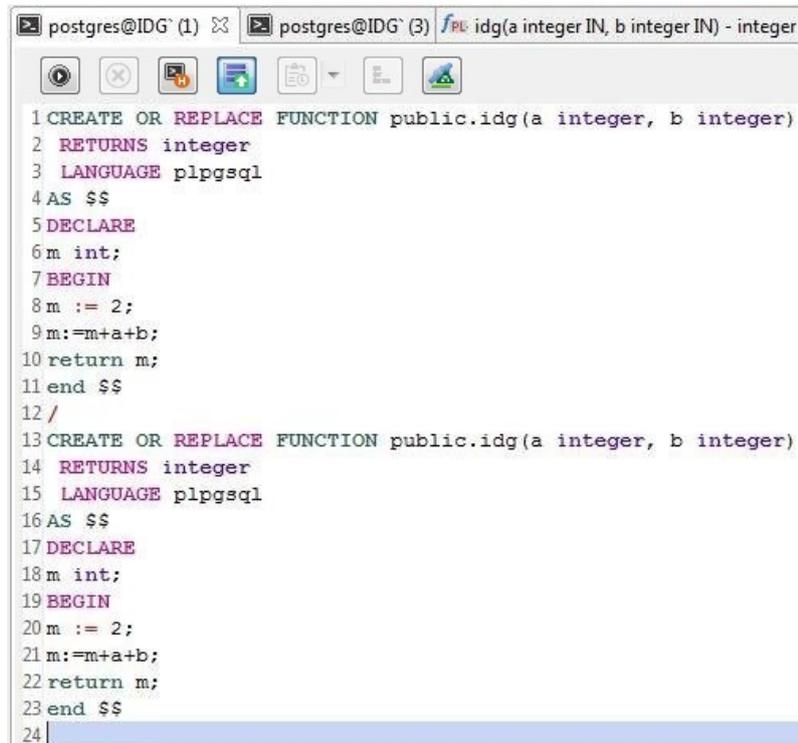
----End

## Execute Multiple Functions/Procedures or Queries

Follow the steps below to execute multiple functions/procedures:

Insert **'/'** in a new line after the function/procedure in the **SQL Terminal**.

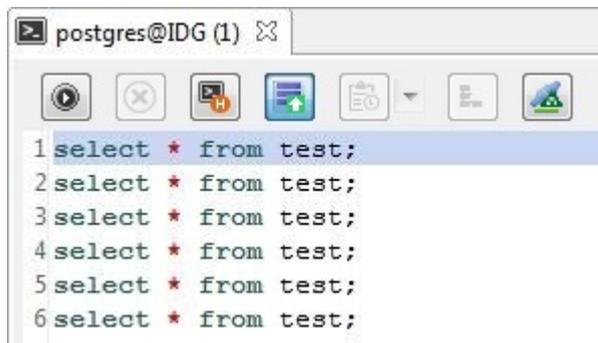
Add the new function/procedure in the next line.



```
postgres@IDG` (1)  postgres@IDG` (3)  /pg idg(a integer IN, b integer IN) - integer
1 CREATE OR REPLACE FUNCTION public.idg(a integer, b integer)
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6 m int;
7 BEGIN
8 m := 2;
9 m:=m+a+b;
10 return m;
11 end $$
12 /
13 CREATE OR REPLACE FUNCTION public.idg(a integer, b integer)
14 RETURNS integer
15 LANGUAGE plpgsql
16 AS $$
17 DECLARE
18 m int;
19 BEGIN
20 m := 2;
21 m:=m+a+b;
22 return m;
23 end $$
24
```

Follow the steps below to execute multiple SQL queries:

**Step 1** Enter multiple SQL queries in the **SQL Terminal** tab as follows:



```
postgres@IDG (1)
1 select * from test;
2 select * from test;
3 select * from test;
4 select * from test;
5 select * from test;
6 select * from test;
```

**Step 2** Click  in the **SQL Terminal** tab, or press **Ctrl+Enter**, or choose **Run > Compile/Execute Statement** from the main menu.

 **NOTE**

- If the queries are not selected for execution, then only the query in the line where cursor is placed will be executed.
- If the cursor is placed next to an empty line, then the next available query statement will be executed.
- If the cursor is placed at the last line which is blank, then no query will be executed.
- If a single query is written in multiples lines and the cursor is placed at any line of the query, then that query is executed. Queries are separated using semicolon (;).

----End

Follow the steps below to execute an SQL query after a function/procedure:

Insert '/' in a new line after the function/procedure in the **SQL Terminal**.

Follow the steps below to execute PL/SQL statements and SQL queries on different connections:

In the toolbar, select the required connection from the connection profiles drop-down list and click  in the **SQL Terminal** tab.

## Rename SQL Terminal

Follow the steps below to rename SQL Terminal:

**Step 1** In the **SQL Terminal** tab right-click and select **Rename Terminal**.

A **Rename Terminal** dialog box is displayed prompting you to provide the new name for the Terminal.

**Step 2** Enter the new name and select **OK** to rename the Terminal.

### NOTE

- Terminal name follows Windows file naming convention.
- **Rename Terminal** accepts a maximum of 150 characters.
- Restore option is not available to revert to the default name. You must manually rename the Terminal to default name.
- Tool tip of the renamed Terminal will display the old name.

----End

## SQL Assistant

The **SQL Assistant** tool provides suggestion or reference for the information entered in the SQL Terminal and PL/SQL Viewer. Follow the steps below to open SQL Assistant:

When Data Studio is launched **SQL Assistant** panel displays with related syntax topics. As you type a query in the SQL Terminal topics related to the query is displayed. It also provides precautions, examples, syntax, function, and parameter description. Select the text and use the right-click option to copy selected information or copy and paste to SQL Terminal.

### NOTE

- You can enable/disable the **SQL Assistant** tool permanently. Refer to [7.3-SQL Assistant](#) to enable/disable this option.
- SQL Assistant icon () from the toolbar can be used to open the SQL Assistant window.
- This feature is available for:
  - GaussDB 200 V100R002C70, GaussDB 200 V100R002C80 , and GaussDB 200 6.5.RC2 versions.
  - GaussDB 100 V100R006C00, GaussDB 100 V300R001C00, and V300R001C10 versions.

## Using Templates

Data Studio provides an option to insert frequently used SQL statements in the SQL Terminal/PL/SQL Viewer using the **Templates** option. Some of the commonly used SQL

statements are saved for ease of use. You can create, modify existing templates or remove templates. Refer to [Adding/Modifying Templates](#) section for information on adding, removing, and creating new templates.

The following table lists the default templates:

Name	Description
df	delete from
is	insert into
o	order by
s*	select from
sc	select row count
sf	select from
sl	select

Follow the steps to use the **Templates** option:

**Step 1** Enter the name of the template in SQL Terminal/PL/SQL Viewer.visual

**Step 2** Press **Alt+Ctrl+Space**.

A list of saved template information is displayed. The list displayed is based on the following criteria.

Exact Match	Display List
On	Displays all entries that match the input text case. <b>Example:</b> Entering "SF" in SQL Terminal/PL/SQL Viewer displays all entries that start with "SF".
Off	Displays all entries that match the input irrespective of the text case. <b>Example:</b> Entering "SF" in SQL Terminal/PL/SQL Viewer displays all entries that start with "SF", "Sf", "sF", or "sf".

Text Selection/Cursor Location	Display List
A text is selected and the shortcut key is used	Displays entries that match the text before the selection to the nearest space or new line character.

Text Selection/Cursor Location	Display List
No text selected and the shortcut key is used	Displays entries that match the text before the cursor to the nearest space or new line character.

 **NOTE**

- Using the shortcut key without entering text in SQL Terminal/PL/SQL Viewer displays all entries in the **Templates**.
- If the text entered in SQL Terminal/PL/SQL Viewer has only a single match, then it will be replaced directly in the SQL Terminal/PL/SQL Viewer without listing them out.
- After you click **Open SQL Assistant** button, the **SQL Assistant** pane is displayed. If you close the pane, it will be closed for all the terminals unless you click **Open SQL Assistant** button again.

----End

## 6.12.11 Exporting Query Results

You can export the results of an SQL query into a CSV, Text or Binary file.

This section contains the following topics:

- [Exporting all data](#)
- [Exporting current page data](#)

### Exporting all data

The following functions are disabled while the export operation is in progress:

- Executing SQL queries in the **SQL Terminal**
- Executing PL/SQL statements
- Debugging PL/SQL statements

Follow the steps below to export all results:

**Step 1** Select the **Result** tab.

**Step 2** Click .

**Export ResultSet Data** window is displayed.

Refer to [6.6.7.4 Exporting Table Data](#) to complete the export operation.

 **NOTE**

You can check the status bar to view the status of the result being exported.

The **Data Exported Successfully** dialog box is displayed.

**Step 3** Click **OK**. Data Studio displays the status of the operation in the **Messages** tab.

 **NOTE**

If the disk is full while exporting the results, then Data Studio displays an error in the **Messages** tab. Clean the disk, re-establish the connection and export the result data.

----End

The **Messages** tab shows the **Execution Time**, **Total result records fetched**, and the path where the file is saved.

## Exporting current page data

It is recommended to export all results instead of exporting the current page.

Follow the steps below to export the current page:

**Step 1** Select the **Result** tab.

**Step 2** Click  to export the current page.

The **Data Studio Security Disclaimer** dialog box is displayed.

**Step 3** Click **OK**.

**Step 4** Select the location to save the current page.

 **NOTE**

You can check the status bar to view the status of the page being exported.

**Step 5** Click **Save**.

The **Data Exported Successfully** dialog box is displayed.

**Step 6** Click **OK**. Data Studio displays the status of the operation in the **Messages** tab.

 **NOTE**

If the disk is full while exporting the results, then Data Studio displays an error in the **Messages** tab. Clean the disk, re-establish the connection and export the result data.

----End

## 6.12.12 Managing SQL Terminal Connections

Data Studio allows you to reuse an existing SQL Terminal connection or create a new SQL Terminal connection for execution plan and cost, visual explain plan, and operations in the resultset. By default, the SQL Terminal reuses the existing connection to perform these operations.

Use new connection when there are multiple queries queued for execution in existing connection as the queries are executed sequentially and there may be a delay. Always reuse existing connection while working on temp tables. Refer to the [6.6.8 Editing Temporary Tables](#) section to edit temp tables.

Complete the steps to enable or disable SQL Terminal connection reuse:

**Step 1** Click  to enable or disable SQL Terminal connection reuse.

Refer to the [FAQs](#) section for the behavior of query execution with reuse and new connection.

 **NOTE**

Use the existing SQL Terminal connection to edit temporary tables.

----End

## 6.13 Batch Operation

### 6.13.1 Overview

You can view database objects to which you have access in **Object Browser** in the tree format. For example, you can view the schema names within the selected database and the corresponding table names within the selected schema.

The **Object Browser** displays only the objects that satisfy the following minimum privilege type requirement for the current user.

Object Type	Privilege to Display in Object Browser
Database	Connect
Schema	Usage
Table	Select
Column	Select
Sequences	Usage
Function/Procedure	Execute
Tablespace	Create

It is not necessary that child objects of a parent object to which you have access will be displayed in **Object Browser**. For example, if you have access to a table but not have access to one of the columns in that table, then **Object Browser** displays only the table with columns to which you have access. The columns which you do not have access are not displayed. If access to an object is revoked when an operation is being performed, then an error message is displayed stating that you do not have access to perform the operation, and the object is removed from **Object Browser** after refresh.

Following database objects are supported (displayed in the tree format):

- Schemas
- Functions/Procedures
- Tables
- Sequences
- Views
- Columns, Constraints, and Indexes

- Tablespaces

All default created schemas except for public are grouped under **Catalogs** and user schemas are grouped under **Schemas** below the respective database.

## 6.13.2 Dropping Batch of Objects

The batch drop operation allows you select multiple objects to drop. You can also perform batch drop operation on searched objects.

### NOTE

- Batch drop is allowed only within the database.
- Batch drop of system objects will result in error, since system objects cannot be dropped.

Follow the steps to drop objects in a batch.

**Step 1** Press **Ctrl+left-click** (select objects one by one) or **Shift+left-click** (select objects in a bunch) to select the objects to be dropped.

**Step 2** Right-click and select **Drop Objects**.

**Drop Objects** tab displays with the list of objects to be dropped.

Column Name	Description	Example
Type	Displays information on the object type.	table, views
Name	Displays the name of the object.	public.bs_operation_201804
Query	Displays the query that will be executed to drop the object.	DROP TABLE IF EXISTS public.a123
Status	<p>Displays the status of the drop operation.</p> <ul style="list-style-type: none"> <li>●  - To start: The drop operation yet to be initiated.</li> <li>●  - In progress: The object is currently being dropped.</li> <li>●  - Completed: The drop operation has been completed.</li> <li>●  - Error: The object has not been dropped due to an error.</li> </ul>	<ul style="list-style-type: none"> <li>● To start</li> <li>● In progress</li> <li>● Completed</li> <li>● Error</li> </ul>
Error Message	Displays the failure reason of the drop operation.	table "abc" does not exist, skipping

**Step 3** Select the required drop option:

Option	Description
Cascade	Cascade drop operation drops their dependent objects and attributes. The dependent objects that are dropped will be removed from the Object Browser only after refresh operation is performed.
Atomic	Atomic drop operation drops all objects in case of success or drops none in case of a failure. This functionality is not applicable for GaussDB 100.
No selection	Unselection of Atomic or Cascade does not drop dependent objects.

**Step 4** Click **Start**.

**Runs** - Displays the number of objects that are dropped from the total list of objects.

**Errors** - Displays the number of object that was not dropped due to errors.

**Step 5** Click **Stop** or close the **Drop Objects** dialog to stop the drop operation.

Refer to [Execute SQL Queries](#) section for information on copy, advanced copy, show/hide search bar, sort, and column reorder options.

 **NOTE**

- Select part of cell content and press **Ctrl+C** or click  to copy selected text from a cell.
- When you select multiple objects in object browser to drop, a batch drop window is opened and its menu icons are enabled in the menubar. If you disconnect the database, the icons will remain disabled and will not be enabled even after reconnecting. You need to reselect the objects to drop and the selected objects will be available in the new terminal window.

---End

## 6.13.3 Granting/Revoking Privileges

The batch grant/revoke operation allows you select multiple objects to grant/revoke privileges. You can also perform batch grant/revoke operation on searched objects.

This feature is only available for OLAP, not for OLTP.

 **NOTE**

Batch grant/revoke is allowed only with the same object type within that schema.

Follow the steps to grant/revoke privileges in a batch.

**Step 1** Press **Ctrl+left-click** (select objects one by one) or **Shift+left-click** (select objects in a bunch) to select the objects to grant/revoke privileges.

**Step 2** Right-click and select **Grant/Revoke**.

**Grant/Revoke** dialog box is displayed.

**Step 3** Refer to [6.4.7 Grant/Revoke Privilege](#) section to grant/revoke privilege.

---End

---

# 7 Personalizing Data Studio

---

- [7.1 Overview](#)
- [7.2 General](#)
- [7.3 Editor](#)
- [7.4 Environment](#)
- [7.5 Export](#)
- [7.6 Result Management](#)
- [7.7 Security](#)

## 7.1 Overview

This section provides details on how to personalize Data Studio using preferences settings.

## 7.2 General

This section provides details on how to personalize shortcut keys.

### Setting the Shortcut Keys

You can customize the Data Studio shortcut keys as required.

Follow the steps below to set or modify the shortcut keys:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **General > Shortcut Mapper**.

The **Shortcut Mapper** pane is displayed.

**Step 3** Select the required shortcut key and click **Modify**.

**Step 4** Enter the required shortcut key in the **Binding** text box.

For example, to change the shortcut key for **Step Into** from **F7** to **F6**, enter **F6** in the **Binding** text box.

**Step 5** Click **OK**.

The **Restart Data Studio** dialog box is displayed.

 **NOTE**

- Multiple shortcut keys can be modified before restarting Data Studio.
- There may be some shortcut keys that are not supported by GaussDB 100.

**Step 6** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then the **Restart Confirmation** dialog box is displayed.

**Step 7** Click **OK** to close running jobs and restart or click **Cancel** to abort restart operation.

----End

Follow the steps below to remove the shortcut keys:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **General > Shortcut Mapper**.

The **Shortcut Mapper** pane is displayed.

**Step 3** Select the required shortcut key and click **Unbind Key**.

**Step 4** Click **Ok**.

The **Restart Data Studio** window is displayed.

 **NOTE**

Multiple shortcut keys can be removed before restarting Data Studio

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then the **Process is running** dialog box is displayed.

**Step 6** Click **OK** to wait for operations to complete or click **Force Restart** to discard operations.

----End

Follow the steps below to restore the default shortcut keys:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **General > Shortcut Mapper**.

The **Shortcut Mapper** pane is displayed.

**Step 3** Click **Restore Defaults**. For more information on default shortcut keys, refer to [5.5 Data Studio Right-Click Menus](#).

**Step 4** Click **Ok**.

The **Restart Data Studio** window is displayed.

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, the **Process is running** dialog box displays.

**Step 6** Click **OK** to wait for operations to complete or click **Force Restart** to discard operations.

----End

## Shortcut Keys

Data Studio supports keyboard short cut keys similar to other windows based application. The following table lists some of the shortcut keys for effective usage of the functionalities provided by Data Studio. To customize the shortcut keys, refer to [Setting the Shortcut Keys](#).

Function	Shortcut Key
Auto-suggest a List of DB Objects	Ctrl+Space
Backward Search	Shift+F3
Cancel the Executing Query	Shift+Esc
Change Text Case to Lower	Ctrl+Shift+L
Change Text Case to Upper	Ctrl+Shift+U
Close an Active PL/SQL Viewer, View Table Data, Edit Data, and Properties Tab	Shift+F4
Collapse selected Open Objects	Alt+Q
Comment/Uncomment lines individually	Ctrl+/ 
Comment/Uncomment block of lines/content	Ctrl+Shift+/ 
Compile/Execute SQL Terminal Statements	Ctrl+Enter
Continue with the plsql debug operation	F9
Copy Qualified Object Name from Object Browser, selected content from Terminal, Result, View Table Data or Edit Table Data tab	Ctrl+C
Copy content from Result, View Table Data or Edit Table Data tab including/excluding column header and row number	Ctrl+Shift+C
Copy Breakpoints Pane Content	Alt+Y
Copy Callstack Pane Content	Alt+J
Copy DB Objects from the Auto-suggest List	Alt+U
Copy Execution Time and Status from the Edit Table Data Tab	Ctrl+Shift+K
Copy the Query displayed in the Edit Data Tab	Ctrl+Alt+C
Copy Variables Pane Content	Alt+K

Function	Shortcut Key
Cut	Ctrl+X
Debug DB Object	Ctrl+D
Debug Menu	Alt+D
Disconnect from DB	Ctrl+Shift+D
Edit Menu	Alt+E
Execute DB Object	Ctrl+E
Execution Plan and Cost	Ctrl+Shift+X
Expand/Collapse all the objects	Ctrl+M
File Menu	Alt+F
Find and Replace	Ctrl+F
Format (SQL, PL/SQL)	Ctrl+Shift+F
Forward Search	F3
Go to Line	Ctrl+G
Help Menu	Alt+H
Help (Show User Manual)	F1
Highlight Object Browser	Alt+X
Locate first element of Object browser	Alt+Page Up or Alt+Home
Locate last element of Object browser	Alt+Page Down or Alt+End
New Database Connection	Ctrl+N
Open Callstack, Breakpoints, and Variables pane	Alt+V
Open SQL Script	Ctrl+O
Paste	Ctrl+V
Redo	Ctrl+Y
Refresh Server Object	F5
Run Menu	Alt+R
Save SQL Script	Ctrl+S
Search Objects	Ctrl+Shift+S
Select All	Ctrl+A
Settings Menu	Alt+G

Function	Shortcut Key
Sorts the resultset of view table, edit table, and query in ascending, descending, or server order.	Alt+Click

## 7.3 Editor

This section provides details on how to personalize syntax coloring, SQL history information, templates, and formatter.

### Syntax Coloring

Follow the steps below to customize the SQL highlight color:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Syntax Coloring**.

The **Syntax Coloring** pane is displayed.

**Step 3** Click the color button to customize the color for the type of syntax.

For example, click  to customize the color for **Strings**. The color picker dialog box is displayed.

Use the color picker to set the required color for a specific syntax category. You can choose basic colors or define custom colors in the color picker.

#### **NOTE**

Click **Restore Defaults** from **Syntax Coloring** pane to reset to default color scheme.

**Step 4** Click **OK**. The **Restart Data Studio** dialog box is displayed.

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then Data Studio displays the **Process is running** dialog box.

**Step 6** Click **Force Restart** to discard operations and restart Data Studio. Click **OK** to continue performing operations.

#### **NOTE**

The *Preferences.prefs* file contains the custom color settings. If the file is corrupted, Data Studio will display the default values.

The custom color(s) will be set after you restart Data Studio.

----**End**

### SQL History

You can customize Data Studio to set the number of SQL history count that can be made available and also the number of characters for the query for each of the query saved in SQL history.

Follow the steps to customize the number of executed queries and number of characters in the query to be saved in SQL History:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > SQL History**.

The **SQL History** pane is displayed.

**Step 3** Set the number of queries to be saved in **SQL History Count** field.

 **NOTE**

Minimum value is 1 and maximum is 1000. The current value set for this preference will be displayed.

**Step 4** Set the number of characters to be allowed in each query that is saved in the SQL History in the **SQL Query Characters** field.

 **NOTE**

Minimum value is 1 and maximum is 1000. Enter "0" in this field to set no character limit. The current value set for this preference will be displayed.

**Step 5** Click **Apply**.

**Step 6** Click **OK**.

 **NOTE**

- Click **Restore Defaults** from **SQL History** pane to reset to default value.
- The default value for **SQL History Count** is 50 and **SQL Query Characters** it is 1000.
- If the new value entered is lesser than the old value and if there is going to be a data loss, then a message is displayed informing about the data loss and if you would like to continue with the operation.
- If there are unsaved changes and you navigate away from this pane, then a message displays to state that there are unsaved changes.
- Pinned queries are not affected by the changes made to the **SQL History Count** field. **Example:** If the number of pinned queries is 50 and the **SQL History Count** is set to 25, then SQL History will show 50 pinned queries.
- The **SQL Query Characters** changes affects only queries added post the configuration change.

---End

## Adding Templates

You can customize Data Studio to create new, edit existing, and remove templates. Refer to the [Using Templates](#) section for detailed information on templates.

 **NOTE**

Restoring the settings to default removes all user defined templates from the list.

Follow the steps below to create templates:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Templates**.

The **Templates** pane is displayed.

**Step 3** Click **New**.

**Step 4** Enter a name for the template in the **Name** field.

**Step 5** Enter description in the **Description** field.

**Step 6** Enter the SQL statement pattern in the **Pattern** field.

 **NOTE**

The text entered in **Pattern** field will be syntax highlighted.

**Step 7** Click **OK**.

----**End**

## Modifying Templates

Follow the steps below to edit templates:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Templates**.

The **Templates** pane is displayed.

**Step 3** Click **Edit**.

**Step 4** Edit the name in the **Name** field, if required.

**Step 5** Edit the description in the **Description** field, if required.

**Step 6** Edit the SQL statement pattern in the **Pattern** field, if required.

 **NOTE**

The text entered in **Pattern** field will be syntax highlighted.

**Step 7** Click **OK**.

----**End**

## Removing Templates

Follow the steps below to remove templates:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Templates**.

The **Templates** pane is displayed.

**Step 3** Select the template to be removed, and click **Remove**.

Removes the template from the **Templates** list.

 **NOTE**

Default templates that are removed can be added back using **Restore Removed** option. It will restore the template to the last updated change. **Restore Removed** option is not applicable to user defined templates.

----End

## Reverting to Default Templates

Follow the steps below to revert to default templates:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Templates**.

The **Templates** pane is displayed.

**Step 3** Select at least one default template that is modified to revert to default template settings.

**Step 4** Click **Revert to Default**.

----End

## Formatter

You can customize Data Studio to set the tab width and convert tab to spaces while performing indent and unindent operation. Refer to [Indent/Unindent Lines](#) section to perform indent/unindent operation and replace tab with spaces.

Follow the steps to customize the indent size and convert tab to spaces:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Formatter**.

The **Formatter** pane is displayed.

**Step 3** Select the **Insert Space** option to replace tab with spaces or **Insert Tab** to add/remove tabs while indenting/unindenting lines.

**Step 4** Enter the indent size in **Indent Size**. Based on the number specified in this field, the indent/unindent/space length is defined.

----End

## Transaction

Follow the steps below to edit Transaction settings:

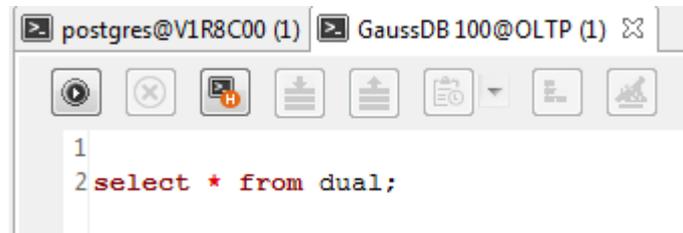
**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Transaction**

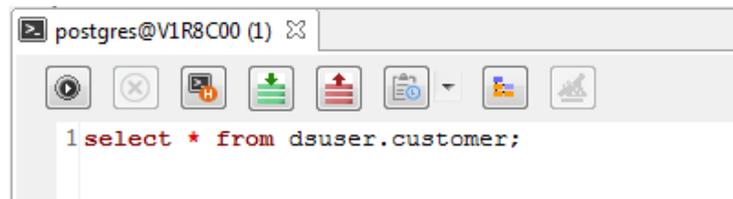
The **Transaction** pane is displayed.

- Step 3** In **Auto Commit** window select **Enable** to switch on the auto commit feature. In this case commit and rollback button will be disabled. Transaction will be committed automatically.



OR

Select **Disable** to switch off the auto commit feature. Commit and Rollback button can be used manually for committing or reverting changes.



**NOTE**

Default behavior for Auto-Commit is ON.

----End

## 7.4 Environment

### Session Setting

Follow the steps below to set Data Studio and file encoding:

- Step 1** Choose **Settings > Preferences** from the main menu.  
The **Preferences** dialog box is displayed.
- Step 2** Choose **Environment > Session Setting**.  
The **Session Setting** pane is displayed.
- Step 3** Select the required Data Studio encoding from **Data Studio Encoding** drop-down.
- Step 4** Select the file encoding from **File Encoding** field.

**NOTE**

Data Studio supports only UTF-8 and GBK file encoding types.

- Step 5** Click **OK**. The **Restart Data Studio** dialog box is displayed.
- Step 6** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then Data Studio displays the **Process is running** dialog box.

**Step 7** Click **Force Restart** to discard operations and restart Data Studio. Click **OK** to continue performing operations.

 **NOTE**

- Click **Restore Defaults** from **Session Setting** pane to reset to default values. The default value for Data Studio Encoding and File Encoding is **UTF-8**.
- Only UTF-8 and GBK encoding is supported by GaussDB 100 database.

----End

## SQL Assistant

Follow the steps below to enable/disable SQL Assistant tool:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Environment > Session Setting**.

The **Session Setting** pane is displayed.

**Step 3** Select **Enable/Disable** from SQL Assistant section.

**Step 4** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Session Setting** pane to reset to default value. The default value for SQL Assistant is **Enable**.

----End

## Query/Function/Procedure Backup

Refer to the [Backup Unsaved Queries/Functions/Procedures](#) section for information on backup feature provided by Data Studio.

Follow the steps below to enable/disable backup of unsaved data in SQL Terminal/PL/SQL Viewer:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Environment > Session Setting**.

The **Session Setting** pane is displayed.

**Step 3** Select/unselect **Auto Save** from **Auto Save** section.

**Step 4** Set the time interval to backup the data in **Interval** field.

**Step 5** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Session Setting** pane to reset to default value. Backup of data will be enabled by default with 5 minutes as the default time interval.

----End

Follow the steps below to enable/disable data encryption of saved data:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Environment > Session Setting**.

The **Session Setting** pane is displayed.

**Step 3** Select/unselect **Encryption** from Auto Save section.

**Step 4** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Session Setting** pane to reset to default value. Encryption will be enabled by default.

----End

## 7.5 Export

This section provides details on how to personalize export DDL operation.

### Export DDL

You can set preference to include tablespace in DDL while exporting DDL using the **Export DDL** setting.

Set include tablespace in DDL:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Export/Import > Export DDL**.

The **Export DDL** pane is displayed.

**Step 3** Select **Include Tablespace in DDL** to include the tablespace while exporting DDL.

**Step 4** Click **OK**.

 **NOTE**

- Click **Restore Defaults** from **Export DDL** pane to reset to default values. The default value is **Include Tablespace in DDL**.
- This feature is not supported by GaussDB 100.

----End

## 7.6 Result Management

This section provides details on how to personalize the column width, number of records to be fetched in the query results, and result copy of column header or row number using the **Query Results** setting. It also provides details on how to display the data encoding in edit, view and query results table.

## Query Results

Set column width of query results:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Result Management > Query Results**.

The **Query Results** pane is displayed.

**Step 3** Select the required option.

Column Width customization options:

Option	Outcome
Content Length	Selecting this option enables you to set the column width based on the content length of the column.
Custom Length	Selecting this option enables you to set the column width based on the value entered in this field. <b>NOTE</b> This column accepts value between 100 and 500.

**Step 4** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Query Results** pane to reset to default values. The default value is **Content Length**.

----End

Set the number of records to be fetched in the query results:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Result Management > Query Results**.

The **Query Results** pane is displayed.

**Step 3** Select the required option.

Option	Outcome
Fetch All records	Selecting this option enables you to fetch all the records in the query results.
Fetch custom number of records	Selecting this option enables you to set the number of records that needs to be fetched in the query results. <b>NOTE</b> This column accepts value between 100 and 5000.

**Step 4** Click **OK**.



Click **Restore Defaults** from **Query Results** pane to reset to default values. The default value is **Fetch custom number of records (1000)**.

----**End**

Set preference to copy column name and row number from query results:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Result Management > Query Results**.

The **Query Results** pane is displayed.

**Step 3** Select the required option.

Option	Outcome
Include column header	Selecting this option enables you to copy column headers from the query results.
Include row number	Selecting this option enables you to copy row numbers from the query results.

**Step 4** Click **OK**.



Click **Restore Defaults** from **Query Results** pane to reset to default values. The default value is **Include column header**.

----**End**

## Edit Table Data

Set save behavior of edit table data operation:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Result Management > Edit Table Data**.

The **Edit Table Data** pane is displayed. Select the required option :

**Table 7-1** Edit table data

Server Type	Auto Commit	Reuse Connection	Table Data Save Option	Behavior
GaussDB 200, GaussDB 300, and DWS	ON	ON	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 200, GaussDB 300, and DWS	ON	ON	Do Not Save	If an error occurs , no data will be saved.
GaussDB 200, GaussDB 300, and DWS	ON	OFF	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 200, GaussDB 300, and DWS	ON	OFF	Do Not Save	If an error occurs , no data will be saved.
GaussDB 200, GaussDB 300, and DWS	OFF	ON	Save Valid Data	If an error occurs, no data will be saved. Perform Commit/ Rollback to proceed further.
GaussDB 200, GaussDB 300, and DWS	OFF	ON	Do Not Save	If an error occurs, no data will be saved. Perform Commit/ Rollback to proceed further.
GaussDB 100 Cluster	ON	ON	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 100 Cluster	ON	ON	Do Not Save	If an error occurs , no data will be saved.

Server Type	Auto Commit	Reuse Connection	Table Data Save Option	Behavior
GaussDB 100 Cluster	ON	OFF	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 100 Cluster	ON	OFF	Do Not Save	If an error occurs , no data will be saved.
GaussDB 100 Cluster	OFF	ON	Save Valid Data	All the valid data will be saved.If any error occurs it roll backs the entire transaction and it saves valid data further.
GaussDB 100 Cluster	OFF	ON	Do Not Save	If an error occurs , no data will be saved.
GaussDB 100 Standalone	ON	ON	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 100 Standalone	ON	ON	Do Not Save	If an error occurs , no data will be saved.
GaussDB 100 Standalone	ON	OFF	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
GaussDB 100 Standalone	ON	OFF	Do Not Save	If an error occurs , no data will be saved.
GaussDB 100 Standalone	OFF	ON	Save Valid Data	All the valid data will be saved. Incorrect data will be omitted.

Server Type	Auto Commit	Reuse Connection	Table Data Save Option	Behavior
GaussDB 100 Standalone	OFF	ON	Do Not Save	If an error occurs , no data will be saved.

**Step 3** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Edit Table Data** pane to reset to default values. The default value is **Save Valid Data**.

----End

## Result Data Encoding

You can enable/disable to display the data encoding type in edit, view, and query results window.

Follow the steps below to modify display of encoding option:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Result Management > Query Results**.

The **Query Results** pane is displayed.

**Step 3** Select **Include result data encoding** to include the **Encoding** drop-down in edit, view, and query results table.

**Step 4** Click **OK**.

 **NOTE**

- Click **Restore Defaults** from **Result Management** pane to reset to default values. **Include result data encoding** will be unselected by default.
- Edit table, view table properties and query execution must be performed again to apply the changes.

----End

## 7.7 Security

This section provides details on how to personalize password and security disclaimer display.

### Save Password Permanently

You can enable/disable to display the permanent option to save password in the connection window.

Follow the steps below to modify display of permanent save password option:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Security > Password**.

The **Password** pane is displayed.

**Step 3** Select the required option. Refer table below to understand the customization options available:

Option	Outcome
Yes	Selecting this option enables you to view the "Permanently" save password option from the <b>Save Password</b> drop-down in the connection window.
No	Selecting this option removes the "Permanently" save password option from the <b>Save Password</b> drop-down in the connection window. Selecting this option removes the saved passwords.

**Step 4** Click **OK**. The **Restart Data Studio** dialog box is displayed.

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then Data Studio displays the **Process is running** dialog box.

**Step 6** Click **Force Restart** to discard operations and restart Data Studio. Click **OK** to continue performing operations.

 **NOTE**

Click **Restore Defaults** from **Password** pane to reset to default values. The default value is **No**.

----End

## Password Expiry

This section provides details on how to continue/discontinue working with Data Studio once password expires using the Password setting.

 **NOTE**

This feature is not supported by GaussDB 100.

Follow the steps below to modify the behavior of Data Studio once password expires:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Security > Password**.

The **Password** pane is displayed.

**Step 3** Select the required option. Refer table below to understand the customization options available:

Option	Outcome
Yes	<p>Selecting this option allows you to login to Data Studio after the password has expired.</p> <p><b>NOTE</b> A message displays informing you that the password has expired and some operations may not work as expected in the following scenarios:</p> <ul style="list-style-type: none"> <li>● Establishing a new connection.</li> <li>● Editing a connection.</li> <li>● Connecting to a database while creating the database when no other database is connected in that connection profile.</li> <li>● Connecting to a database when no other database is connected in that connection profile.</li> </ul>
No	<p>Selecting this option will not allow you to login to Data Studio once the password has expired. A message displays informing you that the password has expired.</p>

**Step 4** Click **OK**. The **Restart Data Studio** dialog box is displayed.

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then Data Studio displays the **Process is running** dialog box.

**Step 6** Click **Force Restart** to discard operations and restart Data Studio. Click **OK** to continue performing operations.

 **NOTE**

The default value is **Yes**.

---**End**

## Security Disclaimer

You can enable/disable to display the security disclaimer for any unsecured connection/file operations.

Follow the steps below to modify the display of security disclaimer:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Security > Security Disclaimer**.

The **Security Disclaimer** pane is displayed.

**Step 3** Select the required option. Refer table below to understand the customization options available:

Option	Outcome
Enable	<p>Selecting this option displays the security disclaimer each time you try to establish an unsecure connection or perform a file operation.</p>

Option	Outcome
Disable	Selecting this option will not display the security disclaimer while establishing an unsecure connection or performing a file operation. You need to agree to the security implications that may arise due to unsecure connection.

**Step 4** Click **OK**. The **Restart Data Studio** dialog box is displayed.

**Step 5** Click **Yes** to restart Data Studio. If any export, import or execution operations are in progress, then Data Studio displays the **Process is running** dialog box.

**Step 6** Click **Force Restart** to discard operations and restart Data Studio. Click **OK** to continue performing operations.

 **NOTE**

Click **Restore Defaults** from **Security Disclaimer** pane to reset to default values. The default value is **Enable**.

----End

# 8 References

## 8.1 Performance Specification

### 8.1 Performance Specification

Data Studio's performance to load and operate on the object browser directly depends on the number of objects to be loaded. These include tables, views, columns and so on.

The memory consumption also depends on the number of objects loaded.

To improve the performance of loading objects and of memory usage efficiency, it is recommended to split the objects across multiple namespaces and avoid having skewed namespaces with a very large number of objects. By default, Data Studio loads the namespaces in the *search\_path* set for the user logged in. Other namespaces and the contained objects are loaded only when needed.

To improve performance load all objects rather than loading them based on user privilege. **Minimum Privileges Requirement** table provides information on the minimum access required for objects to be listed in Object Browser.

**Table 8-1** Minimum Privileges Requirement

Object Type	Privilege Types	Object Browser - Minimum Privilege Type
Database	Create, Connect, Temporary/Temp, All	Connect
Schema	Create, Usage, All	Usage
Table	Select, Insert, Update, Delete, Truncate, References, All	Select
Column	Select, Insert, Update, References, All	Select

Object Type	Privilege Types	Object Browser - Minimum Privilege Type
View	Select, Insert, Update, Delete, Truncate, References, All	Select
Sequences	Usage, Select, Update, All	Usage
Function	Execute, All	Execute
Tablespace	Create, All	Create

To improve performance of **Find/Find and Replace** operation, it is recommended to split lines that have more than 10000 characters in a single line into multiple smaller lines.

The following observations and outcomes of the performance tests done will help understand the performance aspects of Data Studio more clearly:

Recommended maximum memory that can be configured (Current release version)		1.4 GB
Performance with 150K tables and 150K views with 3 columns each at maximum memory configuration:		
>	Time taken to refresh the namespace in the object browser	15s
>	Time taken for initial loading and expanding of all table/views in the object browser	90s-120s
>	Time taken for subsequent loading and expanding of all table/ views in the object browser	<10s
>	Total Memory occupied	700 MB

 **NOTE**

This representative performance data is included for reference only. Actual performance may vary depending on the usage scenarios.

# 9 Troubleshooting

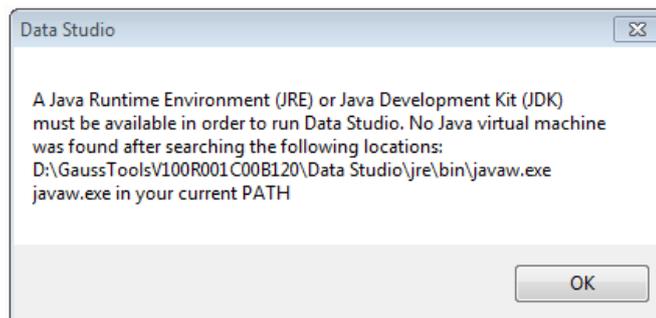
1. **Data Studio does not open. What do I do?**

**Solution:** Check whether JRE is missing. Verify the configured Java path in the environment. Refer to [3.5 System Requirements](#) for supported Java JDK version.

2. **Data Studio does not open and displays a 'Java Runtime' error when I double-click the *Data Studio.exe* file. What do I do?**

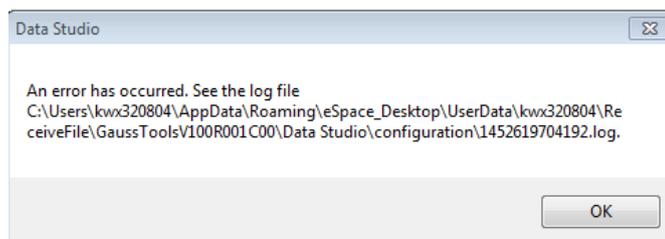
**Solution:**

- For no JRE:



Check whether the Java Runtime Environment (JRE) or Java Development Kit (JDK) version 1.8.0\_141 or above with appropriate bit number is installed on the system and Java Home path is set. If there are more than one version of Java installed, then set the -vm parameter in the configuration file. Refer to the [4.1 Installing and Configuring Data Studio](#) section to set this parameter. This is a prerequisite for running Data Studio.

- For older versions of JRE:



Check the version of Java Runtime Environment (JRE) or Java Development Kit (JDK) that is installed on the system. An older version installed on the system

causes this error. Update the JRE to version 1.8.0\_141 or above with appropriate bit number.

- Java Incompatibility:



```
Data Studio

Java was started but returned exit code=13
C:\ProgramData\Oracle\Java\javapath\javaw.exe
-Dosgi.requiredJavaVersion=1.8
-Xms40m
-Xmx1200m
-Dfile.encoding=UTF8
-Dds.encoding=UTF8
-jar C:\Users\sWX436505\Documents\Docs\English\DS\September 2017
Release\Local
Binary\Sequence\eclipse\plugins/org.eclipse.equinox.launcher_1.3.100.v2015051
1-1540.jar
-os win32
-ws win32
-arch x86_64
-showsplash
-launcher C:\Users\sWX436505\Documents\Docs\English\DS\September 2017
Release\Local Binary\Sequence\eclipse\Data Studio.exe
-name Data Studio
--launcher.library C:\Users\sWX436505\Documents\Docs\English\DS\September
2017 Release\Local
Binary\Sequence\eclipse\plugins/org.eclipse.equinox.launcher.win32.win32.x86_
64_1.1.300.v20150602-1417\eclipse_1611.dll
-startup C:\Users\sWX436505\Documents\Docs\English\DS\September 2017
Release\Local
Binary\Sequence\eclipse\plugins/org.eclipse.equinox.launcher_1.3.100.v2015051
1-1540.jar
--launcher.overrideVmargs
-exitdata 2944_80
-clearPersistedState
-fetchResultData=1000
-viewTableFetchSize=200
-consoleLineCount=5000
-orderByColumn=-1
-enablePermanentPasswordSaveOption=false
-enableSecurityWarning=true
-logfolder=.
-loginTimeout=180
-data @none
-focusOnFirstResult=false
-vm C:\ProgramData\Oracle\Java\javapath\javaw.exe
-vmargs
-Dosgi.requiredJavaVersion=1.8
-Xms40m
-Xmx1200m
-Dfile.encoding=UTF8
-Dds.encoding=UTF8
-jar C:\Users\sWX436505\Documents\Docs\English\DS\September 2017
Release\Local
Binary\Sequence\eclipse\plugins/org.eclipse.equinox.launcher_1.3.100.v2015051
1-1540.jar

OK
```

Check the version of Java Runtime Environment (JRE) or Java Development Kit (JDK) that is installed on the system. Incompatible Java bit version installed on the

system causes this error. Update the JRE version to 1.8.0\_141 or above with appropriate bit number.

It is recommended to run the batch file to check compatibility and launch Data Studio. Refer to [5.1 Starting Data Studio](#) for more information.

3. **While running the *StartDataStudio.bat* file the following message displays. What do I do?**

**Solution:**

Message	Solution
You are attempting to run 32-bit Data Studio on: <ul style="list-style-type: none"><li>● 64 bit OS</li><li>● Microsoft Windows 7 Professional</li><li>● Java 1.8 64-bit JDK (Incompatible)</li></ul> Install Java 1.8 32-bit	Install Java 1.8 32-bit
Data Studio is supported with minimum Java Version of 1.8 Install Java version 1.8 in order to use Data Studio	Install Java version 1.8 with appropriate bit number
You are attempting to run 64-bit Data Studio on: <ul style="list-style-type: none"><li>● 64 bit OS</li><li>● Microsoft Windows 7 Professional</li><li>● Java 1.8 32-bit JDK (Incompatible)</li></ul> Install Java 1.8 64-bit	Install Java 1.8 64-bit
You are attempting to run 64-bit Data Studio on: <ul style="list-style-type: none"><li>● 32 bit OS</li><li>● Microsoft Windows 7 Professional</li><li>● Java 1.8 32-bit JDK (Incompatible)</li></ul> Install 32-bit Data Studio	Install 32-bit Data Studio

4. **Why does Data Studio not connect to the server even with all valid inputs?**

**Solution:** Check whether the server is running in the specified IP and port. Check for availability of the specified user by connecting through **gsqL**.

5. **What do I do for connection issues while using Data Studio?**

**Solution:** A connection issue that may occur while using Data Studio is explained with an example:

Establish a database connection.

Run the query.

When a connection exception occurs in any one of the database (**PostgreSQL**), the connection is closed. When the database connection is closed, all the function and procedure tabs, if open, will be closed too.

An error will be seen and the **Object Browser** will show the status of the database.

 **NOTE**

Only the current database will be disconnected. Other databases will remain connected and re-connection is possible.

Re-connect to the database to proceed with execution.

6. **While fetching a function/procedure containing Chinese comments through a Java application, the Chinese characters are not visible. What do I do?**

**Solution:** Set **Preferences > Session Setting > Data Studio Encoding and File Encoding** to *GBK*, so that Chinese characters are displayed properly.

7. **Connecting to large database, loading large number of queries into SQL Terminal, may result into 'Out of Memory' error or 'Java Heap Space' error. What is the solution?**

**Solution:** 'Out of Memory' error or 'Java Heap Space' error occurs when Data Studio has used up the maximum allocated Java memory. By default, the configuration file *Data Studio.ini* (located in the Data Studio install path) contains the entry "**-Xmx1200m**", where 1200m denotes 1200MB as the maximum Java memory that can be used by Data Studio. The memory usage is based on the size of data fetched by Data Studio.

To resolve this issue, increase the Java memory size to the desired value. For example, update "**-Xmx1200m**" to "**-Xmx2000m**" and restart Data Studio. If the updated memory size is used up as well, the same issue might reoccur.

 **NOTE**

- As an example for 32-bit Data Studio and 8GB RAM the value of the Xmx parameter must not cross 2044MB and for 64-bit Data Studio and 8GB RAM the value of the Xmx parameter must not cross 6000MB. The limit may vary based on user's current memory usage.

For example:

-Xms1024m

-Xmx1800m

- The maximum file size that Data Studio can support in the SQL Terminal is based on the value of the Xmx parameter in the *Data Studio.ini* file and available memory.

8. **While executing an SQL query which returns a large amount of data, Data Studio displays an 'Insufficient Memory' error. What do I do?**

**Solution:** Data Studio will disconnect the connection profile. Re-establish the connection and continue execution.

9. **While exporting DDL or data why do I get export failed message?**

**Solution:** This could happen due to any of the following reasons:

- Invalid file for Client SSL Certificate and/or Client SSL Key have been selected. Select the correct file and try again. Refer to [6.2.2 Adding a Connection](#) on establishing connection.
- Identity of the object in the database could have been changed. Check if the identity of the object has been changed and try again.
- You may have insufficient privileges. Contact the database administrator to obtain appropriate privileges.

10. **While performing Show DDL operation why do I get show DDL failed message?**

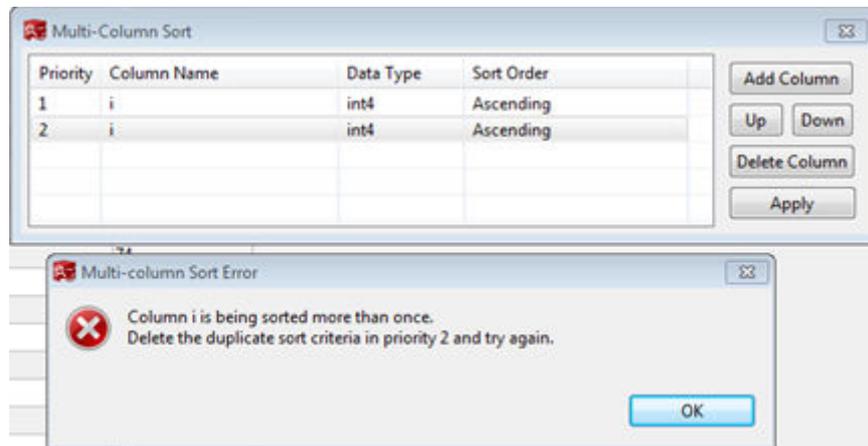
**Solution:** This could happen due to any of the following reasons:

- Invalid file for Client SSL Certificate and/or Client SSL Key have been selected. Select the correct file and try again. Refer to [6.2.2 Adding a Connection](#) on establishing connection.
  - Identity of the object in the database could have been changed. Check if the identity of the object has been changed and try again.
  - You may have insufficient privileges. Contact the database administrator to obtain appropriate privileges.
11. **While performing Show DDL or Export DDL operation, why do I get this error message?**  
**"Failed to start the program, because MSVCRT100.dll is missing. Try reinstalling the program to solve the problem?"**  
**Solution:** This is because the operation requires `gs_dump.exe` to execute, and it requires the Microsoft VC runtime library `msvcrt100.dll`.  
To solve this, copy the `msvcrt100.dll` file from the `\Windows\System32` folder to `\Windows\SysWOW64` folder.
12. **Why does the saved connection profile details not show when I try to establish a connection?**  
**Solution:** This could happen if the *Profile* folder under *User Data* folder is not available or modified manually. Make sure the *Profile* folder is present with correct naming convention.
13. **When I close and reopen Data Studio the SQL query history information is lost. Why does this happen?**  
**Solution:** This could happen if the *Profile* folder is missing under *User Data* folder or the folder has been modified. Make sure the *Profile* folder is present with correct naming convention.
14. **When I try to modify any preference error saving preference message displays. Why do I get this error message?**  
**Solution:** This can happen if the *Preferences* folder is not present or renamed. Restart Data Studio to resolve this issue.
15. **What do I do when Data Studio becomes idle and *Data Studio.log* file states "No more handles"?**  
**Solution:** Restart Data Studio.
16. **What happens if an error occurs after I have edited a table and I am unable to make further changes?**  
**Solution:** All previously edited data will be lost. Close the **Edit Table Data** window and perform the changes again.
17. **Why do I keep getting the message "The number of pasted cell and the selected cell does not match" even though I have made the correct number of cell selection?**  
**A:** This can happen if the settings in **Preferences > Query Results** is set to include column header. The selected cells includes the column header cells as well. Modify the settings to disable include column header option and try again.
18. **Why am I not able to edit the temp table with Reuse Connection option off?**  
**A:** Turning off Reuse Connection option creates new session. Created temp tables are available for current session only. Turn on Reuse Connection option to edit temporary

tables. Refer to [6.12.12 Managing SQL Terminal Connections](#) section for more information.

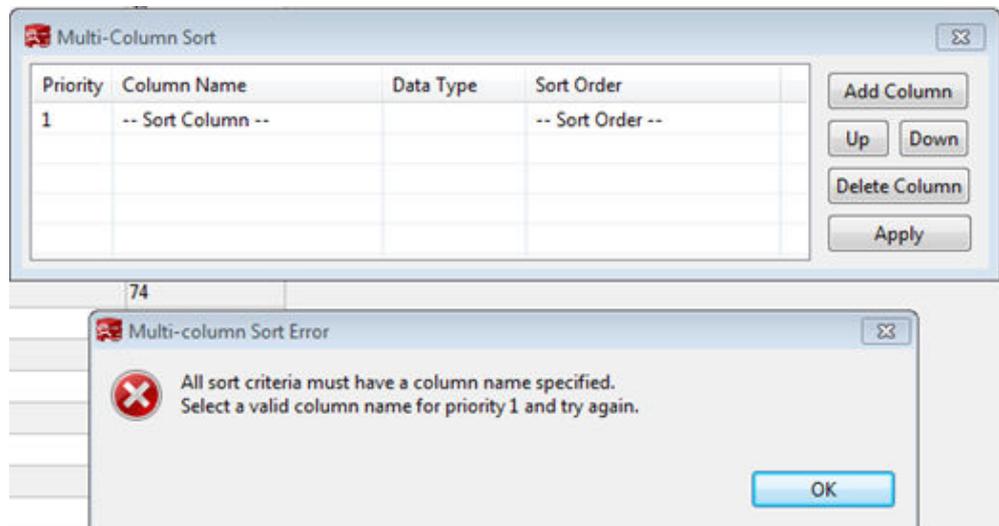
19. What happens if same column is added more than once in Multi-Column sort pop-up?

A: When same column is added more than once in multi-sort pop up table, and **Apply** is clicked, a notification is displayed as follows. You need to click **OK** and select the correct column to sort.



20. What happens if a column name is not selected in at least one of the sort criteria and **Apply** is clicked?

A: Following notification is displayed. Once you select a valid column name and click **Apply** again, this notification is not displayed.



21. What happens when you click on cancel when multiple create table queries are running in SQL terminal window.

A: Cancelling queries might throw an error in console displaying the table name that is not created. In that case it is recommended to drop that particular table in order to perform operation on a table with the same name.

22. When the user is not able to logon to DS due to security keys are compromised?

**Solution:** Follow the steps to generate the new security keys.

- a. Delete the security folder from **Datastudio folder > Userdata > Security folder**.

- b. Restart Datastudio.
- c. New security folder is created and the keys will be regenerated.
- d. Saved password will be lost and user should re-enter the password to logon to Datastudio.

---

# 10 Security Management

---

- [10.1 Overview](#)
- [10.2 Login History](#)
- [10.3 Password Expiry Notification](#)
- [10.4 Securing the Application In-Memory Data](#)
- [10.5 Data Encryption for Saved Data](#)
- [10.6 SQL History](#)
- [10.7 SSL Certificates](#)
- [10.8 Verify Software Package Integrity](#)

## 10.1 Overview

---

**NOTICE**

Ensure that the operating system and the required software's (refer to [System Requirements](#) for more details) are updated with the latest patches to prevent vulnerabilities and other security issues.

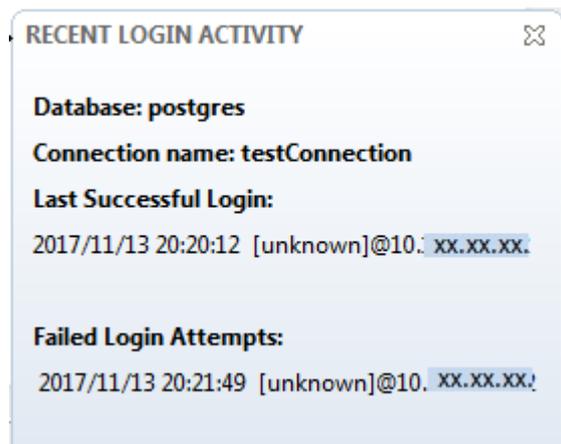
---

This section provides the security management information for Data Studio.

## 10.2 Login History

The following information is critical to manage security for Data Studio:

When you log into the database, Data Studio displays a pop-up with details of the last successful login and failure attempts between the last two successful logins for you on the logged database.



 **NOTE**

- If the pop-up displays the message "Last login details not available", then it implies that the connected database does not support the last login display feature.
- This feature is not available for GaussDB 100 database.

## 10.3 Password Expiry Notification

The following information is critical to manage security for Data Studio:

- Your password will expire within 7 days from the date of notification. If the password expires, contact the database administrator to reset the password.
- The password must be changed every 90 days.

## 10.4 Securing the Application In-Memory Data

The following information is critical to manage security for Data Studio:

While running Data Studio in trusted environment, user must ensure to prevent malicious software to scan or access the memory which is used to store application data including sensitive information.

Alternatively, you can choose Do Not Save while connecting to the database, so that password does not get saved in the memory.

## 10.5 Data Encryption for Saved Data

The following information is critical to manage security for Data Studio:

You can ensure encryption of auto saved data by enabling encryption option from **Preferences** page. Refer to [Query/Function/Procedure Backup](#) section for steps to encrypt the saved data.

## 10.6 SQL History

The following information is critical to manage security for Data Studio:

- SQL History scripts are not encrypted.
- The **SQL History** list does not display sensitive queries that contain the following keywords:
  - Alter Role
  - Alter User
  - Create Role
  - Create User
  - Identified by
  - Password

Few query syntax examples are listed below:

- ALTER USER name [[ WITH ] option [ ... ]]
- CREATE USER name [ [ WITH ] option [ ... ]]
- CREATE ROLE name [ [ WITH ] option [ ... ]]
- ALTER ROLE name [ [ WITH ] option [ ... ]]

## 10.7 SSL Certificates

### NOTICE

The information on using SSL certificates is included only for reference purposes. For details on the certificates and for security guidelines for managing the certificates and related files, refer the database server documentation.

Data Studio can connect to the database using the Secure Sockets Layer [SSL] option. The following files are required to [add a connection](#).

#	Certificate/Key	Description
1	Client SSL Certificate	Provided by System/Database Administrator
2	Client SSL Key	Provided by System/Database Administrator
3	Root Certificate	Provided by System/Database Administrator

## SSL Certificate Generation and Server Configuration

Follow the steps to generate the certificate:

**Step 1 Establish a CA environment-** Assume that user **omm** has been created and the CA path is **test**.

Log in to SUSE Linux as user **root** and switch to user **omm**.

Execute the following command:

```
mkdir test
cd /etc/ssl
```

Copy the configuration file **openssl.cnf** to **test**.

Command:

```
cp openssl.cnf ~/test
cd ~/test
```

Establish the CA environment under the **test** folder.

Create folder in **demoCA./demoCA/newcerts./demoCA/private** path.

Command:

```
mkdir ./demoCA ./demoCA/newcerts ./demoCA/private
chmod 777 ./demoCA/private
```

Create the **serial** file and write it to **01**.

Command:

```
echo '01'>./demoCA/serial
```

Create the **index.txt** file.

Command:

```
touch /home/omm/test/demoCA/index.txt
```

Modify parameters in the **openssl.cnf** configuration file.

Command:

```
dir = /home/omm/test/demoCA
default_md = sha256
```

The CA environment has been established.

## Step 2 Generate a root private key - Generate a CA private key.

Command:

```
openssl genrsa -aes256 -out demoCA/private/akey.pem 2048
```

Generating RSA private key, 2048 bit long modulus.

## Step 3 Generate a root certificate request file- CA root certificate application file named **server.req**.

Command:

```
openssl req -config openssl.cnf -new -key demoCA/private/akey.pem -out demoCA/careq.pem
```

Enter pass phrase for **demoCA/private/akey.pem**

Enter the root private key password **Gauss@MppDB**.

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank.

For some fields there will be a default value,enter '.' to leave the field blank. Enter the following information in the generated server certificate and client certificate.

```
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:shanxi
Locality Name (eg, city) []:xian
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Abc
Organizational Unit Name (eg, section) []:hello
-Common name can be any name
Common Name (eg, YOUR name) []:world
-Email is optional.
Email Address []:
A challenge password []:
An optional company name []:
```

#### Step 4 Generate a self-signed root certificate.

Command:

```
openssl ca -config openssl.cnf -out demoCA/cacert.pem -keyfile demoCA/private/
cakey.pem -selfsign -infiles demoCA/creq.pem
```

Using configuration from openssl.cnf

Enter pass phrase for **demoCA/private/cakey.pem**

Enter the root private key password **Gauss@MppDB**.

Check that the request matches the signature.

```
Signature ok
Certificate Details:
Serial Number: 1 (0x1)
Validity
Not Before: Feb 28 02:17:11 2017 GMT
Not After : Feb 28 02:17:11 2018 GMT
Subject:
countryName = CN
stateOrProvinceName = shanxi
organizationName = Abc
organizationalUnitName = hello
commonName = world
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
F9:91:50:B2:42:8C:A8:D3:41:B0:E4:42:CB:C2:BE:8D:B7:8C:17:1F
X509v3 Authority Key Identifier:
keyid:F9:91:50:B2:42:8C:A8:D3:41:B0:E4:42:CB:C2:BE:8D:B7:8C:17:1F
Certificate is to be certified until Feb 28 02:17:11 2018 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
```

A CA root certificate named **demoCA/cacert.pem** has been issued.

#### Step 5 Generate a private key for the server certificate- Generate a private key file named **server.key**.

Command:

```
openssl genrsa -aes256 -out server.key 2048
```

#### Step 6 Generate a server certificate request file- Generate a server certificate request file **server.req**.

Command:

```
openssl req -config openssl.cnf -new -key server.key -out server.req
```

Enter pass phrase for server.key:

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Set the following information and make sure that it is same as that when CA is created.

```
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:shanxi
Locality Name (eg, city) []:xian
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Abc
Organizational Unit Name (eg, section) []:hello
--Common name can be any name
Common Name (eg, YOUR name) []:world
Email Address []:
-- The following information is optional.
A challenge password []:
An optional company name []:
```

### Step 7 Generate a server certificate - Change the demoCA/index.txt.attr attribute to no.

```
vi demoCA/index.txt.attr
```

Issue the generated server certificate request file. After it is issued, an official server certificate **server.crt** is generated.

```
openssl ca -config openssl.cnf -in server.req -out server.crt -days 3650 -md
sha256
```

Using configuration from /etc/ssl/openssl.cnf

Enter pass phrase for ./demoCA/private/cakey.pem:

Check that the request matches the signature

```
Signature ok
Certificate Details:
Serial Number: 2 (0x2)
Validity
Not Before: Feb 27 10:11:12 2017 GMT
Not After : Feb 25 10:11:12 2027 GMT
Subject:
countryName = CN
stateOrProvinceName = shanxi
organizationName = Abc
organizationalUnitName = hello
commonName = world
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
EB:D9:EE:C0:D2:14:48:AD:EB:BB:AD:B6:29:2C:6C:72:96:5C:38:35
X509v3 Authority Key Identifier:
keyid:84:F6:A1:65:16:1F:28:8A:B7:0D:CB:7E:19:76:2A:8B:F5:2B:5C:6A
Certificate is to be certified until Feb 25 10:11:12 2027 GMT (3650 days)
-- Choose y to sign and issue the certificate.
Sign the certificate? [y/n]:y
-- Select y, the certificate signing and issuing is complete.
1 out of 1 certificate requests certified, commit? [y/n]y
```

```
Write out database with 1 new entries
Data Base Updated
```

**Enable password protection for the private key:** If the password protection of the server private key is not disabled, you need to use **gs\_guc** to encrypt the password.

```
gs_guc encrypt -M server -K Gauss@MppDB -D ./
```

After the password is encrypted using **gs\_guc**, two private key password protection files **server.key.cipher** and **server.key.rand** are generated.

### Step 8 Generate the client certificate and private key - Generate a client private key.

```
openssl genrsa -aes256 -out client.key 2048
```

Generate a certificate request file for a client.

```
openssl req -config openssl.cnf -new -key client.key -out client.req
```

After the generated certificate request file for client is signed and issued, the a formal client certificate **client.crt** is generated.

```
openssl ca -config openssl.cnf -in client.req -out client.crt -days 3650 -md sha256
```

#### NOTE

If **METHOD** is set to **cert** in the **pg\_hba.conf** file of the server, the client must use the **username** (common name) configured in the license file (**client.crt**) for the database connection. If **METHOD** is set to **md5** or **sha256**, the client does not have this restriction.

If password protection for a client private key is not removed, you need to use **gs\_guc** to encrypt the password.

```
gs_guc encrypt -M client -K Gauss@MppDB -D ./
```

After the password is encrypted using **gs\_guc**, two private key password protection, files **client.key.cipher** and **client.key.rand** are generated.

### Step 9 sdknfn.

----End

## Replacing Certificates

Default security certificates and private keys required for SSL connection are configured in LibrA. The formal certificates and keys for the server and client have been obtained from the CA.

### Step 1 Prepare for a certificate and a key. Conventions for configuration file names on the server:

```
l Certificate name: server.crt
l Key name: server.key
l Key password and encrypted file: server.key.cipher and server.key.rand
Conventions for configuration file names on the client:
l Certificate name: client.crt
l Key name: client.key
l Key password and encrypted file: client.key.cipher and client.key.rand
l Certificate name: cacert.pem
l Names of files on in the revoked certificate list: sslcrl-file.crl
```

### Step 2 Create a compressed package.

Package name: db-cert-replacement.zip

Package format: ZIP

Package file list: *server.crt*, *server.key*, *server.key.cipher*, *server.key.rand*, *client.crt*, *client.key*, *client.key.cipher*, *client.key.rand*, *cacert.pem*. If you need to configure the certificate revocation list (CRL), the list must contain *sslcr1-file.crl*.

Command:

```
zip db-cert-replacement.zip client.crt client.key client.key.cipher
client.key.rand server.crt server.key server.key.cipher server.key.rand
zip -u ../db-cert-replacement.zip cacert.pem
```

**Step 3** Invoke the certificate replacement interface to replace a certificate. Upload the prepared package *db-cert-replacement.zip* to any path of a cluster user. For example: */home/gaussdba/test/db-cert-replacement.zip*

Run the following command to perform the replacement in coordinator:

```
gs_om -t cert --cert-file=/home/gaussdba/test/db-cert-replacement.zip
```

Starting SSL cert files replace.

Backing up old SSL cert files.

Backup SSL cert files on BLR1000029898 successfully.

Backup SSL cert files on BLR1000029896 successfully.

Backup SSL cert files on BLR1000029897 successfully.

Backup gds SSL cert files on successfully.

BLR1000029898 replace SSL cert files successfully.

BLR1000029896 replace SSL cert files successfully.

BLR1000029897 replace SSL cert files successfully.

Replace SSL cert files successfully.

Distribute cert files on all coordinators successfully.

You can run the **gs\_om -t cert --rollback** command to remotely invoke the interface and **gs\_om -t cert --rollback -L**

----End

## 10.8 Verify Software Package Integrity

To prevent network security threats caused by malicious tampering or damage during installation package transfer, verify the integrity of the installation package after obtaining it. The deployment can be implemented only after the installation package passes the verification.

### NOTE

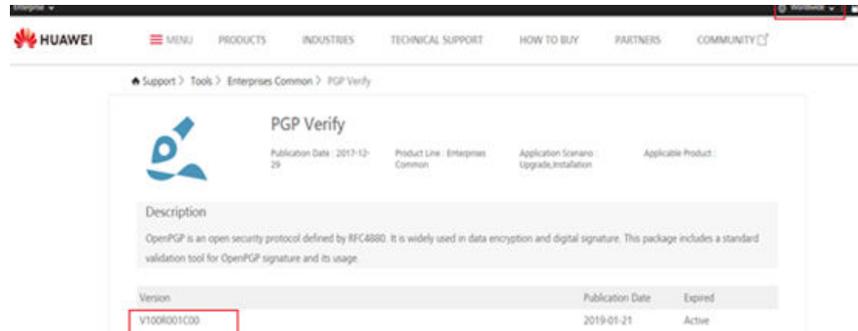
System must have internet access to carry out the verification.

#### Step 1 Obtaining PGPVerify Tool

Download ->PGPVerify Tool from the following uniform resource locator (URL): <http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL100000054>

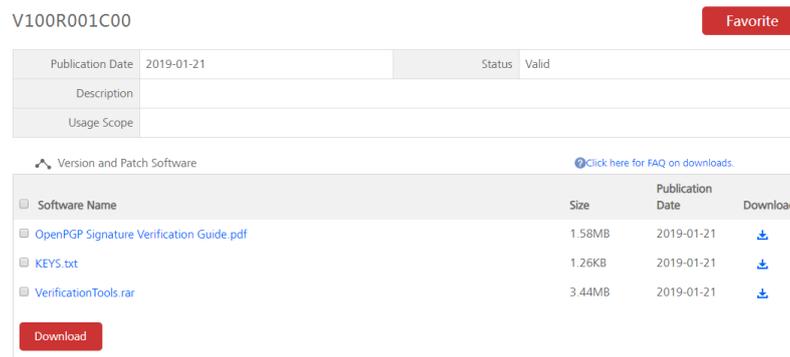
 **NOTE**

The target web page may be displayed in Chinese. To change the language to English, click the earth mark (Worldwide) on the top of the page, and then choose English. Now you can download the English document.



● **Download the PGPVerify tool (PGPVerify.exe)**

- a. Click the version number in the Version list.
- b. In the displayed page, click  corresponding to VerificationTools.rar to download this package.



- c. Unzip the VerificationTools.rar package and obtain the PGPVerify verification tool (PGPVerify.exe).

 **NOTE**

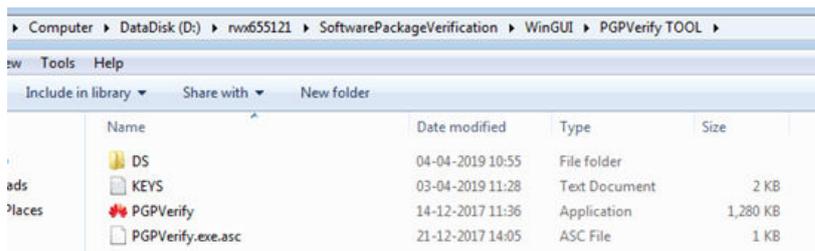
Downloaded file has the PGPVerify.exe for windows in the below path.



Copy and place it under a folder as per convenience.

**Step 2 Obtaining the Public Key File:**

1. Click the version number in the Version list.
2. In the displayed page, click  corresponding to KEYS.txt to download.



### Step 3 Obtaining the DS package and .asc file (Signature file)

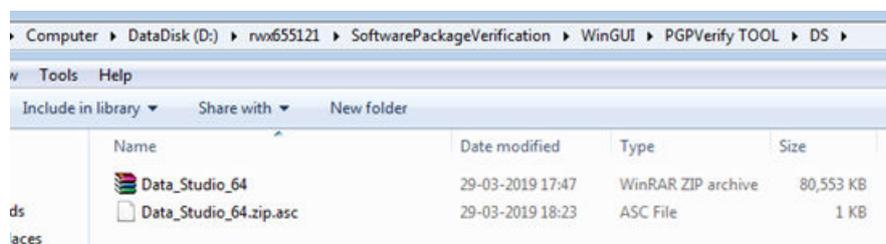
Download DataStudio Zip file and .asc file from VMP. Here is the path below:

Sample Path: \\blrnas03b-rd\IT\_VMP\_HTIPL\_DD\_06F\PB23885630\_Gauss Tools 18.2.0.SPC1.B500\Software\Data Studio

You can download the latest one to be tested.

#### NOTE

Create new folder (DS) and place both the **DataStudio package** and **.asc file**



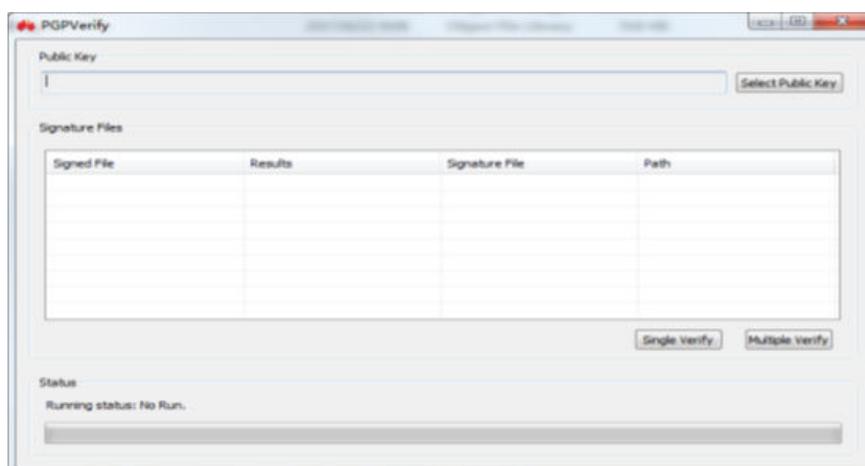
----End

## Verifying Software Integrity

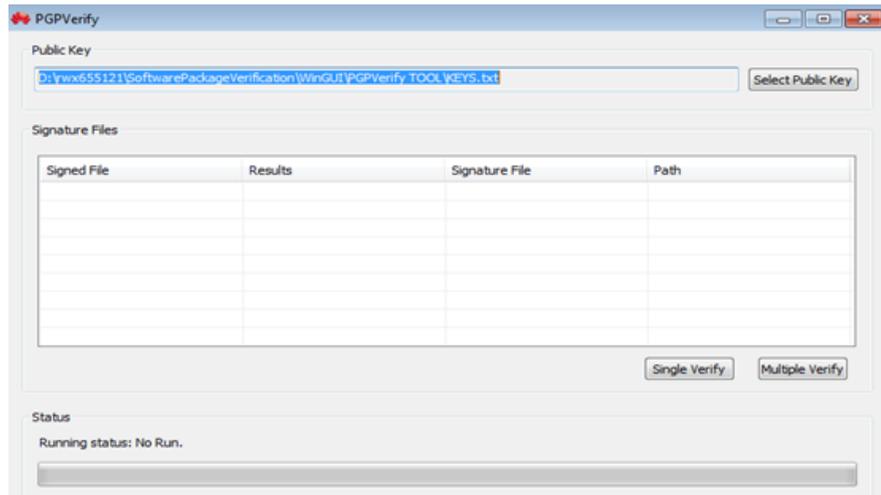
Signature file (.asc) must be in the same path of the software package (DS).

- **Single Verify:**

- a. Double-click PGPVerify.exe to start it,



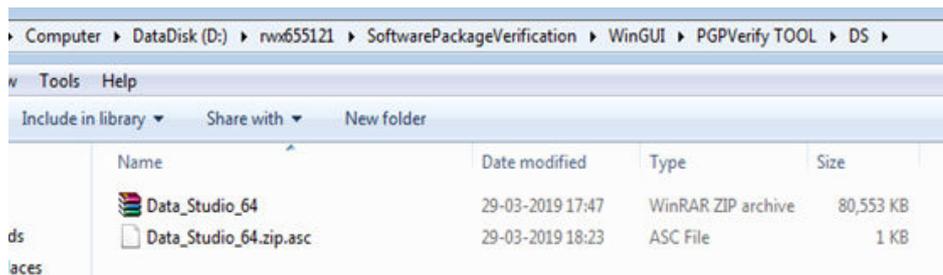
- b. Load the public key file (KEYS.txt).  
Click Select Public Key button to select the downloaded KEYS file.



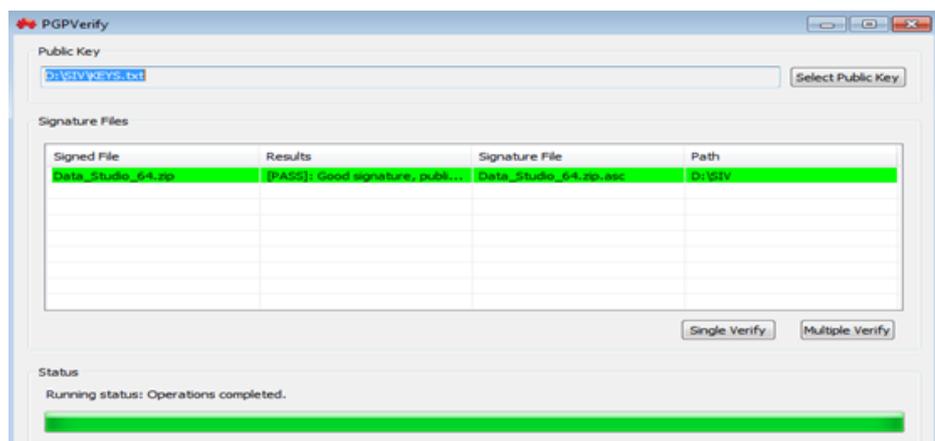
**NOTE**

If you have used this verification tool on the same computer before, the last key you selected will be automatically reloaded when you use this tool once again.

- c. Verify the software integrity using Single Verify. verification file of the software to be verified (Data\_Studio\_64.zip.asc).



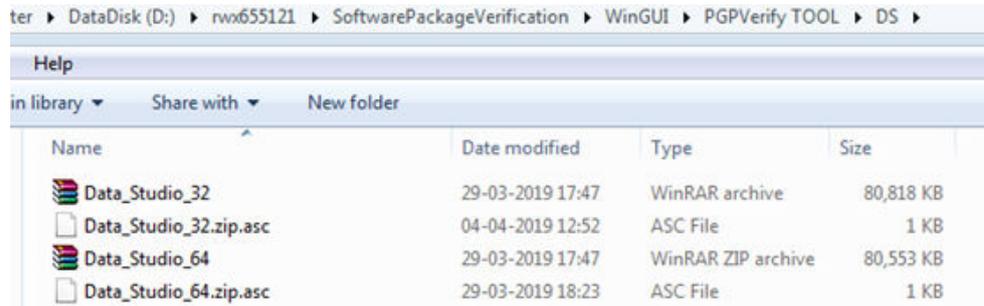
- d. Check the result.



- **Multiple Verify:** If you want to verify multiple packages. For eg: DataStudio\_32.zip & DataStudio\_64.zip, place them all (Software packages and its corresponding .asc files) under same folder (Says DS here).

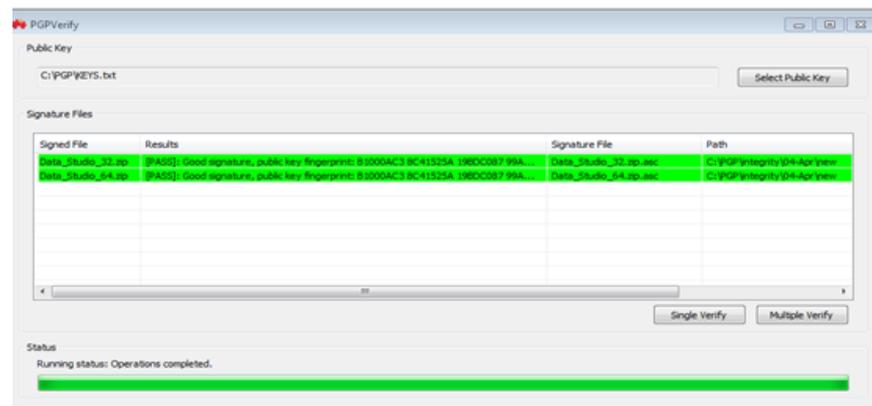
**NOTE**

Below image has 32 bit and 64 bit DS packages, so both files can be verified using multiple verify.



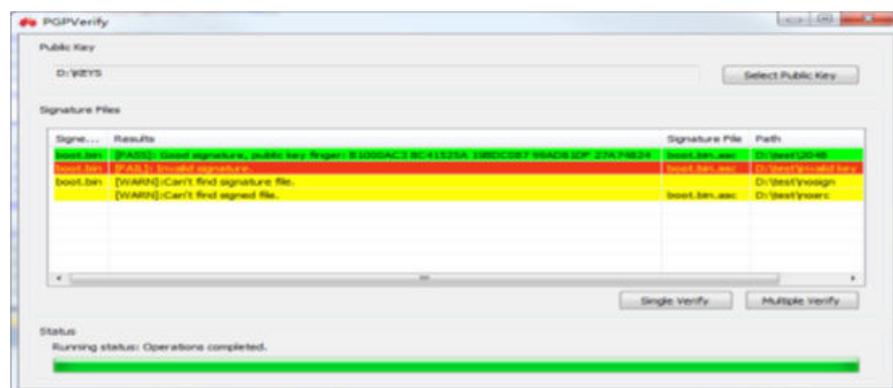
- Double-click PGPVerify.exe to start it.
- Load the public key file (KEYS.txt).
- Click Select Public Key to select the downloaded KEYS file
- Verify Software Integrity for multiple package using Multiple Verify  
Click Multiple Verify and select the folder which has packages and its .asc files (Says DS here).

Sample result:



● **PGPVerify Verification Result:**

Sample result:



If an item is highlighted in yellow and the Results column is WARN, it indicates that the signature cannot be verified.

If an item is highlighted in red and the Results column is FAIL, it indicates that the verification failed.

If an item is highlighted in green and the Results column is PASS, it indicates that the verification using the specified public key is succeeded.

---

# 11 FAQs

---

1. **What aspects must be checked in case of a connection failure?**

A: Check the following:

- Verify the **Connection Properties**, to check whether the input to the connection properties is correct.
- Check whether the server and client versions are compatible.
- Check whether *database\pg\_hba.conf* file is configured properly. Refer to the server manual for more details.
- Check whether *Data Studio.ini* file is configured properly.

2. **When I try to establish a second connection with a different server using the same SSL certificates, why is the connection successful?**

A: If the same SSL certificates are used by different server, then the second connection will be successful because the certificates are cached.

When you try to establish a second connection with a different server using different SSL certificates, the connection will fail because of a certificate mismatch.

3. **When I right-click on the function/procedure and perform 'Refresh' in the Object Browser, the function/procedure is not visible. What can be the reason?**

A: This may happen when you drop a function/procedure and recreate it. In this case, refresh the parent folder to view the function/procedure in the **Object Browser**.

4. **What action must be taken if a critical error occurs in a database session and is unable to proceed?**

A: Critical error can occur in some of the following cases. Check whether:

- The connection is left idle for long time and has timed out.
- The server is running or not.
- There is enough memory available on the server and no "out of memory" is reported on the server.

5. **What is a constraint?**

A: Constraints are used to restrict the insertion of unwanted data in columns. You can create constraints on single or multiple columns of any table. It maintains the data integrity of the table.

There are three types of constraints supported:

- Primary Key constraint
- Unique Key constraint
- Check constraint

6. **What is an index?**

A: An index is a copy of select columns of data from a table that can be searched very efficiently. It also includes a low level disk block address or a direct link to the complete row of data it was copied from.

7. **What is the default encoding for Data Studio's files?**

A: Exported, imported, and system files are encoded with the system's default encoding as configured in *Settings > Preferences*. The default encoding is UTF-8.

8. **When I try to open another instance of Data Studio multiple instances of Data Studio is not supported message displays. Why do I get this error message?**

A: Opening multiple instances of Data Studio by the same user is not supported.

9. **When I try to perform DDL operation on an object, the task keeps running indefinitely and I am unable to cancel the task. What could be the reason?**

A: This can happen if there is another active DML/DDL operation being performed on the same object. Close all active DML/DDL operations on the object and try again. If the problem still persists, it could be that another user might be performing a DML/DDL operation on that object. Try after sometime.

10. **Why is the exported query result different from the data available in the Results tab?**

A: For export result set data, the query is re-executed using a new connection. Hence the exported result may differ from the data shown in the **Results** tab.

11. **Why does last login information show "Last login details not available"?**

A: **Last login details not available** is shown, when you are connected to the older version of the database server or you have logged into the database for the first time after the database has been created.

12. **Why is the error marked incorrectly in the SQL Terminal?**

A: This happens when server returns the incorrect line number. Review the error message in **Messages** tab and navigate to the corresponding line number to fix the error.

13. **Do Show DDL and Export DDL display dropped column information?**

A: Yes, **Show DDL** and **Export DDL** operation will display the dropped column information.

14. **Why does Data Studio not launch after I have modified the -Xmx parameter?**

A: This happens if the value defined for -Xmx parameter may be invalid. Refer to [4.1 Installing and Configuring Data Studio](#).

15. **How can I access a Terminal quicker if I have opened multiple number of Terminals or tab?**

A: After the number of opened Terminals or tabs reaches a certain limit based on screen resolution an icon (  ) displays with a drop-down option at the end of the Terminal list. Click the  icon and select the required Terminal from the drop-down list. If the  is not available, then use the tooltip to identify the Terminal or tab. Terminal name can be searched by typing the search value above the list of SQL Terminal names.

**Example:**

- \*s, this displays all Terminal name that starts with s.
- test, this displays all Terminal name that starts with test.
- \*2, this displays all Terminal name that contains 2 in them.

16. **Why is that after I change the language DS restarts but the language does not change?**

**A:** Sometimes the language may not reflect the selected change post restart. Manually restart DS to open the tool in selected language.

17. **Why does the last login details information not display?**

**A:** At times the server returns an error while trying to fetch last login details. In such scenarios the last login pop-up message does not display.

18. **When viewing/exporting DDL, why does the Chinese text not show properly?**

**A:** This happens if the SQL, DDL, object names or data contains Chinese text and the Data Studio file encoding is not set to GBK. To solve this, go to Settings > Preferences > Environment > **File Encoding Preferences** and set the encoding to GBK.

The supported combinations of Database and Data Studio encoding for export operation are shown in **Table1 Supported combinations of file encoding**.

**To open/view the exported files in Windows Explorer:** Files exported with UTF-8 encoding can be opened/viewed by double-clicking it or by right-clicking on the file and selecting **Open**. Files exported with GBK encoding must be opened in MS-Excel® using the import external data feature (**Data > Get External Data > From Text**).

**Table 11-1** Supported combinations of file encoding used in the Database and Data Studio

Database Encoding	Data Studio File Encoding	Support for Chinese Text in Table Names	Support for English Text in Table Names
GBK	GBK	Yes	Yes
GBK	UTF-8	No - Incorrect details	No - Incorrect details
UTF-8	GBK	No - Export Fails	No - Incorrect details
UTF-8	UTF-8	Yes	Yes
UTF-8	LATIN1	No - Export Fails	Yes
SQL_ASCII	GBK	Yes	Yes
SQL_ASCII	UTF-8	No - Incorrect details	No - Incorrect details

19. **Why do I get the error message "Conversion between GBK and LATIN1 is not supported"?**

**A:** This message occurs if the Data Studio and Database encoding selected are incompatible. To solve this, select the compatible encoding. Compatible encoding is shown in [Table 11-2](#)

**Table 11-2** Compatible Encoding Formats

Data Studio Encoding	Database Encoding	Compatible
UTF-8	GBK	Yes
	LATIN1	Yes
	SQL_ASCII	Yes
GBK	UTF-8	Yes
	LATIN1	No
	SQL_ASCII	Yes
SQL_ASCII	UTF-8	Yes
	LATIN1	Yes
	GBK	Yes

20. **Why is the PL/SQL procedure I compiled and executed is saved as PL/SQL function?**

**A:** The database does not differentiate between PL/SQL function and procedure. According to the database all procedures are functions. Hence PL/SQL procedure is saved as PL/SQL function.

21. **Why is that I am not able to edit the distribution key?**

**A:** The database allows you to edit the distribution key only for the first insert operation.

22. **While editing table data if I do not enter a value for default value column, will the value be added by the database server?**

**A:** Yes, the database server will add the value but the value will not be visible after save in the Edit Table Data tab. Use the refresh option from the Edit Table Data tab or re-open the table again to view the added default value(s).

23. **While modifying/deleting table data why do I get a pop-up stating that more than one matching row found?**

**A:** This happens because there are additional rows detected for modification/deletion based on Custom Unique Key or All Columns selection. If Custom Unique Key is selected, then it will delete/modify the rows that have exact match of the data in the column selected for deletion/modification. If All Columns is selected, then it will delete/modify the rows that match data in all columns. Hence this duplicate records matching the Custom Unique Key or All Columns will be deleted/modified if Yes is selected. If No is selected, the row that is not saved will be marked for correction.

24. **When I right-click on a text box I see additional context menu options. Why does this happen?**

**A:** The additional context menu options like Right to left Reading order, Show Unicode control characters and so on are provided by Windows 7 in case the keyboard you are using supports right to left and left to right input.

25. **What are the objects that are not supported for batch export DDL & DDL and Data operations?**

**A:** Following objects are not supported for DDL & DDL and Data operations.

**Export DDL:**

Connection, database, tablespace, foreign table, sequence, column, index, constraint, partition, function/procedure group, regular tables group, views group, schemas group, and catalogs group.

**Export DDL and Data:**

Connection, database, tablespace, foreign table, sequence, column, index, constraint, partition, function/procedure, view, regular tables group, schemas group, and catalogs group.

26. **Will the queries in SQL Terminal commit if the resultset is modified and saved with Reuse Connection On and Auto Commit Off?**

**A:** No. Queries will only be committed when COMMIT command is executed in the Terminal.

Auto Commit	Reuse Connection	Resultset Save
On	On	Commit
On	Off	Commit
Off	On	Does not commit
Off	Off	Not supported

27. **When I query a temp table from a new SQL Terminal the resultset displays incorrect table details. Why does this happen?**

**A:** When you query a temp table from a new SQL Terminal or with the **Reuse Connection Off**, the resultset displays information of a regular/partition/foreign table, if a table with the same name as the temp table exists.

 **NOTE**

If the **Reuse Connection** is **On**, the resultset displays information of the temp table even if another table with the same name exists.

28. **Which are the operations that are performed on a locked object does not run in the background but needs to be manually closed?**

**A:** Following are the operations that do not run in background while the object is locked in another operation:

Operations	
Renaming table	Creating constraint
Setting schema on table	Creating index
Setting tablespace in table	Renaming schema

Operations	
Setting description in table	Adding column
Renaming partition	-

29. **Do we have a limit on the column and row size while exporting table data to excel?**  
**A:** Yes, xlsx format supports maximum of 1 million rows and 16384 columns and xls format supports maximum of 64K rows and 256 columns.

# 12 Glossary

The following table contains abbreviations, terminologies and their descriptions:

Term	Definition
Breakpoint	A location in a program at which execution is halted so that a programmer can examine the program's status, the contents of Variable, and so on.
Client	A computer or program that connects to or requests the services of another computer or program.
Column constraints	Column constraints are restrictions on the data that can be inserted into a given column.
Compile	Performs a PL/SQL compilation of the function.
Consistency	Transactions always operate on a consistent view of the data. Data is consistent as long as it conforms to a set of invariants, that is, no two rows in the customer table can have the same customer ID and all orders have an associated customer row. When one transaction gets inconsistent, other transactions cannot see these inconsistencies, and it be eliminated when the transaction ends.
CSV	<p>A comma-separated value (CSV) (also sometimes called character-separated value, because the separator character does not have to be a comma) file stores tabular data (numbers and text) in plain-text form. Plain text means that the file is a sequence of characters, with no data that has to be interpreted instead, as binary numbers.</p> <p>A CSV file consists of a number of records, separated by line breaks of some kind; each record consists of fields, separated by some other character or string, most commonly a literal comma or tab. Usually, all records have an identical sequence of fields.</p>
Database (DB)	<p>Database is a collection of related information, typically organized to make common retrievals easy and efficient.</p> <p>Properties of a database:</p> <ul style="list-style-type: none"><li>● Database name</li><li>● Endian file formats (BIG_ENDIAN or LITTLE_ENDIAN)</li><li>● Relations</li><li>● A database without relation cannot exist</li></ul>

Term	Definition
Database Administrator (DBA)	<p>A database administrator (short form DBA) is a person responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization.</p> <p>The role includes the development and design of database strategies, monitoring and improving database performance and capacity, and planning for future expansion requirements. They may also plan, co-ordinate and implement security measures to safeguard the database.</p>
DBMS	<b>Database Management System:</b> A database management system (DBMS) is a software package with computer programs that controls the creation, maintenance, and use of a database. It allows organizations to conveniently develop databases for various applications.
DDL	A data definition language or data description language is syntax that is similar to a computer programming language for defining data structures, especially database schemas.
Debugging	Debugging is a methodical process of finding and reducing the number of bugs, or defects, in a computer program or a piece of electronic hardware, thus making it behave as expected.
Debug Object	Debug Object is an abstraction of any database debug-able objects like functions/procedures.
Default	The pre-defined configuration of a system or an application. In most programs, the defaults can be changed to reflect personal preferences.
DML	<b>Data Manipulation Language:</b> A data manipulation language (DML) is a family of syntax elements similar to a computer programming language used for inserting, deleting and updating data in a database. Performing read-only queries of data is sometimes also considered a component of DML.
Drop-down menu	A menu that opens vertically on-screen to display context-related options. Also called pop-up menu or pull-down menu.
Execute	To perform an instruction.
Expression	An SQL statement that returns a value.
Field	<p>A field is a segment of database record for query and display. It is a part of a record used for a particular category of data.</p> <p>Properties of a field:</p> <ul style="list-style-type: none"><li>● Field name</li><li>● Field type</li><li>● Field size</li><li>● Wildcard Value: Default value (this is provided only for Field types, namely, UINT8, UINT16, UINT32, STRING, VSTRING, and IP_ADDRESS).</li></ul>

Term	Definition
GUI	<p><b>Graphical User Interface:</b> A working environment in which a computer user is presented with a screen on which there are pictures or Icons representing programs, actions or files, and either uses a mouse cursor (or similar pointing device) to select the appropriate icon or uses a keyboard with directional buttons or keys to move around the screen and select the appropriate icon. There are often drop-down menus available when the mouse is placed over certain parts of the screen.</p> <p>A GUI typically makes use of Object-Oriented or event-driven programming; instead of following a pre-determined sequence of actions, the application waits for an event such as a mouse-click over a particular icon, to determine what action is required and execute the appropriate piece of code; the application then goes back into the "wait" state until another event occurs, such as a mouse click over a different icon.</p>
Hadoop Distributed File System (HDFS)	Hadoop distributed file systems allow data access with a high throughput and are applicable to large-scale data set applications.
HTML	An application of the Standard Generalized Markup Language that uses tags to mark elements, such as text and graphics, in a document to indicate how Web browsers must display these elements to the user and must respond to user actions.
In-memory database	An in-memory database (IMDB; also main memory database system or MMDB) is a database management system that primarily relies on main memory for computer data storage. It is contrasted with database management systems which employ a disk storage mechanism. Main memory databases are faster than disk-optimized databases since the internal optimization algorithms are simpler and execute fewer CPU instructions. Accessing data in memory reduces the I/O reading activity when querying the data which provides faster and more predictable performance than disk. In applications where response time is critical, such as telecommunications network equipment and mobile ads networks, main memory databases are often used.
Key	A key is a field, or combination of fields, that uniquely identifies a record in a table.
Key Store	A key store is a file that contains your public and private keys.
Menu Bar	The horizontal strip across the top of an application's window. Each word on the strip has a context sensitive drop-down menu containing features and actions that are available for the application in use.
Null Value	A field that does not contain a data item is said to have a null value. In a numeric field, a null value is not the same as a value of zero; in a character field, a null value is not the same as a blank -- both the numeric zero and blank character are definite values. A null value indicates that the field's value is undefined -- its value is not known.
Object Browser	<p>The object browser gives you access to all information that is relevant to PL/SQL development:</p> <ul style="list-style-type: none"> <li>● Create, view, edit, rename and drop objects.</li> <li>● View properties of the database and table.</li> <li>● Query and edit the data of tables and views.</li> </ul>

Term	Definition
ORDBMS	<b>Object-relational Database Management System:</b> An object-relational database (ORD), or object-relational database management system (ORDBMS), is a database management system (DBMS) similar to a relational database, but with an object-oriented database model: objects, classes and inheritance are directly supported in database schemas and in the query language. In addition, just as with pure relational systems, it supports extension of the data model with custom data-types and methods.
PL/pgSQL	<b>Procedural Language/PostgreSQL:</b> A procedural programming language which is supported by PostgreSQL ORDBMS.
PL/SQL	PL/SQL stands for Procedural Language extension of SQL. PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL.
PL/SQL Functions	A function is a named PL/SQL block which is similar to a procedure. The major difference between a procedure and a function is, a function must always return a value, but a procedure may or may not return a value.
Port	A network portal through which two computing processes can communicate.
Postgres/PostgreSQL	PostgreSQL, often simply "Postgres", is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance. As an GaussDB 200, GaussDB 300, and DWS database, its primary function is to store data, securely and supporting best practices, and retrieve it later, as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).  It is free and open source software, released under the terms of the PostgreSQL License, a permissive free software license.
Primary Key	A primary key uniquely specifies a tuple within a table. In order for an attribute to be a primary key it must not repeat. While natural attributes (attributes used to describe the data being entered) are sometimes good primary keys, surrogate keys are often used instead.
Procedures	Procedures, also known as routines, subroutines, methods, or functions (not to be confused with mathematical functions, but similar to those used in functional programming), simply contain a series of computational steps to be carried out.
Procedural Language	Procedural programming can sometimes be used as a synonym for imperative programming (specifying the steps the program must take to reach the desired state), but can also refer (as in this article) to a programming paradigm, derived from structured programming, based upon the concept of the procedure call. Procedures, also known as routines, subroutines, methods, or functions (not to be confused with mathematical functions, but similar to those used in functional programming), simply contain a series of computational steps to be carried out. Any given procedure might be called at any point during a program's execution by other procedures or by itself.

Term	Definition
Query	A complete select statement that specifies 1) the columns and tables from which data is to be retrieved, 2) optionally, conditions that the data must satisfy, 3) optionally, computations that are to be performed on the retrieved column values, and 4) optionally, a desired ordering of the result set.
Relational Database	A relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Data in a single table represents a relation, from which the name of the database type comes. In typical solutions, tables may have additionally defined relationships with each other.
Relational Model	A database in which inter-table relationships are organized based on common data columns which define a one-to-many relationship between a row of the primary key table and one or more rows of the matching foreign key table. Besides describing how the database tables are related, the relational model also defines how the related data can be accessed and manipulated. SQL is the most commonly used relational model database language.
Row	One set of related values for all of the columns declared in a given table. Also known as a record occurrence.
Schema	A schema is a collection of logical structures of data or schema objects.
SSL	<b>Secure Sockets Layer:</b> A security protocol that works at a socket level. This layer exists between the TCP layer and the application layer to encrypt/decode data and authenticate concerned entities.
Step Into	Step Into executes a single program statement at a time. If the execution point is located on a call to a subprogram, Step Into steps into that subprogram and places the execution point on its first statement. If the execution point is located on the last statement of a subprogram, Step Into returns from the subprogram, placing the execution point on the line of code that follows the call to the subprogram from which you are returning.
Step Out	Step Out leaves the current subprogram and goes to the next statement.
Step Over	Step Over bypasses the next subprogram (unless the subprogram has a breakpoint) and goes to the next statement after the subprogram. If the execution point is located on a subprogram call, it runs that subprogram without stopping (instead of stepping into it), then positions the execution point on the statement that follows the call. If the execution point is located on the last statement of a subprogram, Step Over returns from the subprogram, placing the execution point on the line of code that follows the call to the subprogram from which you are returning.
Stored Procedure	A stored procedure or in simple a proc is a named PL/SQL block which performs one or more specific task. This is similar to a procedure in other programming languages. A procedure has a header and a body. The header consists of the name of the procedure and the parameters or variable passed to the procedure. The body consists or declaration section, execution section and exception section similar to a general PL/SQL Block. A procedure is similar to an anonymous PL/SQL block but it is named for repeated usage.

Term	Definition
SQL	SQL referred to as Structured Query Language, is a special-purpose programming language designed for managing data in relational database management systems (RDBMS).
Table	A collection of closely related columns. A table consists of rows each of which shares the same columns but vary in the column values.
Tablespace	A tablespace is used to optimize the performance of the database.
Trust Store	A trust store is a key database file that contains the public keys for your partners' self-signed and CA certificates. The public key is stored as a signer certificate. For commercial CA, the CA root certificate is added. Because the trust store file does not contain your private key, the trust store file can be more publicly accessible than the key store file.
URI	<b>Uniform Resource Identifier:</b> A URI is the unique name used to access the resource. It is not necessarily a specific file location (For example, it may be a call to an application or a database), which is why it is preferred over the similar acronym URL (Uniform Resource Locator).
URL	<b>Uniform Resource Locator:</b> It is the global address of documents and other resources in the world wide web.
View	Views restrict access to specific rows or columns of a table. A view can be created from one or more tables and is determined by the query used to create the view.