

**Data Studio**  
**3.1.0**  
**User Manual**

**Issue** 01  
**Date** 2022-09-30



# 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 .....	3
2.5 Third Party Licenses .....	5
2.6 Reference Documents .....	5
<b>3 About Data Studio .....</b>	<b>7</b>
3.1 Overview .....	7
3.2 Supported Functions .....	8
3.3 Constraints and Limitations .....	11
3.4 Structure of Release Package .....	14
3.5 System Requirements .....	17
<b>4 Installing Data Studio .....</b>	<b>19</b>
4.1 Installing and Configuring Data Studio .....	19
4.2 Configuring a Cluster Database .....	26
4.2.1 Configuration Description .....	26
4.2.2 Configuring a Whitelist for openGauss .....	26
4.2.3 Accessing the Data Studio .....	27
<b>5 Getting Started .....</b>	<b>28</b>
5.1 Starting Data Studio .....	28
5.2 Supporting Command Line Supply of Connection Parameters .....	29
5.3 Data Studio User Interface .....	31
5.4 Data Studio Menus .....	32
5.4.1 File .....	32
5.4.2 Edit .....	33
5.4.3 Run .....	37
5.4.4 Settings .....	37
5.4.5 Help .....	38
5.5 Data Studio Toolbars .....	39

5.6 Data Studio Right-Click Menus .....	39
<b>6 Using Data Studio .....</b>	<b>45</b>
6.1 Overview .....	45
6.2 Connection Profiles .....	45
6.2.1 Overview .....	45
6.2.2 Adding a Connection .....	46
6.2.3 Renaming a Connection .....	52
6.2.4 Editing a Connection .....	53
6.2.5 Removing a Connection .....	53
6.2.6 Viewing Connection Properties .....	54
6.2.7 Refreshing a Database Connection .....	54
6.3 Databases .....	58
6.3.1 Creating a Database .....	58
6.3.2 Disconnecting All Databases .....	59
6.3.3 Connecting to Database .....	59
6.3.4 Disconnecting Database .....	59
6.3.5 Renaming a Database .....	60
6.3.6 Dropping a Database .....	60
6.3.7 Viewing a Database Properties .....	61
6.4 Schemas .....	61
6.4.1 Overview .....	61
6.4.2 Creating a Schema .....	61
6.4.3 Exporting Schema DDL .....	62
6.4.4 Exporting Schema DDL and Data .....	64
6.4.5 Renaming a Schema .....	65
6.4.6 Support Sequence DDL .....	65
6.4.7 Grant/Revoke Privilege .....	66
6.4.8 Dropping a Schema .....	66
6.4.9 Synonym Management .....	67
6.5 Functions/Procedures .....	70
6.5.1 Creating Function/Procedure .....	70
6.5.2 Editing a Function/Procedure .....	72
6.5.3 Grant/Revoke Privilege .....	73
6.5.4 Working with Functions/Procedures .....	74
6.5.4.1 Overview .....	74
6.5.4.2 Selecting a DB Object in the PL/SQL Viewer .....	74
6.5.4.3 Exporting a Function/Procedure DDL .....	75
6.5.4.4 Viewing Object Properties in the PL/SQL Viewer .....	76
6.5.4.5 Dropping a Function/Procedure .....	77
6.5.4.6 Executing a Function/Procedure .....	78
6.5.4.7 Grant/Revoke Privilege .....	80

6.5.5 Debugging a PL/SQL Function .....	80
6.5.5.1 Overview .....	80
6.5.5.2 Using Breakpoints .....	81
6.5.5.3 Controlling Execution .....	86
6.5.5.4 Checking Debug Information .....	88
6.5.6 Supporting Code Folding/UnFolding .....	90
6.6 Tables .....	94
6.6.1 Overview .....	94
6.6.2 Creating Regular Table .....	95
6.6.2.1 Overview .....	95
6.6.2.2 Working with Columns .....	102
6.6.2.3 Working with Constraints .....	104
6.6.2.4 Working with Indexes .....	105
6.6.3 Creating Foreign Table .....	107
6.6.4 Creating Partition Table .....	107
6.6.4.1 Overview .....	107
6.6.4.2 Working with Partitions .....	111
6.6.5 Grant/Revoke Privilege - Regular/Partition Table .....	112
6.6.6 Managing Table .....	113
6.6.6.1 Overview .....	113
6.6.6.2 Renaming a Table .....	114
6.6.6.3 Truncating a Table .....	114
6.6.6.4 Reindexing a Table .....	114
6.6.6.5 Analyzing a Table .....	115
6.6.6.6 Vacuuming a Table .....	115
6.6.6.7 Setting the Table Description .....	115
6.6.6.8 Setting the Tablespace .....	116
6.6.6.9 Setting the Schema .....	116
6.6.6.10 Dropping a Table .....	116
6.6.6.11 Viewing Table Properties .....	117
6.6.6.12 Grant/Revoke Privilege .....	118
6.6.6.13 Show Related Sequences .....	118
6.6.7 Managing Table Data .....	119
6.6.7.1 Overview .....	119
6.6.7.2 Exporting Table DDL .....	119
6.6.7.3 Exporting Table DDL and Data .....	120
6.6.7.4 Exporting Table Data .....	121
6.6.7.5 Showing DDL .....	123
6.6.7.6 Importing Table Data .....	123
6.6.7.7 Viewing Table Data .....	125
6.6.7.8 Editing Table Data .....	127
6.6.8 Editing Temporary Tables .....	131

6.6.9 Supporting ER for openGauss .....	132
6.7 Sequences .....	134
6.7.1 Creating Sequence .....	134
6.7.2 Grant/Revoke Privilege .....	135
6.7.3 Working with Sequences .....	135
6.8 Views (Including Materialized Views) .....	136
6.8.1 Creating a View .....	136
6.8.2 Grant/Revoke Privilege .....	137
6.8.3 Working with Views .....	137
6.9 Tablespaces .....	140
6.9.1 Creating a Tablespace .....	140
6.9.2 Working with Tablespaces .....	142
6.10 Users/Roles .....	143
6.10.1 Create User/Role .....	143
6.10.1.1 openGauss .....	144
6.10.2 Working with User/Role .....	145
6.10.3 Revoke User/Role .....	146
6.11 SQL Terminal .....	146
6.11.1 Opening SQL Terminals .....	146
6.11.2 Managing SQL Query Execution History .....	151
6.11.3 Opening And Saving SQL Scripts .....	154
6.11.4 Viewing Object Properties in the SQL Terminal .....	155
6.11.5 Canceling Execution of SQL Queries .....	157
6.11.6 Formatting of SQL Queries .....	157
6.11.7 Selecting a DB Object in the SQL Terminal .....	158
6.11.8 Viewing the Query Execution Plan and Cost .....	160
6.11.9 Viewing the Query Execution Plan and Cost Graphically .....	163
6.11.10 Working with the SQL Terminals .....	164
6.11.11 Exporting Query Results .....	178
6.11.12 Managing SQL Terminal Connections .....	179
6.12 Batch Operation .....	179
6.12.1 Overview .....	179
6.12.2 Dropping Batch of Objects .....	180
6.12.3 Exporting DDL Batch of Objects .....	182
6.12.4 Exporting DDL and Data Batch of Objects .....	183
6.12.5 Granting/Revoking Privileges .....	183
<b>7 Personalizing Data Studio .....</b>	<b>184</b>
7.1 Overview .....	184
7.2 General .....	184
7.3 Editor .....	188
7.4 Security .....	194

7.5 Environment .....	196
7.6 Result Management .....	200
7.7 Export/Import .....	205
<b>8 References .....</b>	<b>208</b>
8.1 Performance Specification .....	208
<b>9 Troubleshooting .....</b>	<b>210</b>
<b>10 Security Management .....</b>	<b>216</b>
10.1 Overview .....	216
10.2 Login History .....	216
10.3 Password Expiry Notification .....	217
10.4 Securing the Application In-Memory Data .....	217
10.5 Data Encryption for Saved Data .....	217
10.6 SQL History .....	217
10.7 SSL Certificates .....	218
<b>11 FAQs .....</b>	<b>228</b>
<b>12 Glossary .....</b>	<b>233</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
- executing SQL statements or SQL scripts
- editing and executing PL/SQL statements
- viewing graphically the query execution plan and cost
- exporting table data operations

The creating and managing database objects include

- database
- schema
- functions
- procedures
- tables
- sequences
- indexes
- views
- tablespaces
- synonym

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

Date	Version	Change Description

2022-09-30	3.0.0	<ul style="list-style-type: none"><li>• Add the option of saving passwords by default</li><li>• Add third_Party_Open_Source_Software_List file</li><li>• Add open source license file</li><li>• Operation process optimization of compilation guidance in Readme</li><li>• Modify some known bugs</li></ul>
2022-03-30	3.0.0	<ul style="list-style-type: none"><li>• Update JDK version 11</li><li>• Add javafx-sdk 18 / openjfx.p2 18</li><li>• Modify some bugs</li></ul>
2021-09-30	2.1.0	<ul style="list-style-type: none"><li>• Revoke user/roles: newly added. 6.10.3</li><li>• The list/hash/interval partition is added to the partition table. 6.6.4</li><li>• Viewing partition data. 6.6.4</li><li>• Unique primary keys and unique indexes for column-stored tables: newly added. 6.6.2.1</li></ul>
2021-03-30	2.0.0	<ul style="list-style-type: none"><li>• Debugging a PL/SQL Function: newly added. 6.5.5</li></ul>
2020-09-30	1.0.1	<ul style="list-style-type: none"><li>• Foreign key: newly added.</li><li>• Views (Including Materialized Views): name updated.</li></ul>
2020-06-30	1.0.0	This issue is the first official release.

## 2.4 Document Conventions

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

### Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Symbol	Description
	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.
	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

## 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 **GaussDB Tools 1.0.1\_Open Source Software Notice.doc** for license information and details.

**Table 2-1** List of Third Party Software

Third Party Software
Apache POI 4.1.1
Apache Log4j 2.13.0
ANTLR, ANother Tool for Language Recognition 4.7.2
Apache Jakarta Commons IO 2.6
Apache Commons Collections 4.4
Apache Jakarta Commons Math 3.6.1
google-guava 28.0
Google-guice 4.2
gson 2.8.6
JSqlParser 1.2
Eclipse Nebula NatTable 1.6.0
Eclipse Tools Graphical Editing Framework (GEF) 5.1.0
Eclipse for RCP and RAP Developers 4.12
Eclipse efxclipse 3.5.0
jQuery JavaScript Library 3.4.1
Apache XMLBeans 3.0.2
Apache Commons Compress 1.19
Apache Jakarta Commons Math 3.6.1

## 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
GaussDB Tools 1.0.1_Open Source Software Notice.doc	Written offer listing the open source software used and their license information
openGauss 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

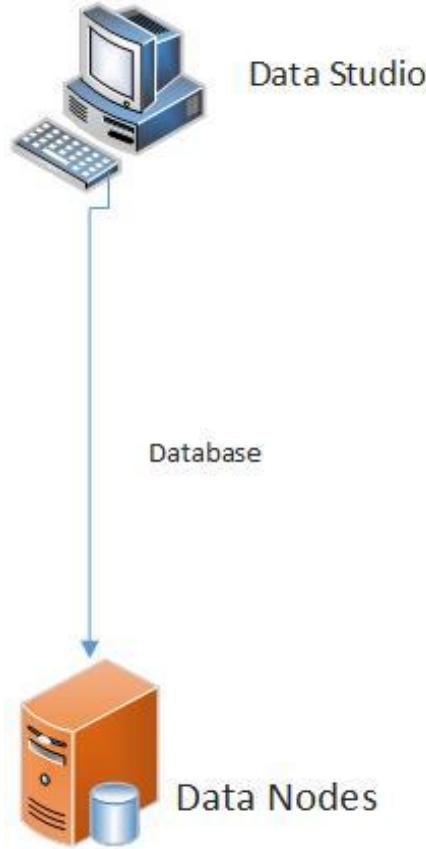
- Manage and Create database objects
- Executing SQL statements or SQL scripts
- Editing and executing PL/SQL statements
- Importing and Exporting table data

Creating database objects include

- database
- schema
- functions
- procedures
- tables
- sequences
- columns
- indexes
- constraints
- views
- tablespaces
- synonym

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

**Figure 3-1** Operational Context of Database and Data Studio



## 3.2 Supported Functions

Data Studio supports Online Analytical Processing database refer to openGauss products.

Following table describes the functions/operations of Data Studio that are supported by openGauss:

### 6.2 Connection Profiles

- 6.2.2 Adding a Connection
- 6.2.3 Renaming a Connection
- 6.2.4 Editing a Connection
- 6.2.5 Removing a Connection
- 6.2.6 Viewing Connection Properties

### 6.3 Databases

- 6.3.1 Creating a Database
- 6.3.2 Disconnecting All Databases

- 6.3.3 Connecting to Database
- 6.3.4 Disconnecting Database
- 6.3.5 Renaming a Database
- 6.3.6 Dropping a Database
- 6.3.7 Viewing a Database Properties

#### 6.4 Schemas

- 6.4.2 Creating a Schema
- 6.4.3 Exporting Schema DDL
- 6.4.4 Exporting Schema DDL and Data
- 6.4.5 Renaming a Schema
- 6.4.8 Dropping a Schema
- 6.4.9 Synonym Management

#### 6.5 Functions/Procedures

- 6.5.1 Creating Function/Procedure
- 6.5.4.5 Dropping a Function/Procedure
- 6.5.4.3 Exporting a Function/Procedure DDL
- 6.5.2 Editing a Function/Procedure

#### 6.6 Tables

- 6.6.2 Creating Regular Table
- 6.6.3 Creating Foreign Table
- 6.6.4 Creating Partition Table

#### 6.6.6 Managing Table and 6.6.7 Managing Table Data

- 6.6.6.2 Renaming a Table
- 6.6.6.3 Truncating a Table
- 6.6.6.4 Reindexing a Table
- 6.6.6.5 Analyzing a Table
- 6.6.6.6 Vacuuming a Table
- 6.6.6.7 Setting the Table Description
- 6.6.6.8 Setting the Tablespace
- 6.6.6.9 Setting the Schema
- 6.6.6.10 Dropping a Table
- 6.6.6.11 Viewing Table Properties
- 6.6.7.2 Exporting Table DDL
- 6.6.7.3 Exporting Table DDL and Data
- 6.6.7.4 Exporting Table Data
- 6.6.7.5 Showing DDL
- 6.6.7.6 Importing Table Data
- 6.6.7.7 Viewing Table Data
- 6.6.7.8 Editing Table Data
- 6.6.9 Supporting ER for openGauss

#### 6.6.2.2 Working with Columns

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

#### 6.6.2.3 Working with Constraints

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

#### 6.6.2.4 Working with Indexes

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

#### 6.6.4.2 Working with Partitions

- [Rename a Partition](#)
- [Drop a Partition](#)

### 6.7 Sequences

- [6.7.1 Creating Sequence](#)
- [Dropping a Sequence](#)
- [Dropping a Sequence Cascade](#)

### 6.8 Views (Including Materialized Views)

- [6.8.1 Creating a View](#)
- [6.8.2 Grant/Revoke Privilege](#)
- [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](#)

### 6.9 Tablespaces

- [6.9.1 Creating a Tablespace](#)

### 6.10 Users/Roles

- [6.10.1 Create User/Role](#)
- [Viewing/Editing User/Role Properties](#)
- [Viewing the User/Role DDL](#)

#### [Search Objects](#)

##### 6.11 SQL Terminal

- [Auto Commit](#)
- [6.11.7 Selecting a DB Object in the SQL Terminal](#)
- [Error Locator](#)
- [6.11.10 Working with the SQL Terminals](#)
- [SQL Assistant](#)
- [6.11.6 Formatting of SQL Queries](#)
- [10.6 SQL History](#)
- [6.11.9 Viewing the Query Execution Plan and Cost Graphically](#)
- [Execute SQL Queries](#)
- [Multi-Column Sort](#)
- [Backup Unsaved Queries/Functions/Procedures](#)
- [Search in PL/SQL Viewer or SQL Terminal](#)
- [Go to Line in PL/SQL Viewer or SQL Terminal](#)
- [Comment/Uncomment/Indent/Un-indent Lines/Insert Space](#)
- [Execute Multiple Functions/Procedures or Queries](#)
- [Rename SQL Terminal](#)
- [Using Templates](#)

##### 6.11.10 Working with the SQL Terminals

- [Export Results Data](#)
- [Search Results](#)

##### 6.12 Batch Operation

## 3.3 Constraints and Limitations

The following are known limitations in Data Studio:

### Object Browser Filter Tree

The filter count and filtering status of the tree are not supported. Filtering stops after timeout and display result for filtering done in time period. You can set **Timeout** period in **Preference-->General-->Object Browser-->Filter Timeout**. For information, refer [Setting Filter Timeout](#).

### 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 [7.5 Environment](#).

## 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 6.6.7.8 Editing 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 64 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 up to 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.

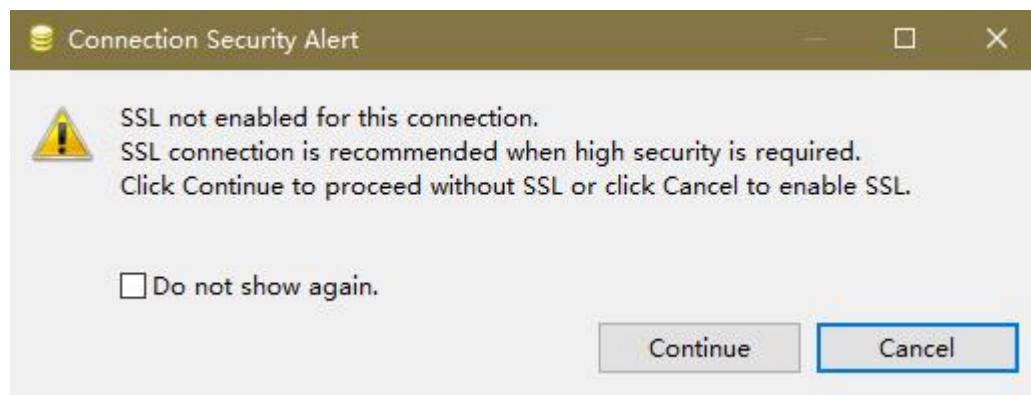
## Security

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 config folder under the corresponding profile folder and restart DS.

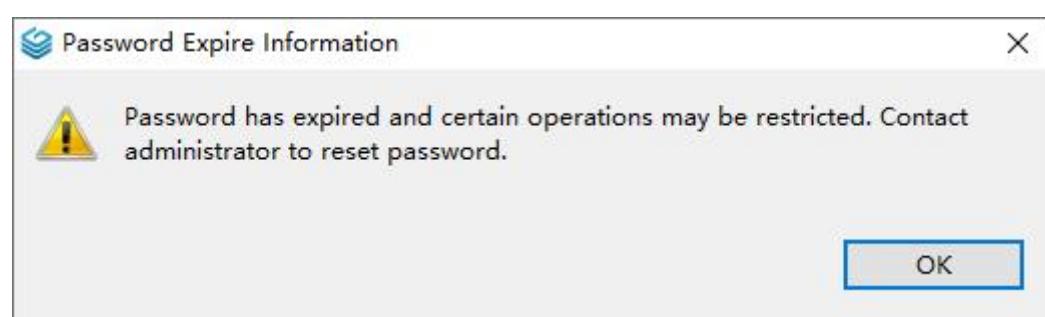
If the authentication mechanism is disabled and Enable SSL is not selected, a risk message is displayed.



Periodically update passwords and notify users of risks.

### NOTE

If the password is not updated in a timely manner, the Data Studio displays a message.



Complexity check: If complexity check is disabled, a message is displayed indicating that the risk is not involved. The Data Studio does not check the password complexity. The database verifies the password complexity.

```
postgres=> create user test123 with password '123456';
ERROR: Password must contain at least 8 characters.
postgres=> create user test123 with password '12345678';
ERROR: Password must contain at least three kinds of characters.
postgres=> create user test123 with password 'ABCD1234';
ERROR: Password must contain at least three kinds of characters.
postgres=> create user test123 with password 'Test@123';
CREATE ROLE
```

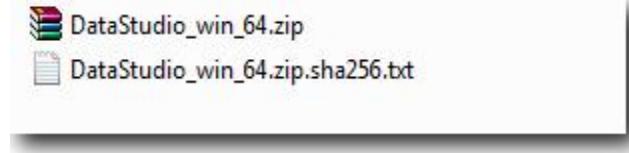
## SQL Terminal

- 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.
- DS supports basic formatting of simple SELECT statements only and may not work as expected for complex queries.

## 3.4 Structure of Release Package

### Preparing Software Packages

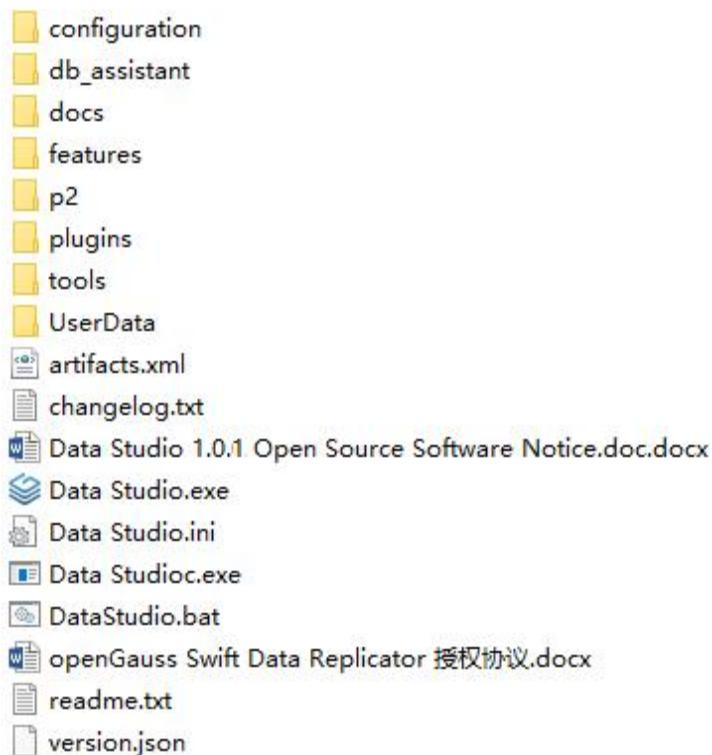
Download the Data Studio Software package based on the OS in use and the corresponding verification files to the directory.



**Table 3-1** Data Studio Software Packages

Software Package	Description
DataStudio_win_64.zip	Data Studio software package for windows operating system.
DataStudio_win_64.zip.sha256.txt	Reliability verification file for the DataStudio software package.

The release package structure of Data Studio is as follows:



**Table 3-2** Structure Description

Folders/Files	Description
configuration	Contains information about the application launcher and the required <b>Eclipse</b> plug-in path.
db_assistant	Contains SQL assistant related files.
docs	Contains <i>Data Studio User Manual.pdf</i> which provides you with details on using Data Studio.
features	Contains Eclipse (rich client protocol-GUI) and Data Studio features.
p2	Contains files required for provisioning and managing <b>Eclipse</b> and <b>Equinox</b> -based applications.
plugins	Contains the required <b>Eclipse</b> and Data Studio plugins.
tools	Contains Data Studio dependent tools.
UserData<USERNAME>/ • Autosave • Logs/ • Preferences/ • Profile<PROFILE1, PROFILE2,...>/	Contains separate folders for each OS user who uses Data Studio. <b>Autosave</b> - Contains the auto saved information of unsaved queries and functions/procedures. <b>Logs</b> - Contains Data Studio.log which

Folders/Files	Description
- History/ • config/	<p>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 <i>connection.properties</i>, SQL History and <i>Profiles.txt</i> files required to manage connection profiles in Data Studio.</p> <p><b>Config</b> - Contains Data Studio's internal folder.</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>• <b>Logs</b> folder, language, memory settings and log level are common for all users.</li><li>• The <b>Logs</b> folder, <b>Data Studio.log</b> file, <b>Preferences</b> folder, <b>Preferences.prefs</b> file, <b>Profile</b> folder, <b>connection.properties</b> file, <b>Profiles.txt</b> file, and config folder are created after launching Data Studio.</li><li>• If <b>Logs</b> folder path is provided in <b>Data Studio.ini</b> file, then logs are created in the specified path.</li><li>• When user is not able to log in to the <b>Data Studio</b> due to security keys are corrupted. Follow the steps to generate new security keys:<ol style="list-style-type: none"><li>1. Delete the config folder from <b>Data Studio\UserData\&lt;UserId&gt;\Profile\&lt;Profile ID&gt;\config</b>.</li><li>2. Restart Data Studio.</li></ol></li></ul>
artifacts.xml	Contains the product build information.
changelog.txt	Contains the detailed change log information of release version.
Data Studio 1.0.1 Open Source Software Notice.doc	Written offer listing the open source software used and their license information.
Data Studio.exe	Allows you to connect to the database and perform various operations (such as managing database objects, editing or executing PL/SQL programs and so on).
DataStudio.bat	Allows you to connect to the database and perform various operations in Windows.
Data Studio.ini	Contains run-time configuration information of Data Studio.
Data Studioc.exe	Allows to launch Data Studio in command line.

Folders/Files	Description
openGauss Data Studio 授权协议.docx	This License Agreement applies to the OpenGauss Data Studio (the Software) developed by the OpenGauss Community.
readme.txt	Contains the features or fixed issues of current release version.
version.json	Contains metadata information file.

## 3.5 System Requirements

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

### Hardware Requirements

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

**Table 3-3** List of Hardware Requirements for Data Studio

Hardware Requirement	Configuration
CPU	x86 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 100 MB free space in user's home directory.
Network Requirements	Gigabit Ethernet

### Software Requirements

#### Operating System Requirements

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

**Table 3-4** Supported OS and Corresponding Installation Packages

Server Type	Operating System	Supported Version
Universal x86 servers	Microsoft Windows	Windows 7 (64 bit)
		Windows 10 (64 bit)
		Windows 2012 (64 bit)
		Windows 2016 (64 bit)

### Browser Requirements

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

**Table 3-5** Browser Requirements

Operating System	Version
Microsoft Windows	IE 11 and above

### Other Software Requirements

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

**Table 3-6** List of Software Requirements for Data Studio

Software	Specification
Java	Recommended Open JDK version is 11 along with JavaFx or above with appropriate bit number.

**Table 3-7** Supported Database Versions

Database	Version Numbers
openGauss	1.0.1
openGauss	2.0.0
openGauss	2.1.0
openGauss	3.0.0

#### NOTE

Minimum screen resolution recommended for best experience is **1080 x 768**. UI abnormalities may occur if screen resolution is less than the earlier mentioned value.

# 4 Installing Data Studio

## 4.1 Installing and Configuring Data Studio

This section describes the installation and configuration steps to be followed to use Data Studio.

This section contains the following topics:

[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](#)

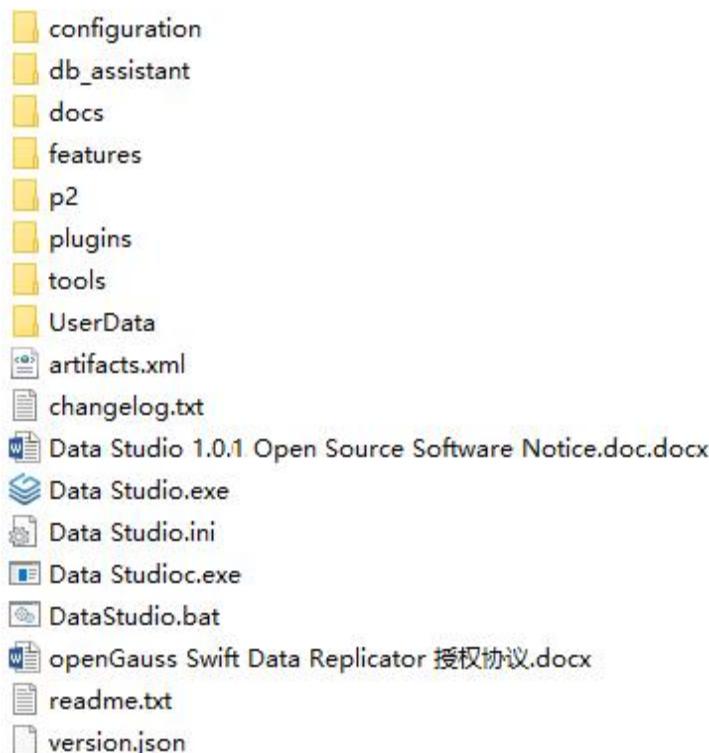
## Installing Data Studio

Data Studio can be run after extraction of package.

Follow the steps to install Data Studio:

- Step 1** Extract the required package. If the user prefer to install in other folder, then admin should control the folder access permissions to users.

You can 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 following 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	plugins/org.eclipse.equinox.launcher_1.5.400.v20190515-0925.jar
--launcher.library	Defines the libraries required to load Data	N/A	plugins/org.eclipse.equinox.launcher.win

Parameter	Description	Value Range	Default Value
	Studio. This information varies based on the version used.		32.win32.x86_64_1.1.1000.v20190125-016
-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	5000
-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;username&gt;\logs</i> . Refer to <a href="#">Providing Location to Create Log File</a> 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
-enableSSL	This parameter is used to enable SSL.	True, False	True
-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
-focusOnFirstResult	Defines auto focus	True/False	False

Parameter	Description	Value Range	Default Value
	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.		
<b>NOTE</b> <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
-Dosgi.requiredJavaVersion	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_181.
-Xms	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: <ul style="list-style-type: none"><li>• -Xms40m</li><li>• -Xms120m</li></ul> Refer to Java documentation for more information.	N/A	-Xms40m
-Xmx	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	N/A	-Xmx1200m

Parameter	Description	Value Range	Default Value
	<p>indicate megabytes, g or G to indicate gigabytes.</p> <p>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>		
-Dorg.osgi.framework.bundle.parent=ext	This parameter specifies which class loader is used for boot delegation.	boot, app and ext	boot
-Dosgi.framework.extensions=org.eclipse.fx.osgi	This parameter is used to specify a list of framework extension names. Framework extension bundles are fragments of the system bundle (org.eclipse.osgi). As a fragment, user can provide extra classes with the framework to use.	N/A	N/A

#### NOTE

- The user should not change the **Dorg.osgi.framework.bundle.parent=ext** and **Dosgi.framework.extensions=org.eclipse.fx.osgi** parameters.
- If user sees the following message - **SocketException : Bad Address: Connect**

Then user should check if the client connection to the server is being established through IPv6 or IPv4 protocol. Based on the user preference, the connection can be established by providing the following statements in .ini file:

**-Djava.net.preferIPv4Stack=true**  
**-Djava.net.preferIPv6Stack=false**

Following Table 4-2 are supported:

The **top row** and **left column** represent various node types attempting to communicate. An **x** indicates that these nodes can communicate with each other.

**Table 4-2** Communication Scenario

(Nodes)	V4 Only	V4/V6	V6 Only
<b>V4 Only</b>	x	x	No Communication Possible
<b>V4/V6</b>	x	x	x
<b>V6 Only</b>	No Communication Possible	x	x

## Providing Location to Create Log File

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

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

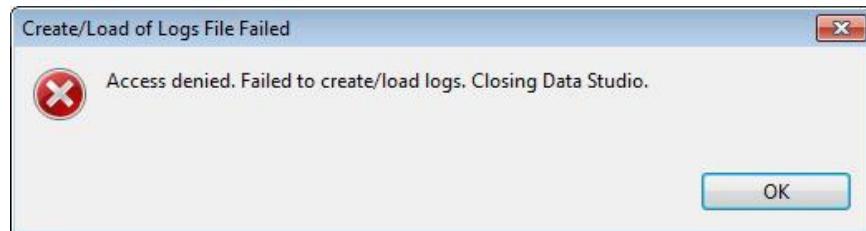
For example:

```
-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 users does not have access to the path mentioned in the **Data Studio.ini** file, then Data Studio closes with the following 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:

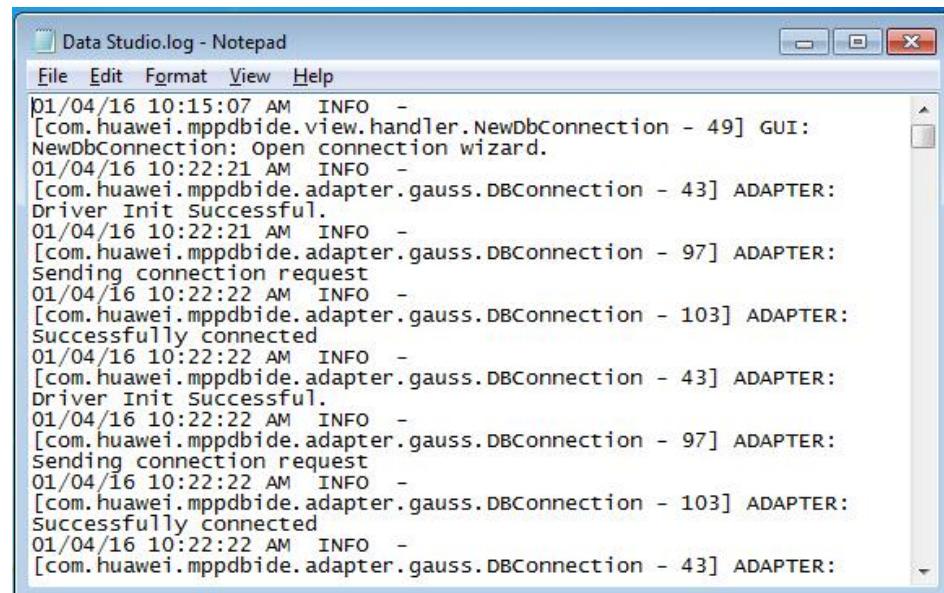
```
-logfolder=.
```

- The path provided does not exist.

### NOTE

Refer to server manual for more 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:

```
06/29/16 02:39:01 INFO - [com.huawei.mppdbide.view.handler.NewDbConnection - 49] GUI: NewDbConnection: open connection wizard.
```

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**. 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. The Data Studio deletes the old log file that is **Data Studio.log.1**. For example, the **Data Studio.log.5** renames to **Data Studio.log.4**, the **Data Studio.log.4** renames to **Data Studio.log.3** and so on.

## 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 .
- **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 4-3** Logging Level

-	FATAL	ERROR	WARN	INFO	TRACE
OFF	X	X	X	X	X

-	FATAL	ERROR	WARN	INFO	TRACE
FATAL	✓	✗	✗	✗	✗
ERROR	✓	✓	✗	✗	✗
WARN	✓	✓	✓	✗	✗
INFO	✓	✓	✓	✓	✗
TRACE	✓	✓	✓	✓	✓
ALL	✓	✓	✓	✓	✓

✓ - Log file created    ✗ - Log file not created

## 4.2 Configuring a Cluster Database

### 4.2.1 Configuration Description

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 openGauss.

- For details about how to configure a whitelist for openGauss, see 4.2.2 Configuring a Whitelist for openGauss.

### 4.2.2 Configuring a Whitelist for openGauss

- Configure the **pg\_hba.conf** file to allow users to connect to a database through a local Windows host.
  - Log in to any host in a 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.
  - 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 -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:
  - 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.

### 4.2.3 Accessing the Data Studio

OpenGauss DB User requires the following grant permission to access the Data Studio:

```
GRANT SELECT ON pg_catalog.pg_roles to <user>;
GRANT SELECT ON pg_catalog.pg_user_status to <user>;
GRANT ALL PRIVILEGES on TABLESPACE pg_default,pg_global TO <user>;
```



#### NOTE

If the DB user is not Admin User then Tablespace - Show DDL will be disabled.

# 5 Getting Started

- 5.1 Starting Data Studio
- 5.2 Supporting Command Line Supply of Connection Parameters
- 5.3 Data Studio User Interface
- 5.4 Data Studio Menus
- 5.5 Data Studio Toolbars
- 5.6 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 3.4 Structure of 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 (64bit)	OS (bit)	Java (bit)	Outcome
64	32	32	<a href="#">error message</a> is displayed
64	64	32	<a href="#">error message</a> is displayed
64	64	64	Launches Data Studio

----End

## 5.2 Supporting Command Line Supply of Connection Parameters

Connection related parameters can be supplied to Data Studio executable to connect to database server. Connection dialog is not launched again when parameters are supplied through command line.

Information about parameter name and what value they take can be found in the following Table 5-1:

### Usage:

```
"DataStudio.bat" connectionName=MY_CONNECTION host=XX.XX.XX.XX hostPort=DDDD  
dbName=DB_NAME userName=USER_NAME [savePassword=[current_session|do_not_save]]
```

### Usage with ssl parameters:

```
"DataStudio.bat" connectionName=MY_CONNECTION host=XX.XX.XX.XX hostPort=DDDD  
dbName=DB_NAME userName=USER_NAME [savePassword=[current_session|do_not_save]]  
[sslEnable=[true|false]] [sslClientCert=CLIENT_CERT] [sslClientKey=CLIENT_KEY]  
[sslRootCert=ROOT_CERT] [sslMode=[allow|require|verify_ca|verify_full]]
```

#### NOTE

Once earlier mentioned commands are executed, then the db password will be prompted in console.

For example

```
"DataStudio.bat" connectionName=my_connection host=10.XX.XX.XX hostPort=2554  
dbName=postgres userName=dsuser savePassword=current_session
```

**Table 5-1** Parameter Details

Sr. No	Parameter Name	Default Value	Range of Values	Mandatory / Optiona l	Validation	Comments
1.	connectionName	None	None	Mandatory	Validation rules same to connection dialog.	-
2.	host	None	None	Mandatory	Validation rules same to connection dialog.	-
3.	hostPort	None	None	Mandatory	Validation rules same to connection dialog.	-
4.	dbName	None	None	Mand	Validation	-

Sr. No	Parameter Name	Default Value	Range of Values	Mandatory / Optiona l	Validation	Comments
				atory	rules same to connection dialog.	
5.	userName	None	None	Mandatory	None	-
6.	sslEnable	false	1. True 2. false	Optional	Check if value is either true/false.	1. If value is something other than true/false, a warning message is printed and default value is considered to proceed with execution. 2. Value for this parameter is case in-sensitive that is, "TrUe" is also considered as "true".
7.	sslClient Cert	None	None	Optional	Valid path and file exists or not.	-
8.	sslClient Key	None	None	Optional	Valid path and file exists or not.	-
9.	sslRootCert	None	None	Optional	Valid path and file exists or not.	-
10.	sslMode	allow	1) require 2) verify_ca 3) verify_full 4) allow	Optional	Direct value check as only 4 possibilities.	-
11.	savePassword	current_session	1. current_session 2. do_not_save	Optional	Direct value check as only 2 values are possible	-

 NOTE

1. Save password permanently option does not support through command line as preferences will not be loaded while processing the arguments.
2. Command line arguments can only be entered in English in command prompt.
3. In Windows, if any error occurs while validating parameters, then error message is displayed in console.
4. Data studio workbench closes if any validation fails.
5. This feature is supported for one connection only.
6. When pressed **Ctrl+C** during while DS is running, **suppress terminate batch job (y/n)** is prompted in console in windows. No matter, what input is given at this point (Y or N), DS will exit. This is OS behavior as DataStudio is launched through bat script.
7. When DS launched through command line arguments is restarted, you need to press **ENTER** before typing password.

## Constraints

Command line arguments have following constraints:

- Arguments need to be provided in option = value format
- There should not be any space on either side of =
- Two different arguments need to be separated with one or more spaces
- If a value contains space, value needs to be enclosed in double quotes.  
For example, connectionName ="my connection"

## 5.3 Data Studio User Interface

This section describes the user interface of Data Studio.

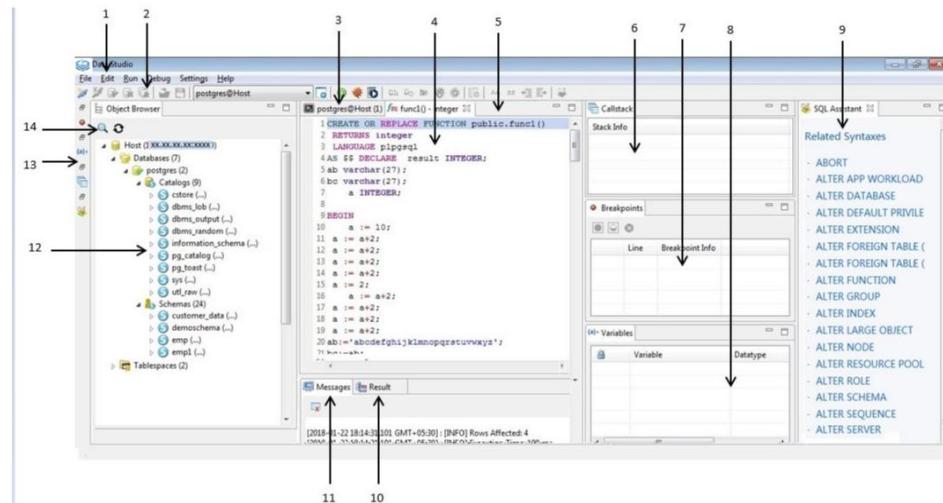
The Data Studio user interface contains the following:

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. **Variables** pane shows variables and their values.
8. **SQL Assistant** tab displays suggestion or reference for the information entered in the SQL Terminal and PL/SQL Viewer.
9. **Result** tab displays the result(s) of an executed function/procedure, or an SQL statement.
10. **Messages** tab displays the output of a process, such as standard input, standard output, and standard error(s).
11. **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.
12. **Minimized Window Panel** is used to open Callstack Variables pane. This panel is displayed only when CallstackVariables pane or all three are minimized.

13. **Search Toolbar** is used to search objects from the Object browser.
14. **Visual Explain** displays a graphical representation of the sql query using information from the extended JSON format.

**NOTE**

Item 13 and 14 are not visible until specific functionality is triggered. The following figure uses openGauss as an example.



## 5.4 Data Studio Menus

### 5.4.1 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> .

Function	Button	Shortcut Key	Description
			<b>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
Import Connections		-	Select <b>Import Connections</b> to import the connection profiles to the connection Wizard.
Export Connections		-	Select <b>Export Connections</b> to save the connection profiles to the disk.
Exit	-	Alt+F4	Exits from Data Studio and closes the connection. Force Exit: Exit without saving unsaved SQL history. Graceful Exit: Exit after saving unsaved SQL History and queries/functions/procedures. <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.



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

## 5.4.2 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
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.
Comment/Uncomment Lines	-	Ctrl+/	Comments/Uncomments each selected line
Comment/Uncomment Block	-	Ctrl+Shift+/	Comments/Uncomments all selected lines or a selected block

## 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

Object Type	Copied Format
Columns	object name
Constraints	object name
Partition names	object name
All other object types	schema.object name
Sequence	schema.object name
Synonym	object name

## Search Objects

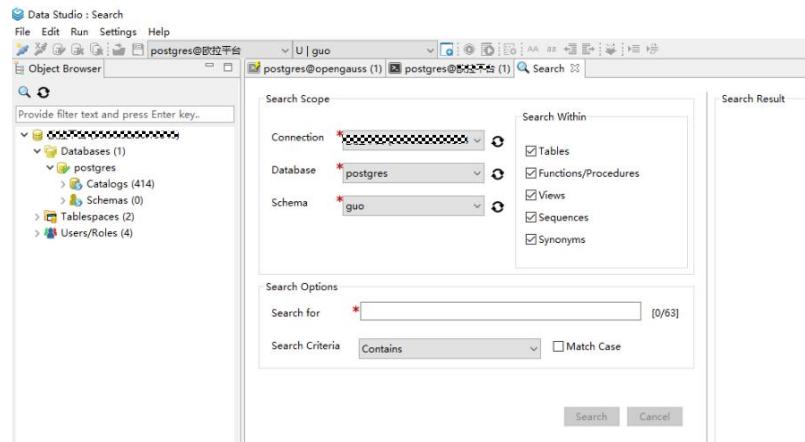
Search Objects option allows you to search for object(s) from the Object Browser based on the search criteria entered. The search operation can be executed either from **Edit > Search**

**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

Search Options	Search Behavior
	displayed.
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>
%	<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</li></ul>

Search Value	Search Behavior
	any number of characters, followed by character "t", followed by any number of characters, and followed by character "e".

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

### 5.4.3 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.
Compile/Execute in New Tab		CTRL+ALT+ENTER	Executes statement in new tab retaining old. Disabled, when <b>Retain Current</b> is selected.
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.

### 5.4.4 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 to modify 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.



Save all data before modifying the language.

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

All the connections are closed and prepare for restart.

If you click **No**, the language cannot be modified even after Data Studio restart.

----End

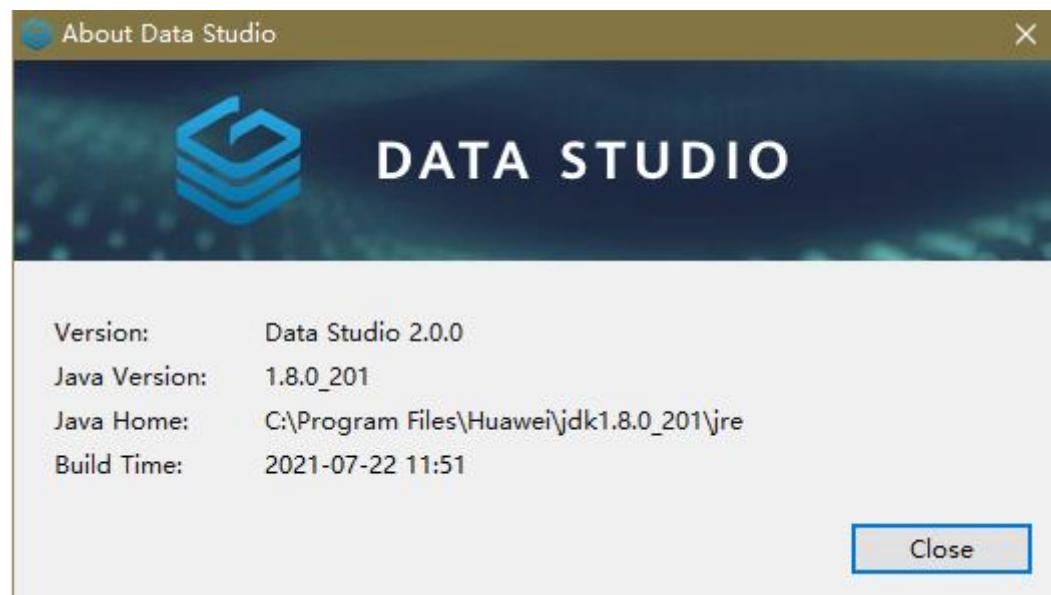
## 5.4.5 Help

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.



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.

## 5.5 Data Studio Toolbars

The following image shows the toolbar:



The toolbar contains the following actions:

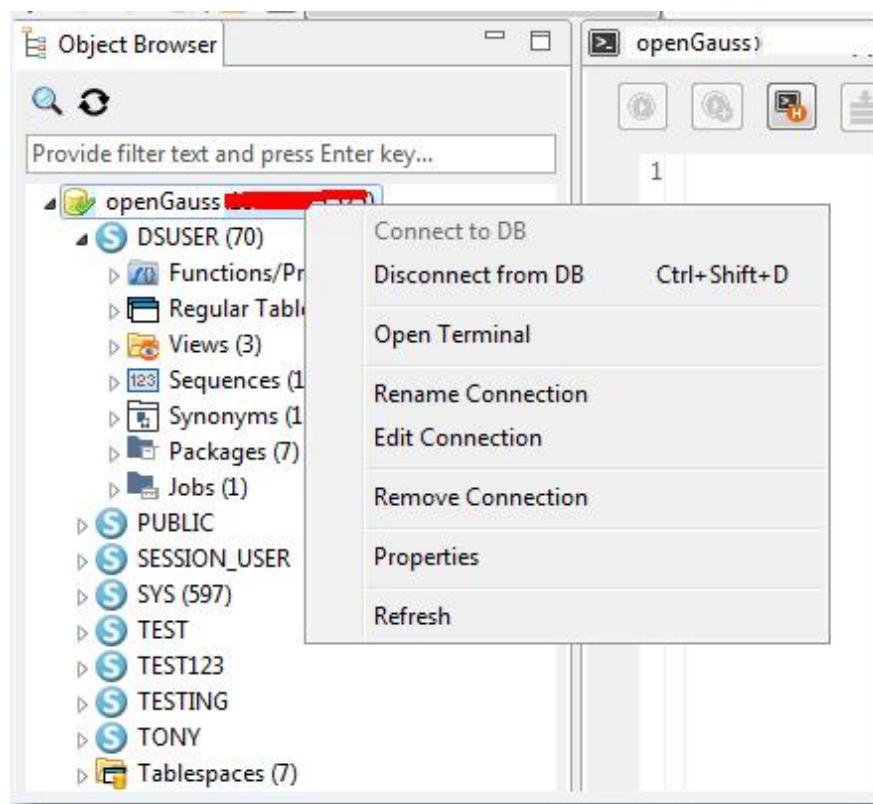
- [6.2.2 Adding a Connection](#)
- [6.2.5 Removing a Connection](#)
- [6.3.3 Connecting to Database](#)
- [6.3.4 Disconnecting Database](#)
- [6.3.2 Disconnecting All Databases](#)
- [6.11.3 Opening And Saving SQL Scripts](#)
- [6.11.3 Opening And Saving SQL Scripts](#)
- [6.11.1 Opening SQL Terminals](#)
- [6.11.1 Opening SQL Terminals](#)
- [6.5.4.6 Executing a Function/Procedure](#)
- [6.5.1 Creating Function/Procedure](#)
- [6.11.6 Formatting of SQL Queries](#)
- [Upper Case](#)
- [Lower Case](#)
- [SQL Assistant](#)

## 5.6 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**, **Remove Connection**, and **Properties** 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**, **Grant/Revoke** 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 **Export DDL**, **Export DDL and Data**, **Rename Schema**, **Drop Schema**, **Grant/Revoke**, and **Refresh** 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

Menu Item	Shortcut Key	Description
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** and **Grant/Revoke** options.

Menu Item	Shortcut Key	Description
Create Function	-	Creates Function
Create Procedure	-	Creates 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** and **Grant/Revoke** options.

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

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

Menu Item	Shortcut Key	Description
Create View	-	Creates View
Grant/Revoke	-	Grant/revokes the access of views
Refresh	F5	Refreshes the View

Right-clicking the **PL/SQL Viewer** allows you to **Cut, Copy, Paste, Select All, Comment/Uncomment Lines, Comment/Uncomment Block, Compile, Execute**.

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	CTRL+/ /	Comment/Uncomment selected line
Remark/Unremark Lines	-	Remark/Unremark selected line
Comment/Uncomment Block	CTRL+SHIFT+/ /	Comment/Uncomment selected Block.
Format	CTRL+SHIFT+F	Formats the selected SQL statements.
Compile	CTRL+ENTER	Compiles the function/procedure
Execute	Ctrl+E	Executes the function/procedure

Right-clicking the **SQL Terminal** allows you to **Cut, Copy, Paste, Select All, Execute Statement, Open, Save, Find and Replace, Execution Plan, Comment/Uncomment, Save As, Format 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	Ctrl+A	Selects all text
Execute Statement	CTRL+ENTER	Executes query
Open	CTRL+O	Opens file
Save	CTRL+S	Saves the query
Find and Replace	CTRL+F	Finds and replaces text in SQL Terminal
Execution Plan	-	Executes the query
Comment/Uncomment Lines	Ctrl+/ /	Comment/Uncomment selected lines
Comment/Uncomment Block	Ctrl+Shift+/ /	Comment/Uncomment selected block of lines
Format	CTRL+SHIFT+F	Formats the selected SQL statements.
Cancel	-	Cancels execution
Save As	CTRL+ALT+S	Saves the query in a new file.

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](#)
- [6.7 Sequences](#)
- [6.8 Views \(Including Materialized Views\)](#)
- [6.9 Tablespaces](#)
- [6.10 Users/Roles](#)
- [6.11 SQL Terminal](#)
- [6.12 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.

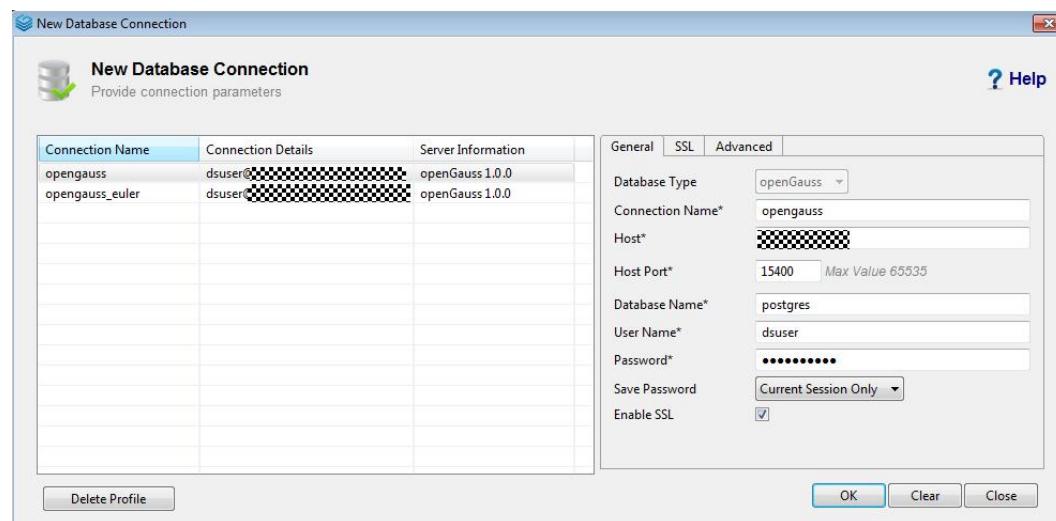
## 6.2.2 Adding a Connection

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**.

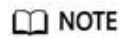
**NOTE**

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

- Clicking **Delete Profile** 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 **Delete Profile** 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.

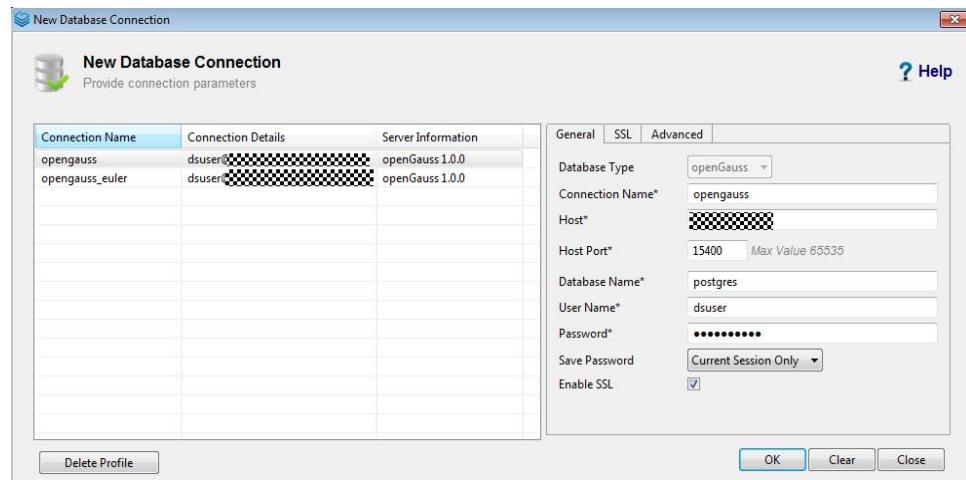
Field Name	Description	Example
Database Type	Select the database type.	openGauss
Connection Name	Provide a connection name.	My_Connection_DB
Host	Provide the IP address (IPv4) or server domain name.  <b>NOTE</b> <ul style="list-style-type: none"><li>• If domain name length is greater than 25 characters, then the complete domain name will not be displayed.</li></ul> <b>Example:</b> <i>test1(db.dws...com:25xxx)</i> <ul style="list-style-type: none"><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:</li><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></ul>	db.dws.mycloud.com 10.xx.xx.xx
Host Port	Provide the port address.	25001
Database Name	Provide the database name.	postgres

Field Name	Description	Example
User Name	Provide the user name to connect to the selected database.	-
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
    - 6.11.10 Working with the SQL Terminals
- **Enable SSL** check box is selected by default.

#### Step 4 Follow the steps to enable SSL:

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.
- openGauss 1.0.1 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.
allow	Selecting allow first attempts to establish a non-SSL connection. If the connection fails, the system attempts to establish an SSL connection.

**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 Warning](#) 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 to set the **Fast Load Options**:

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 Table 8-1 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. Enable **Load child objects of Tables and Views** option. If this option is unchecked then **Child objects of Tables and Views (Columns/Constraints/Indexes)** cannot be loaded during connection . If it sets to true **All child objects of Table/Views** are loaded while Connection and **Load Limit** text box is enabled and user can update load limit. If it sets false then child objects of Table/Views cannot be loaded and **Load Limit** text box is disabled.
6. Enter the load limit in **Load Limit** field. The maximum value allowed is 30000. This is the database object count.

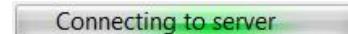
 **NOTE**

- If the number of object types (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:

 Connecting to server

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

 **NOTE**

- Data Studio allows you to log in 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 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.

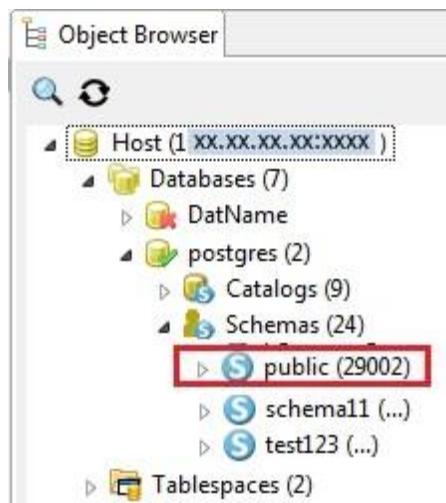
**Step 3** Click **OK**.

----End

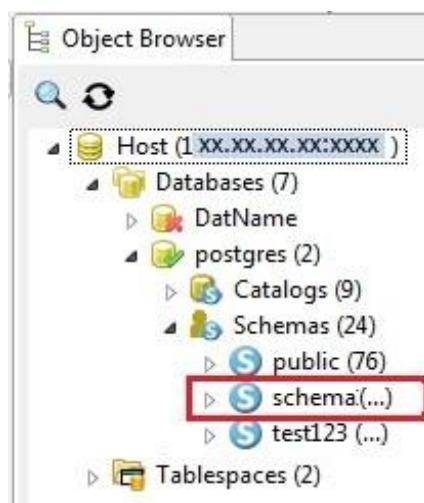
## Lazy Loading

Lazy loading feature delays the loading of objects until required.

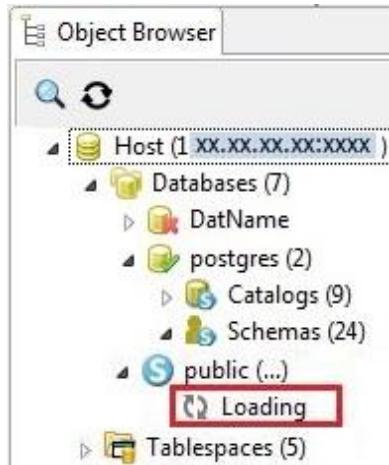
When you connect to a database only child objects of schema saved under **search\_path** will be loaded as shown below:



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 keywords, data types, 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 a user logs in to the Data Studio, **pg\_catalog** is loaded automatically.

### 6.2.3 Renaming a Connection

Follow the steps 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 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.

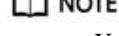


### NOTE

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 log in 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 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

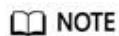
## 6.2.6 Viewing Connection Properties

Follow the steps 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.



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 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 is 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 is 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

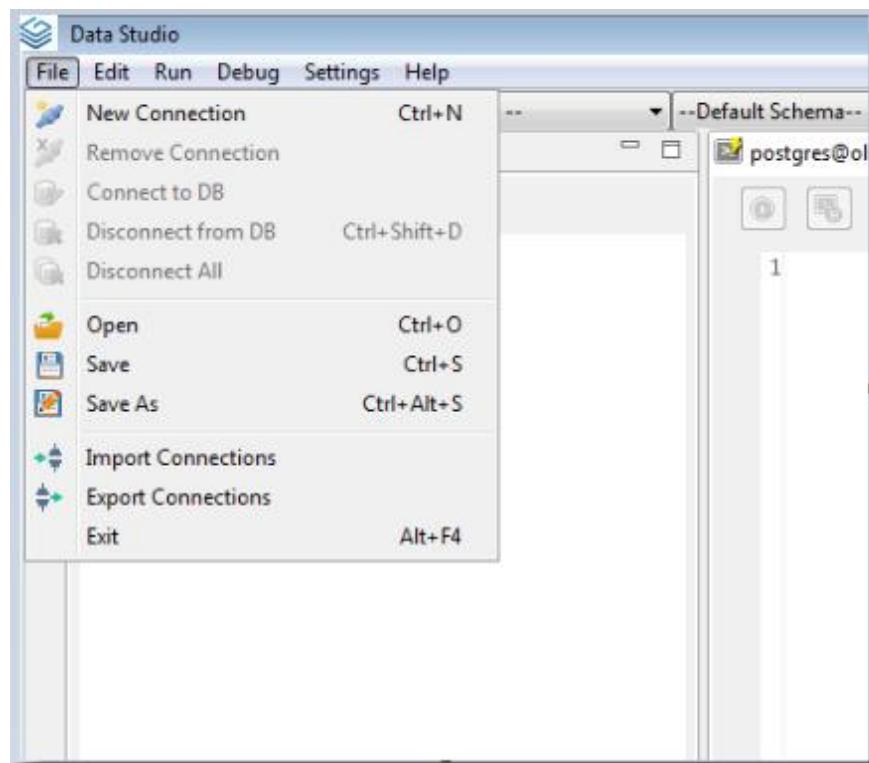
Data Studio provides the option to export/import connection details from the connection dialog for future reference.

Following fields are exported:

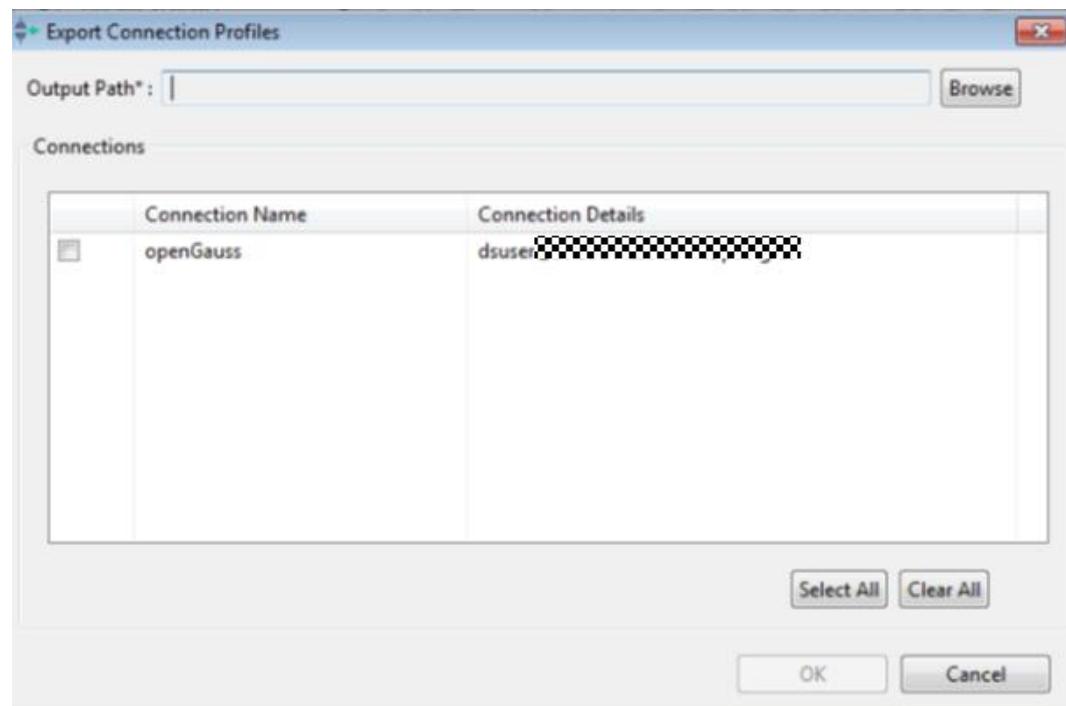
- SSL Mode
- Connection name
- Server IP Address
- Server Port
- Database Name
- Username
- cSSLCertificatePath
- cSSLKeyPath
- rootCertFilePathText
- schemaExclusionList
- schemaInclusionList
- loadLimit

Follow the steps to access the feature:

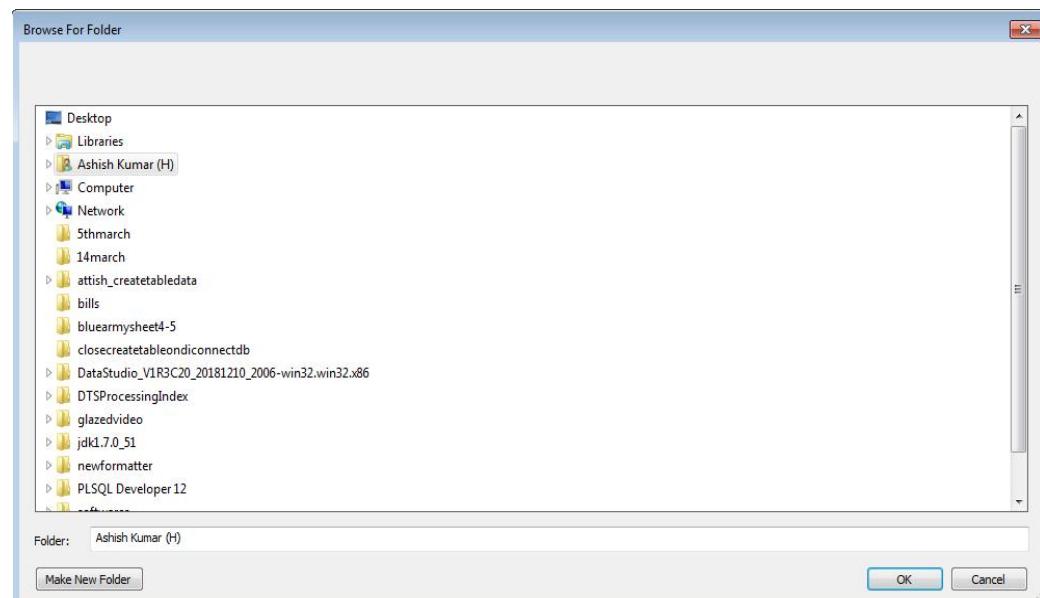
**Step 1** Click **File** on Menu Bar. Following window is displayed:



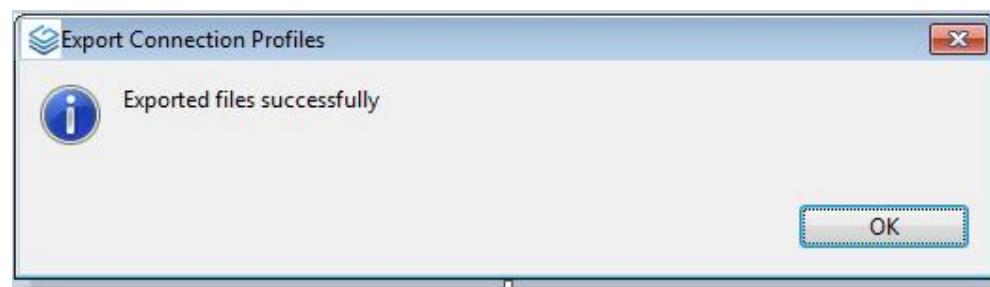
**Step 2** Select **Export Connections** to export the connection profiles. **Export Connection Profiles** window is displayed to the user to select the connections which needs to be exported.



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

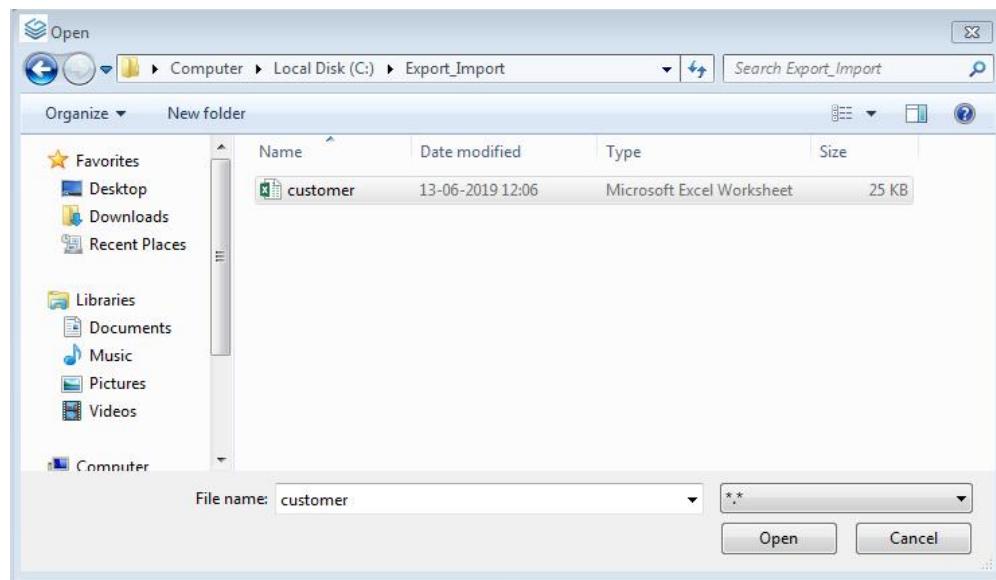


A dialog box is displayed once the connections are exported successfully.



**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 cannot be exported.

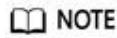
## 6.3 Databases

### 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 to create a database:

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



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.

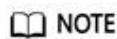
- Step 3** Select the required type of encoding character set from the **Database Encoding** drop-down list.

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.



Data Studio allows you to log in 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 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 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.



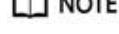
#### NOTE

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.



#### NOTE

Connection properties populates all connection parameters (except password) that were provided during the last successful connection with the database.

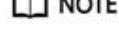
----End

### 6.3.3 Connecting to Database

You can connect to the database.

Follow the steps to connect a database:

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



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 log in 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.
- Data Studio allows to log in based on the preference settings even if the password is expired.  
• 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 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 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 6.12.2 Dropping Batch of Objects section for batch drop.

Follow the steps 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 to view the properties of a database:

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



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.



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**.



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

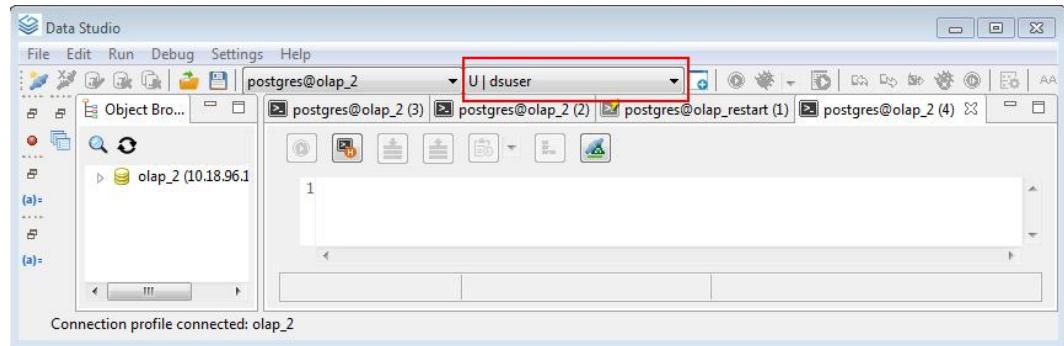
You can perform the following actions on a schema:

- 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)
- Grant/Revoke Privilege (Refer to 6.4.7 Grant/Revoke Privilege 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 the drop down, selected schema will be set as default schema, overriding previous setting.

The selected schema is set as default schema for all active connections of the database (selected in the database list drop down).

## 6.4.3 Exporting Schema DDL

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

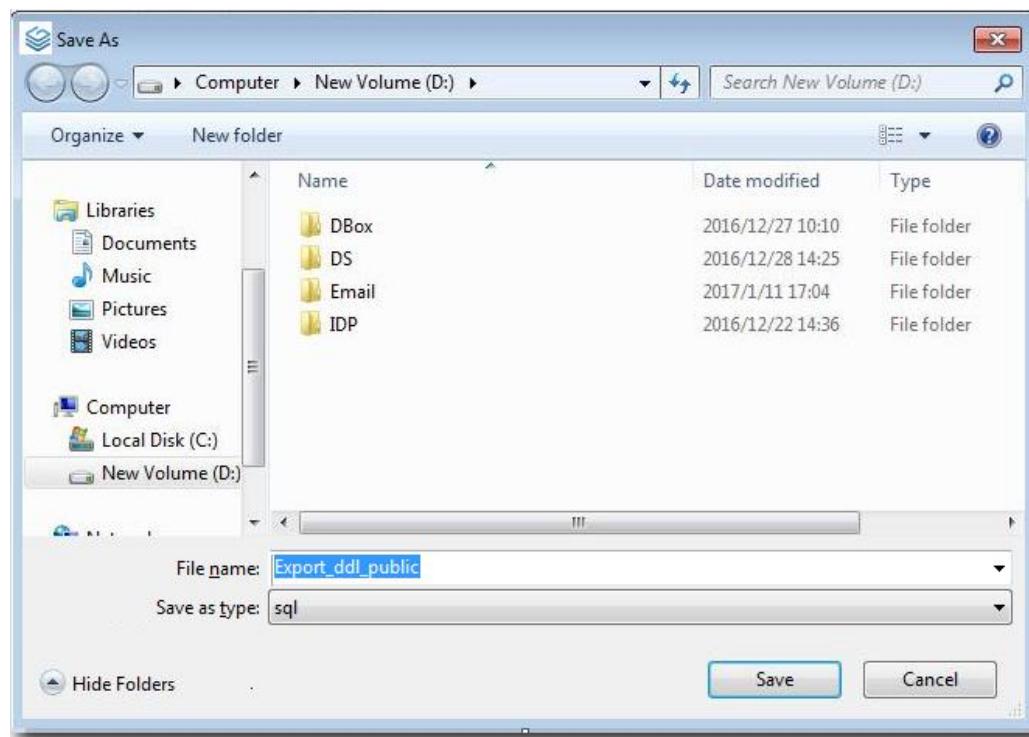
Follow the steps to export the schema DDL:

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

The **Security Warning** dialog box is displayed. You can turn off this security warning message. Refer to [Security Warning](#) 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 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 **Security Warning** dialog box is displayed. You can turn off this security warning message. Refer to [Security Warning](#) 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

Database Encoding	File Encoding	Supports Exporting DDL
	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 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:

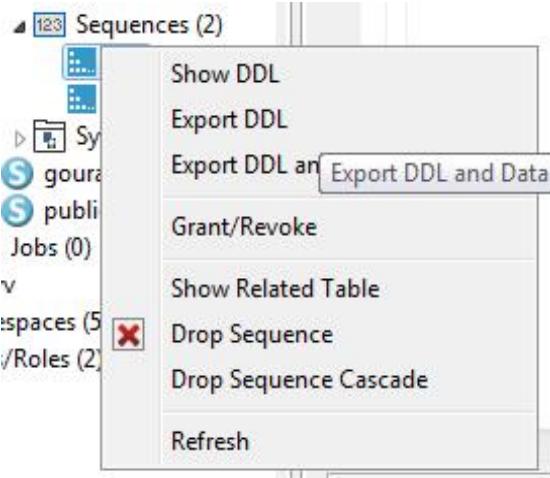
**Step 1** In **Object Browser** right click on any object under **Sequences**. A menu option is displayed.

**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 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 6.12.2 Dropping Batch of Objects section for batch drop.

Follow the steps 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**.

The status bar displays the status of the completed operation.

----End

## 6.4.9 Synonym Management

Data Studio supports synonym management for openGauss.

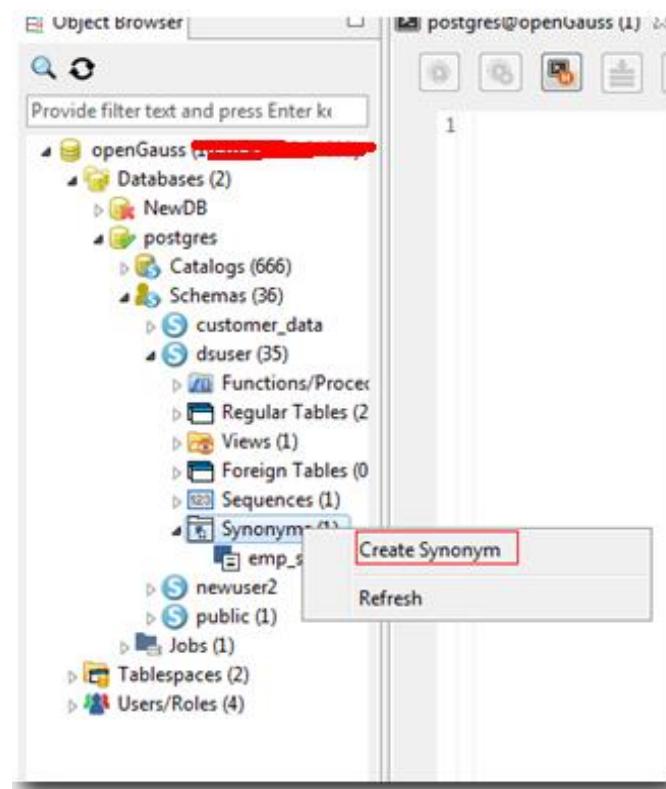
### Prerequisites

Synonyms must be displayed under all schemas.

### Procedure

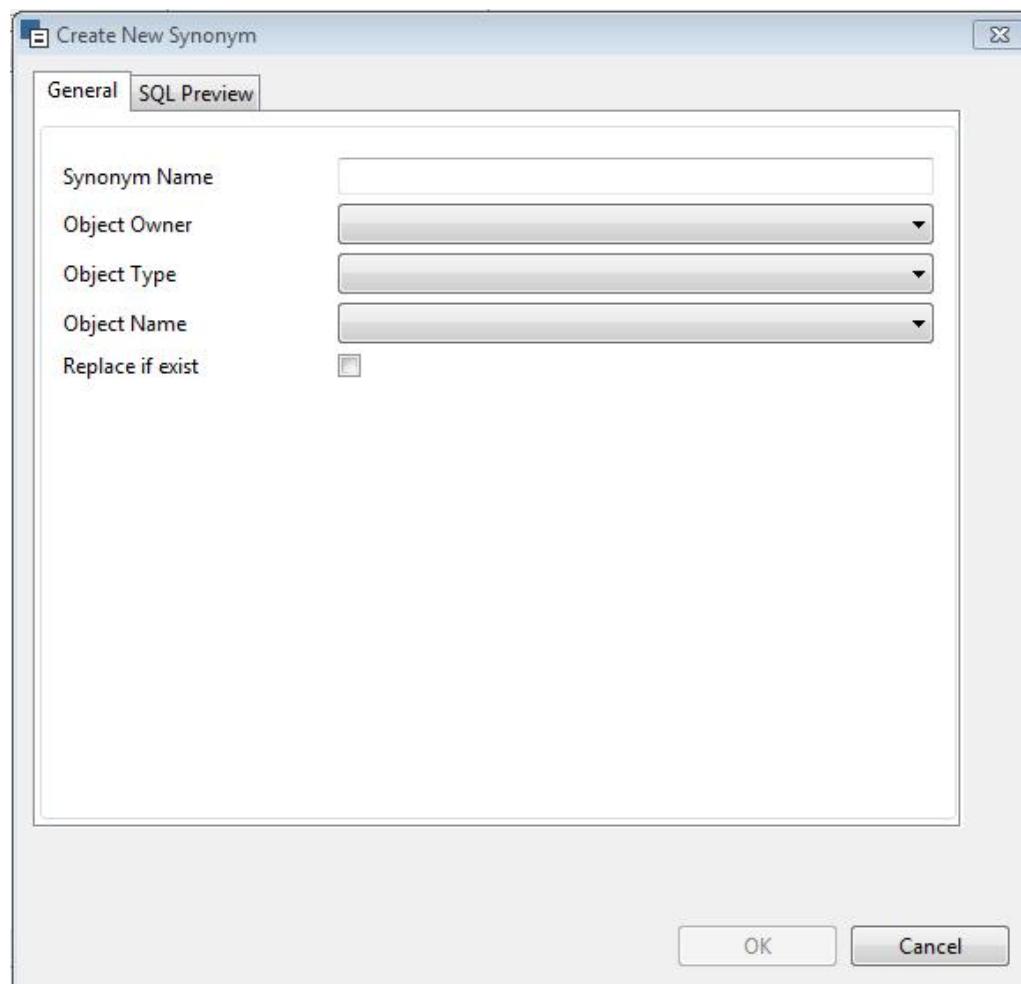
**Step 1** Right-click on the Synonyms item popup to create synonym menu.

**Step 2** Select **Create Synonym**.



**Step 3** Click **Create Synonym**.

The **Create New Synonym** dialog box is displayed.



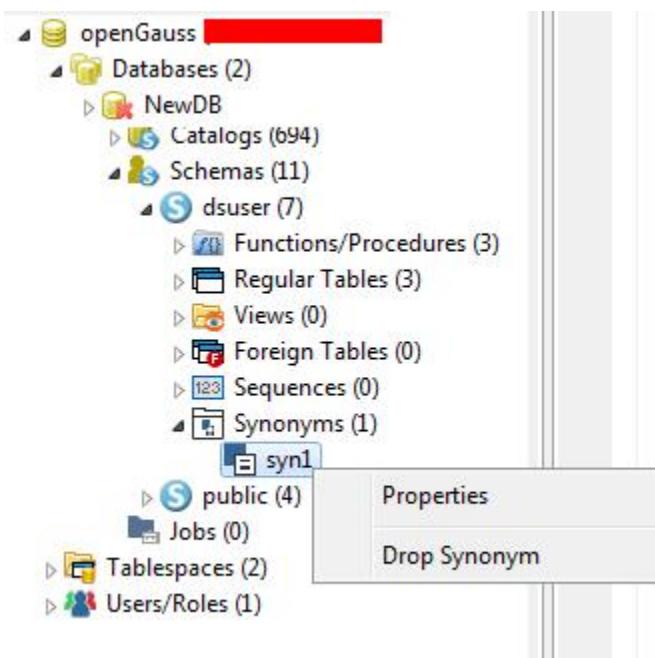
Follow the description of the parameters under **General** tab:

- **Synonym Name:** The name of the synonym.
- **Object Owner:** Shows the list of Owners/Schemas.
- **Object Name:** Shows the object name. Object name is populated based on the object owner.
- **Object Type:** Shows the object type.
- **Replace if exist:** Synonym is updated, if the synonym exists while creating a new synonym.

**Step 4** Click **SQL Preview** to view details.

**Step 5** To refresh synonyms, select **Refresh**.

**Step 6** To drop synonyms, select **Drop Synonym**. Batch drop is applicable.



**Step 7** To know the properties, click **Properties** option.

The details are displayed.

The screenshot shows the 'Properties' dialog box for the synonym 'emp\_synonym'. It contains a table with four rows:

Property	Value
1 Synonym Name	emp_synonym
2 Owner	dsuser
3 Object Owner	dsuser

**Table 6-1** Property Description

SL. No	Property Name	Value
1.	Synonym Name	emp_synonym
2.	Owner	duser
3.	Object Owner	duser
4.	Object Name	emp_data

The following operations can be performed:

**Table 6-2** Operation

Operation	Shortcut Key	Description
	Ctrl+C	Copy
	-	Refresh

Operation	Shortcut Key	Description
	Ctrl+H	Show/Hide
	Ctrl+ S	Search
	-	Clear search text

----End

## 6.5 Functions/Procedures

### 6.5.1 Creating Function/Procedure

Follow the steps 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.

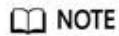
Data Studio opens a new tab with the selected template.

- 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 .

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**.



A popup message displays the status of the completed operation. The status bar does not display the status of this operation.

----End

## Supporting Compile Function

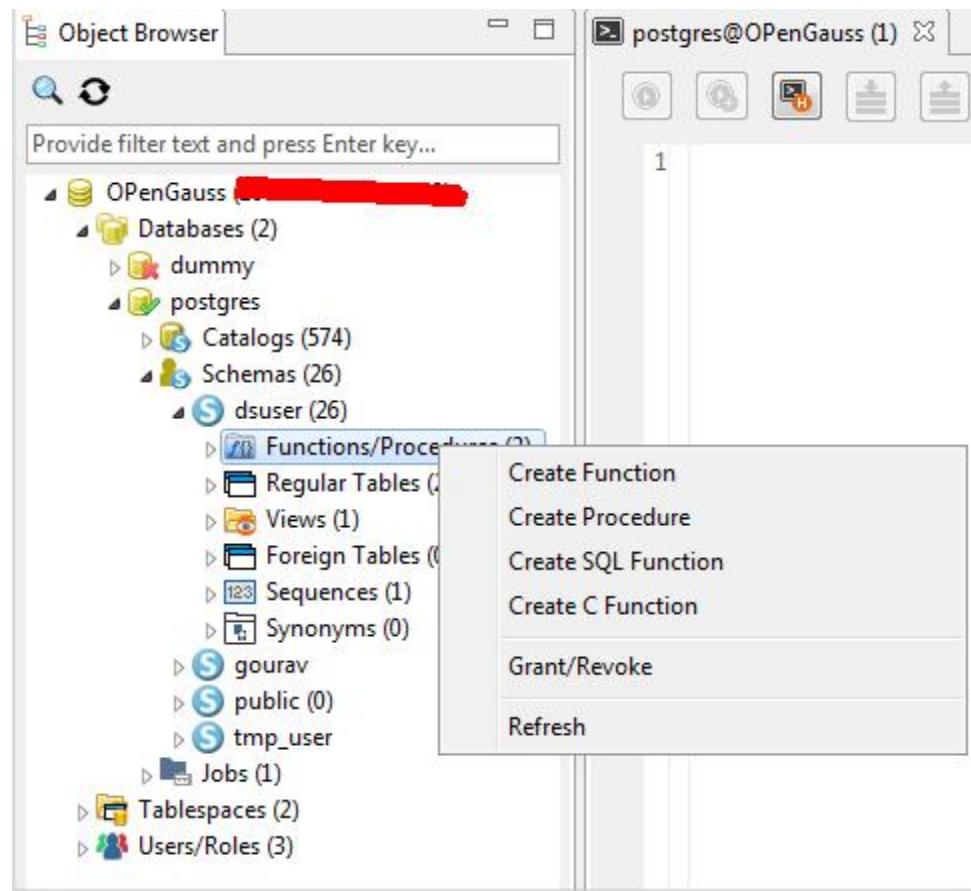
When a new PLSQL Object, either from the template or by editing an existing PLSQL Object, the newly created SQL Object will open in a new tab.

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. The following figure uses openGauss as an example.



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

```
1 CREATE [OR REPLACE] FUNCTION DSUSER.function_name ([ parameter datatype[,parameter datatype] ])
2
3     RETURN return_datatype
4 IS | AS
5
6     /*declaration_section*/
7
8 BEGIN
9
10    /*executable_section*/
11
12 END;
```

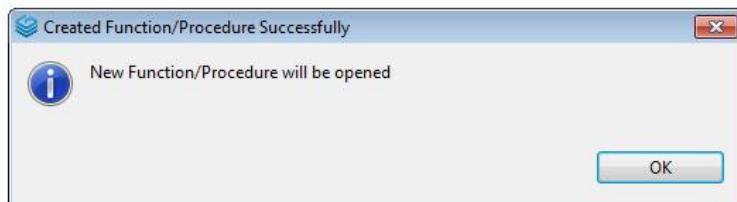
**Step 4** Edit the code.

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

```
7   FOR input_val IN input_val..i_no_of_tables
8     LOOP
9       EXECUTE IMMEDIATE 'CREATE TABLE public.' || 'tbl' || input_val
10      ' ( a int, b
11        dbms_output.put_line ( 'result is: '||input_val );
12      |
13      END LOOP;
14    RETURN 1;
15 END;
16 $$
```

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

```
CREATE OR REPLACE FUNCTION dsuser.add_numbers(a integer, b integer)
RETURNS integer
LANGUAGE plpgsql
NOT FENCED NOT SHIPPABLE
AS $$
```



This function is displayed in a new tab.

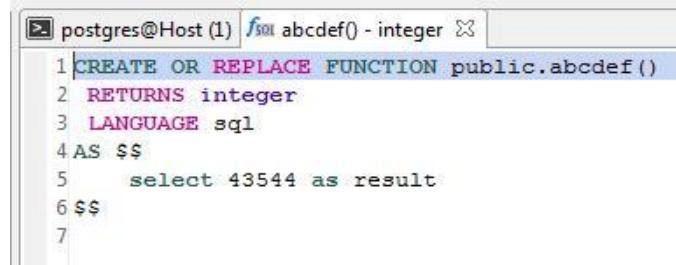
----End

## 6.5.2 Editing a Function/Procedure

Follow the steps 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) f10 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 6.5.4.6 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 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

## 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.

#### NOTE

Data Studio supports plpgsql and SQL languages for the operations are listed as follows:

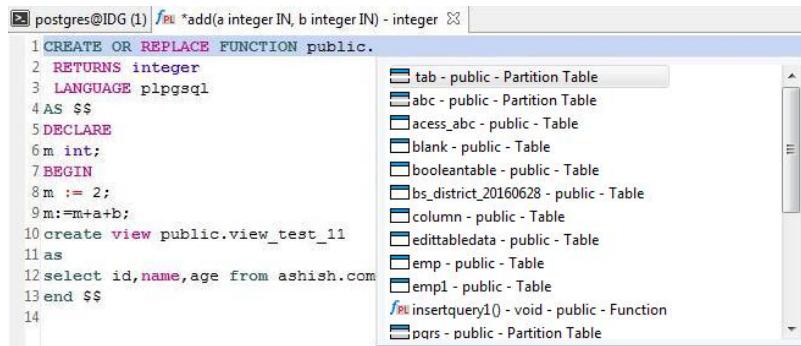
- 6.5.1 Creating Function/Procedure
- 6.5.2 Editing a Function/Procedure
- 6.5.4.3 Exporting a Function/Procedure DDL
- 6.5.4.5 Dropping a Function/Procedure

### 6.5.4.2 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**.

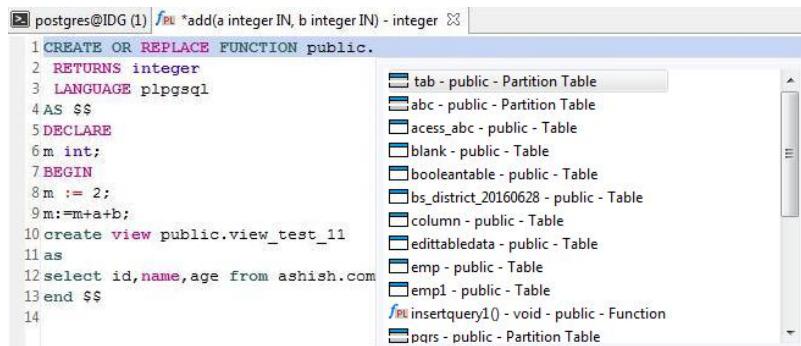
Follow the steps below to select a DB object:

- 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. 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.

- 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 keywords, data types, schema names, table names, views, and table name aliases in the same way as shown above for all schema objects that you have access.

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>
```

- Child objects of table alias will be displayed on pressing dot (.) after alias. The DB objects list **will not be refined** as you continue typing DB object name. To refine DB object name user has to press **CTRL+SPACE** or key configured in preference for 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.

- Auto suggest for table alias is not supported in PL/SQL editor.

**Step 5** Auto suggest popup should be displayed on minimum characters typed by user. The DB objects list is refined as you continue typing the DB object name. Minimum character configuration is provided in **Preference--> Editor--> Auto Suggest**. Default value set is 2 characters. On typing 2nd character matching, character list will be displayed. Refer [7.3 Auto Suggest](#).

----End

### 6.5.4.3 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 **Security Warning** 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
	GBK	Yes
	LATIN1	Yes
GBK	GBK	Yes
	UTF-8	Yes
	LATIN1	No
LATIN1	LATIN1	Yes
	GBK	No
	UTF-8	Yes

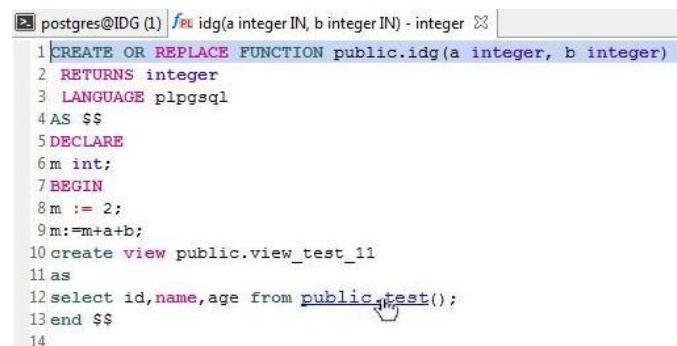
----End

#### 6.5.4.4 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) fpg 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.

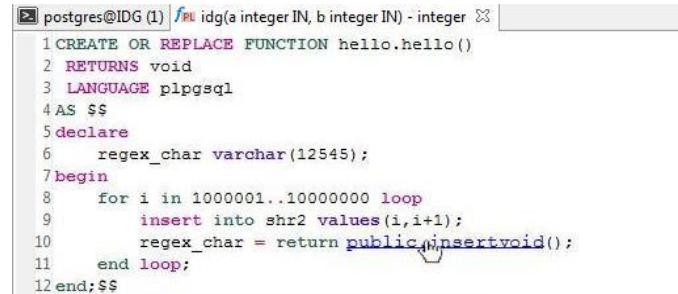


The table properties are read-only.

----End

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

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



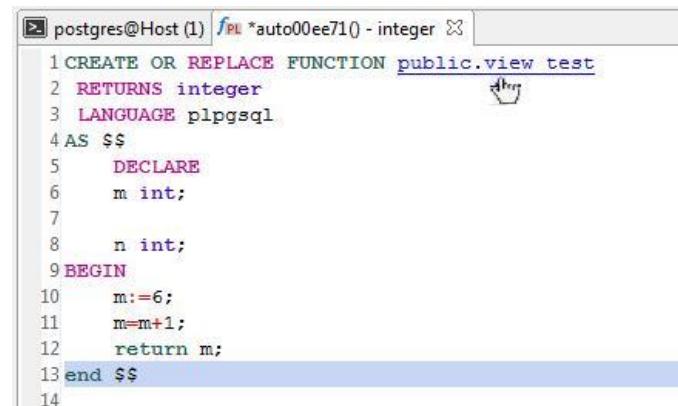
```
postgres@IDG (1) f idg(a integer IN, b integer IN) - integer
1 CREATE OR REPLACE FUNCTION hello.hello()
2 RETURNS void
3 LANGUAGE plpgsql
4 AS $$
5 declare
6   regex_char varchar(12545);
7 begin
8   for i in 1000001..10000000 loop
9     insert into shr2 values(i,i+1);
10    regex_char = return public.insertvoid();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

**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 to View Object DDL:

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



```
postgres@Host (1) f *auto00ee71() - 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

#### 6.5.4.5 Dropping a Function/Procedure

Individual or batch drop can be performed on functions/procedures. Refer to 6.12.2 Dropping Batch of Objects section for batch drop.

Follow the steps 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** To drop objects in batches, right-click two or more selected function/procedure objects and choose **Drop Objects**.

**Step 3** In the confirmation dialog box, click **Yes** to complete the operation successfully.

The status bar displays the status of the completed operation.

----End

#### 6.5.4.6 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')*;
- Data Studio will not execute any function/procedure with unknown data type parameters.

Right-click on the function/procedure in the **Object Browser** has the following functionalities:

- Refresh the program to get the latest program from the server
- Execute the function/procedure or SQL function
- View Source Code
- Drop PL/SQL object
- Export DDL
- Grant or Revoke Privileges

Result tab contains IN and INOUT parameter along with OUT parameter.

All parameter details contain **Name**, **Data Type**, **Parameter Type** and **Value**. OUT parameter row is displayed in grey color to differentiate from IN and INOUT parameter. The following figure uses openGauss as an example.

```

Data Studio : postgres@test123 (7)
File Edit Run Debug Settings Help
postgres@test123 (7) 10
Object Browser
Provide filter text and press Enter key...
auto
bipin_schema
code_beautifier
customer_data
demo
digital
disch
dschema
duser (206)
Functions/Procedures (81)
fn.add(integer IN, integer IN) - integer
fn.add111(integer IN, integer IN) - integer
fn.add2(bigint IN, bigint IN, bit) IN, _varbit IN, _var
fn.add2(integer IN) - integer
fn.add2(interval IN, json IN, hex IN, macaddr IN, mi
fn.add2(numeric IN) - integer
fn.add2(path IN, point IN, polygon IN, real IN, real IN
fn.add2(text IN, time IN, time IN, time IN, time IN, ti
fn.add2(time with time zone IN, varbit IN) - integer
fn.add2(time with time zone IN, timestamp with tim
fn.add2(time with time zone IN, varbit IN) - integer
fn.add2(hquery IN, vector IN, tsid_snapshot IN, us
fn.add2(varbit IN, _varbit IN, _varbit IN) - integer

```

Contains	Enter a search term here.		
Name	Data Type	Parameter Type	Value
1	integer	IN	5
2	integer	IN	5
	integer	OUT	10

If the result is **cursor type** then Data Studio displays 3 dots [...]. Double click on 3 dots displays the result in new popup.

```

Data Studio : postgres@test123 (3)
File Edit Run Debug Settings Help
postgres@test123 (3) 10
Object Browser
Provides filter text and press Enter key...
testproc_out2(integer IN) - refcursor

```

Contains	Enter a search term here.		
Name	Data Type	Parameter Type	Value
1	integer	IN	100
	refcursor	OUT	—

Contains	Enter a search term here.		
1	empid	ename	salary
2	150	bob	2800
3	150	bob	2800
4	150	bob	2800
5	150	bob	2800
6	150	bob	2800
7	150	bob	2800
8	150	bob	2800
9	150	bob	2800
10	150	bob	2800
11	150	bob	2800

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

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

**Step 1** Double-click and open the PL/SQL program or SQL function. 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"c*.

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 6.11.10 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

#### 6.5.4.7 Grant/Revoke Privilege

Follow the steps 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

### 6.5.5 Debugging a PL/SQL Function

#### 6.5.5.1 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.

 NOTE

SQL language function does not support debugging operations.

### 6.5.5.2 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
  - Removing a breakpoint on a line
- A disabled breakpoint will not suspend execution of PL/SQL program.

When you run a PL/SQL program, the execution pauses at every line where you set a breakpoint. When the program execution is paused, Data Studio retrieves information about the current program state, such as the values of the program variables.

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

- Step 1** Set a breakpoint at the line where PL/SQL program execution should pause.
- Step 2** Start the debugging session. When a line with a breakpoint is reached, monitor the state of the application in the debugger pane, and continue the execution.
- Step 3** Close the debugging session.

----End

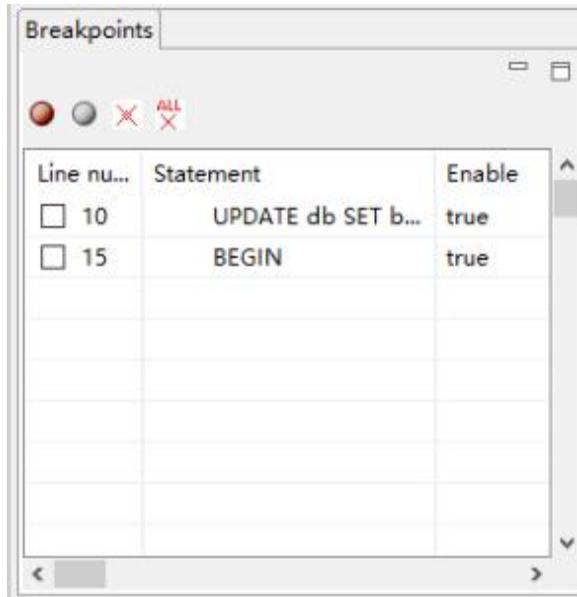
Data Studio provides debugging options in the toolbar that helps you step through the debug objects.

## Using the Breakpoints Pane

You can use the **Breakpoints** pane to view and manage the currently set breakpoints. From the minimized window panel, click the breakpoints option to open the breakpoints pane.

The **Breakpoints** pane lists each breakpoint with the line number and the debug object name.

You can delete all the breakpoints by clicking  in the **Breakpoints** pane. You can enable, disable or delete a specific breakpoint(s) by selecting the breakpoint check box and clicking ,  or  in the **Breakpoints** pane. Double-click the required breakpoint in the Breakpoint Info column to locate the breakpoint in the **PL/SQL Viewer** pane.



#### 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 **Ctrl+C**.

## Setting or Adding Breakpoints on a Line

Follow the steps 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.

## Enabling or Disabling a Breakpoint on a Line

Once a breakpoint is set, you can temporarily disable it by clicking the breakpoint in **PL/SQL Viewer**. Disabled breakpoints will be grayed out [ ] in the **PL/SQL Viewer** and the Enable column of **Breakpoints** pane will turn False.

To enable a disabled breakpoint, select the corresponding breakpoint and click again.

## 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.

## 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.



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 9, 10, 17.

```
postgres@90.90.44.171 (1) postgres@90.90.44.171 (2) debugger_test(integer IN) - integer
1 CREATE OR REPLACE FUNCTION "test"."debugger_test"("i" integer)
2   RETURNS integer
3   LANGUAGE "plpgsql"
4   NOT FENCED NOT SHIPPABLE
5   AS $$*
6   declare
7     v_result int;
8   begin
9     v_result := 0;
10    if i<0 then
11      raise notice 'Please enter i >=0.';
12      raise exception '';
13    end if;
14    for x in 0..i loop
15      v_result := v_result + x;
16    end loop;
17    return v_result;
18  exception
19  when others then
20    v_result := 0;
21    return v_result;
22  end;
23  $$*
24 /
```

- Step 2** To start debugging, click or press F11. The **Debug Function/ Procedure** dialog box appears prompting for your input.



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 and the line turns highlight. The arrow indicates the line number at which execution will resume from. and the red dot mark appears at the beginning of the debugging code line, the beginning of the executed statement is marked with a green dot .

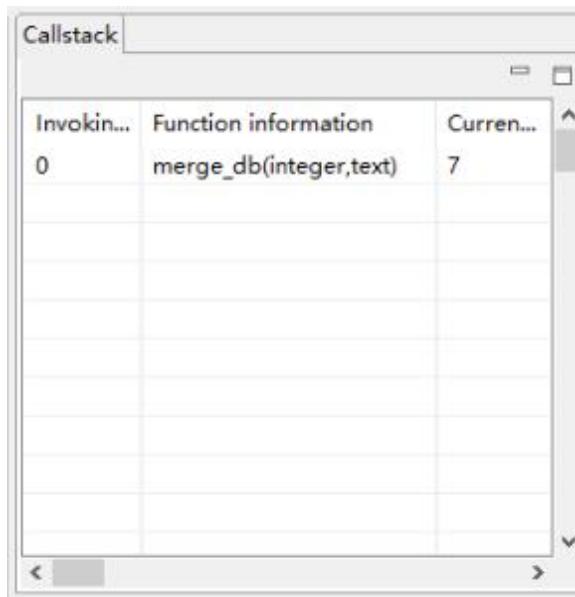
```
1 CREATE OR REPLACE PROCEDURE public.insert_data11(
2 param1 INT = 0,
3
4 param2 CHAR(20),
5
6 param3 CHAR(20),
7 param4 INT = 0 )
8 AS
9 DECLARE
10 BEGIN
11 IF param1 > 100
12 THEN
13     param4=1;
14 END IF;
15 IF param4 > 100 THEN
16     param4=1;
17 END IF;
18 IF param2 > 100 THEN
19     param4=1;
20 END IF;
21 END;
22 /
23 /
```

```
1 CREATE OR REPLACE PROCEDURE public.insert_data11(
2 param1 INT = 0,
3
4 param2 CHAR(20),
5
6 param3 CHAR(20),
7 param4 INT = 0 )
8 AS
9 DECLARE
10 BEGIN
●11 IF param1 > 100
12 THEN
●13     param4=1;
14 END IF;
15 IF param4 > 100 THEN
●16     param4=1;
17 END IF;
18 IF param2 > 100 THEN
●19     param4=1;
20 END IF;
21 END;
22 /
23 /
```

You can terminate debugging by clicking from the toolbar, or pressing **Ctrl+F2**, 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.

The screenshot shows a table titled 'Variables' with four columns: 'Variable', 'Value', 'Dat...', and 'Para...'. Several variables are listed, including 'key', 'data', and several entries starting with '\_gsdb\_sql\_c...' which are highlighted in yellow.

Variable	Value	Dat...	Para...
key	960	inte...	in
data	testing	text	in
_gsdb_sql_c...	<NULL>	bool	syst...
_gsdb_sql_c...	<NULL>	bool	syst...
_gsdb_sql_c...	f	bool	syst...
_gsdb_sql_c...	<NULL>	inte...	syst...
sqlstate	<NULL>	text	temp

You can step through the code using Step In, Step Out or Step Over. For more details, refer to [6.5.5.3 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.

To remove the breakpoint, do the following:

- Double-click again on the breakpoint to remove it from the PL/SQL Viewer.

----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.

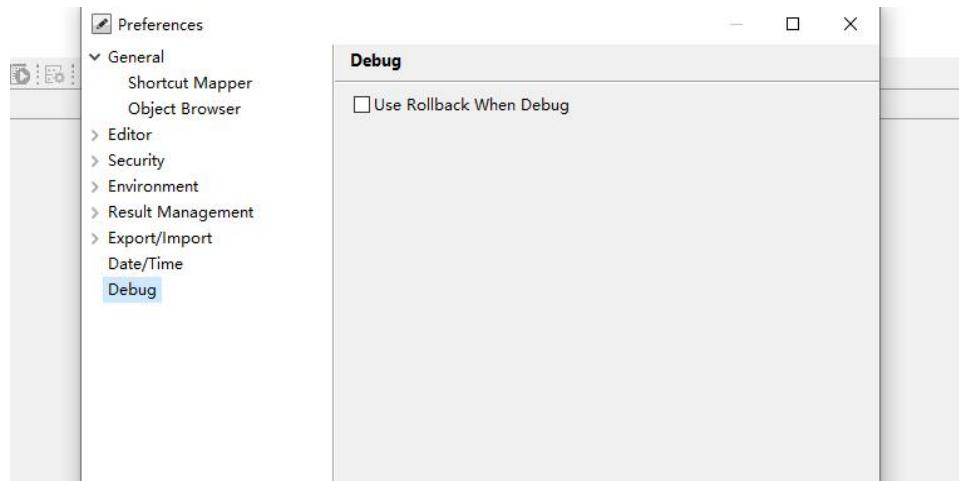
## Supporting Rollback/Commit During Debugging

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

- If **Use Rollback When Debug** option is enabled, then after PL/SQL execution result after debugging is not saved in the database.
- If **Use Rollback When Debug** 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** Click **Setting**, click **Preferences**, choose **Use Rollback When Debug** in **Debug** item.



### 6.5.5.3 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 objects 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 In

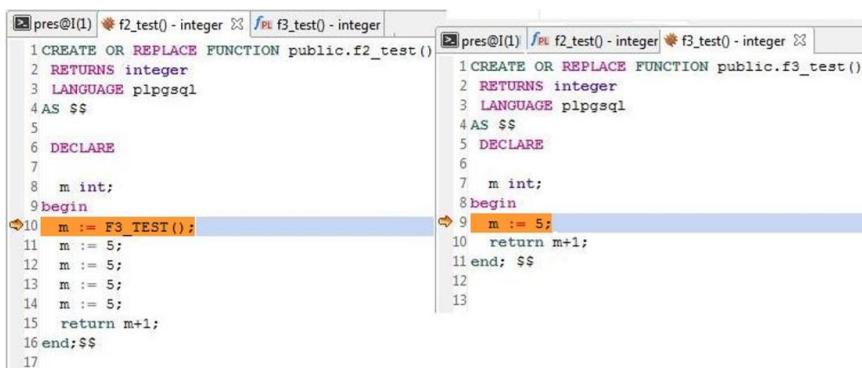
To step through code one statement at a time, select **Step In** from the **Debug** menu, or press button  , or press **F6**.

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 In (F6)** 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 In (F6)** button repeatedly. Once you have stepped through all the statements in that function, Data Studio jumps to Line 10 in f2\_test() .



```
pres@I(1) f2_test() - integer f3_test() - integer
1 CREATE OR REPLACE FUNCTION public.f2_test()
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$

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

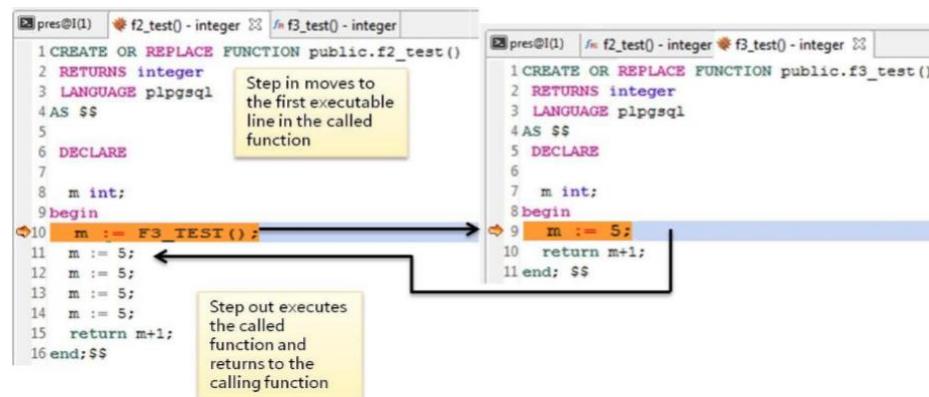
pres@I(1) f2_test() - integer f3_test() - integer
1 CREATE OR REPLACE FUNCTION public.f3_test()
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$

5 DECLARE
6
7 m int;
8 begin
9 m := 5;
10 return m+1;
11 end; $$
12
13
```

## Step 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.

Stepping out of a function will execute a function. The shortcut key for the step out operation is **F7**.



In the preceding example,

- Choose Debug > Step In 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 (F8)** 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.

### 6.5.5.4 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. If the variables changed during debugging, the row of will turn yellow in the **Variables** pane.

Variable	Value	Dat...	Para...
key	777	inte...	in
data	test	text	in
_gsdb_sql_c...	<NULL>	bool	syst...
_gsdb_sql_c...	<NULL>	bool	syst...
_gsdb_sql_c...	f	bool	syst...
_gsdb_sql_c...	<NULL>	inte...	syst...
sqlstate	<NULL>	text	temp

### NOTE

In the **Variables** pane, the parameter value will be displayed as NULL , if the input to the parameter value is string literal 'NULL'.

## 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 [6.11.10 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.

## 6.5.6 Supporting Code Folding/UnFolding

The SQL Terminal supports code folding and unfolding feature. It supports the following scenarios:

- Folding/Unfolding of Procedures and functions blocks:
  - Begin and End Block
  - If and end if Block
  - DML statements (Select, Update, Insert, Delete, Truncate) Blocks
  - DDL Statements (Create, Drop, Alter) Blocks

**Table 6-3** Code Folding and Unfolding Feature

Block	Start	End	Nested
<b>DECLARE, BEGIN and END</b>	Starts when DECLARE or BEGIN Keyword Occurs  <b>DECLARE:</b> Starts when DECLARE Keyword Occurs  <b>BEGIN:</b> Starts when BEGIN Keyword Occurs.	<b>DECLARE:</b> Ends when the Next BEGIN Keyword Occurs.  <b>BEGIN:</b> Ends when the Corresponding END Keyword Occurs.  <b>NOTE</b> <b>Optional Keyword :</b> DECLARE is Optional Keyword.	<b>DECLARE :</b> 1. No Self-Nesting. 2. No Nesting for BEGIN Block. 3. All Remaining Blocks Nesting Supported.  <b>BEGIN :</b> 1. Self-Nesting. 2. All Remaining Blocks Nesting Supported.
<b>If and end if</b>	Starts when IF or ELSE or ELSIF Keyword Occurs	<b>IF:</b> Ends when the Corresponding END IF. Ends when the next ELSE. Ends when the next ELSIF.  If it does not occur then <b>If</b> block will end till end of file.  <b>ELSE:</b> Ends when the Corresponding END IF. Ends when the next ELSIF.  If it does not occur then Else block will end till end of file.  <b>ELSIF:</b> Ends when the Corresponding END IF. If it does not occur then ELSIF block will end till end of file.	1. Self-Nesting. 2. All Remaining Blocks Nesting Supported.
<b>SELECT</b>	Starts when SELECT	1. Ends when ';' delimit occurs.	1. Self-Nesting. 2. All Remaining

Block	Start	End	Nested
	Keyword Occurs	<ul style="list-style-type: none"> <li>2. Ends when Other Block Keyword occur.</li> <li>Except Select with '( ',' UNION ',' INTERSECT' )</li> <li>3. If it does not occur then SELECT block will end till end of file.</li> </ul>	Blocks Nesting not Supported.
<b>INSERT</b>	Starts when INSERT Keyword Occurs.	<ul style="list-style-type: none"> <li>1. Ends when ';' delimit occurs.</li> <li>2. Ends when Other Block Keyword occur without Nesting</li> <li>3. When the INSERT block is in end of file.</li> </ul>	<ul style="list-style-type: none"> <li>1. <b>No</b> Self-Nesting.</li> <li>2. Select statement Nesting.</li> </ul>
<b>UPDATE</b>	Starts when UPDATE Keyword Occurs	<ul style="list-style-type: none"> <li>1. Ends when ';' delimit occurs.</li> <li>2. Ends when Other Block Keyword occur without Nesting.</li> <li>3. When the <b>UPDATE</b> block is in end of file.</li> </ul>	<ul style="list-style-type: none"> <li>1. <b>No</b> Self-Nesting.</li> <li>2. Select statement Nesting. (Select Should be inside '(')</li> </ul>
<b>DELETE</b>	Starts when DELETE Keyword Occurs	<ul style="list-style-type: none"> <li>1. Ends when ';' delimit occurs.</li> <li>2. Ends when Other Block Keyword occur without Nesting</li> <li>3. When the <b>DELETE</b> block is in end of file.</li> </ul>	<ul style="list-style-type: none"> <li>1. <b>No</b> Self-Nesting.</li> <li>2. Select statement Nesting. (Select Should be inside '(')</li> </ul>
<b>TRUNCATE</b>	Starts when TRUNCATE Keyword occurs.	<p>Ends when ';' delimit occurs.</p> <p>If it does not occur then TRUNCATE block will end till end of file.</p>	<ul style="list-style-type: none"> <li>1. No Self-Nesting.</li> <li>2. No other statement Nesting</li> </ul>
DDL Statements (Create, Drop, Alter) Blocks			
<b>CREATE</b>	Starts when CREATE Keyword Occurs	<ul style="list-style-type: none"> <li>1. Ends when ';' delimit occurs.</li> <li>2. Ends when 'CREATE' Keyword Occur again before ';' .</li> <li>3. If it does not occur then CREATE block will end till end of file.</li> </ul>	<ul style="list-style-type: none"> <li>1. <b>No</b> Self-Nesting.</li> </ul> <p><b>NOTE</b></p> <p>Create Table, Create View and so on are applicable.</p>
<b>Procedures, Functions and Packages</b>	Starts when CREATE OR REPLACE Procedures,	<ul style="list-style-type: none"> <li>1. Ends when '\$\$' delimit occurs.</li> <li>2. Ends when 'END' Keyword occurs.</li> </ul>	<ul style="list-style-type: none"> <li>1. All Remaining Block Nesting Supported.</li> <li>2. No check for</li> </ul>

Block	Start	End	Nested
	Functions and Package Keywords Occurs	3. If it does not occur then Procedures and Functions block will end till end of file.	self-nesting.
<b>Alter</b>	Starts when ALTER Keyword Occurs	1. Ends when ';' delimit occurs. 2. Ends when ALTER Keyword Occur again before ';' . 3. When the <b>Alter</b> block is in end of file.	1. <b>No</b> Nesting. <b>NOTE</b> Alter Table, Alter View and so on are applicable.
<b>LOOP, END LOOP</b>	Starts when LOOP Keyword Occurs	1. Ends when the Corresponding END LOOP Keyword Occurs. 2. When the <b>LOOP, END LOOP</b> block is in end of file.	1. All Remaining Block Nesting Supported (as per syntax it is not supported). 2. Self-nesting.

## 6.5.7 History coverage

“SQL 终端”支持代码折叠/展开特性，具体场景如下：

### 6.5.7.1 Overview

The function supports opengauss version 3.0 and above databases.

#### 说明

- Can realize the function history debugging result statistics function.
- Select the list to view for export and deletion

### 6.5.7.2 coverage list

press button ,the pop-up window shows the coverage list of this stored process. You can view the process debugging information as needed, or export the document and delete the corresponding function.



Coverage_Report - 192.168.31.120								
Serial No	Total Rows	Total Running Lines	Total Coverage	Mark Row	Mark Execution L...	Remark Coverage	Input Para...	
1	2	2	100.0%	[8]	[]	0.0%	[2]	
2	2	2	100.0%	[8]	[]	0.0%	[2]	
3	1	2	200.0%	[9]	[9]	100.0%	[2]	
4	2	2	100.0%	[7, 9]	[7, 9]	100.0%	[2]	
5	1	2	200.0%	[9]	[9]	100.0%	[2]	
6	1	2	200.0%	[9]	[9]	100.0%	[2]	
7	2	3	150.0%	[7, 9]	[7, 9]	100.0%	[2]	
8	2	3	150.0%	[7, 9]	[7, 9]	100.0%	[2]	
9	2	2	100.0%	[8]	[]	0.0%	[2]	
10	2	2	100.0%	[8]	[]	0.0%	[2]	
11	0	0	null	null	null	null	[2]	
12	0	0	null	null	null	null	[1]	
13	0	0	null	null	null	null	[null]	
14	2	3	150.0%	[7]	[7]	100.0%	[1]	
15	2	3	150.0%	[7, 9]	[7, 9]	100.0%	[1]	

### 6.5.7.3 Export report

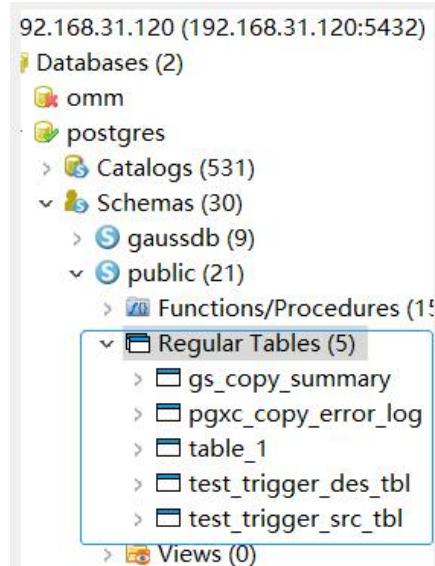
Click the Export button  to export the debugging status at that time. The document contains the debugging status and coverage statistical status (if the stored procedure is not highlighted, the code area and mark execution coverage rate display, if selected highlighted, the debugging information of the highlight part will be displayed).

Serial No	Total Rows	Total Running Lines	Total Coverage	Mark Row	Mark Execution L...	Remark Coverage	Input Para...	
1	2	2	100.0%	[8]	[]	0.0%	[2]	
2	2	2	100.0%	[8]	[]	0.0%	[2]	
3	1	2	200.0%	[9]	[9]	100.0%	[2]	
4	2	2	100.0%	[7, 9]	[7, 9]	100.0%	[2]	
5	1	2	200.0%	[9]	[9]	100.0%	[2]	
6	1	2	200.0%	[9]	[9]	100.0%	[2]	
7	2	 HINT: The report has been downloaded to: E:\whq\项目资料\01交付中 --openGauss\DataStudio 适配dbe_pidebugger以及存储过程覆盖\开发版本 \Data Studio\16391_func111_1667201833601.html						
8	2							
9	2							
10	2							
11	0							
12	0							
13	0							
14	2							
15	2	3	150.0%	[7, 9]	[7, 9]	100.0%	[1]	

Execute statement								
Serial Number	Total Rows	Total Running Lines	Total Coverage	Mark Row	Mark Execution Line	Marker Coverage	Input Params	Update Time
1	2	2	100.0%	[7, 9]	[7, 9]	100.0%	[2]	2022-10-28 18:20:43

The debugging result will automatically generate the corresponding table structure file in the Regular Tables, refer to 6.6.6 Managing Table.



**NOTE**

**The same plpgsql can select and export multiple execution records, and saved to the system installation directory in html file format.**

----END

#### 6.5.7.4 Delete / Delete all

Click delete and delete all to delete the coverage information of the stored procedure as required.

## 6.6 Tables

### 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.
- Constraints creation, renaming and dropping can be supported for regular table and partition table.
- Automatically refresh updates object browser after creating, altering and deleting table (Regular/Partition). Auto refresh is applicable for view creation and rename the created view also.

## 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 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](#).
- 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** Define the column constraints for different constraint types such as primary key, unique, check, and foreign key. For more details, refer to [Defining Table Constraints](#).
- Step 5** 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.



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

Following table lists the supported fields for each Regular type tables:

**Table 6-4** Supported Fields

Field Name	Row Table	Column Table
Tablespace	Normal	Normal
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:

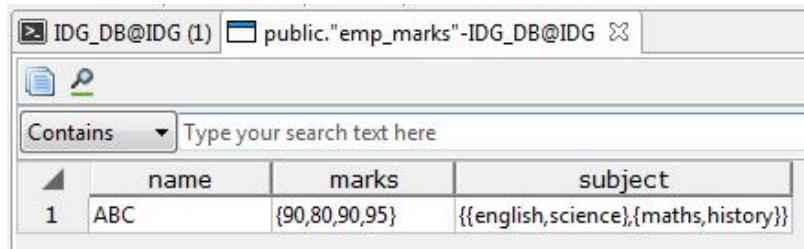
- Step 1** Enter the column name in **Column Name** field. It specifies the name of a column to be created in the new table. This must be a unique name in the table.

 **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.
- If XML data format is disabled for the database and select XML for a column to create a table, the error message "unsupported XML feature" will be displayed.

- 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** To include comments for **Column** in **Create Regular Table**, add column information in **Description of Column (Max 5000 chars)** text box and click **Add** button. This is applicable for at **Right click New Column** dialog box in the object browser and **Table Properties Window** for regular table.

**Step 8** 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

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

**Table 6-5** Supported Fields

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

## Deleting a Column

Follow the steps to delete a column:

**Step 1** Select the required column.

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

----End

## Editing a Column

Follow the steps 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**.

## 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
- Foreign 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.

### Foreign Key

A **Foreign Key** is a column or a combination of columns whose values match a Primary **Key** in a different table.

- Step 1** Select **FOREIGN KEY** from the **Constraint Type** drop-down list box and enter **Constraint Name**.
- Step 2** Select the object column from **Available Column**.
- Step 3** Choose the namespace, table name, and column name to verify the related column of the foreign key.
- Step 4** Ensure that the corresponding namespace contains tables and the table is refreshed in the object browser, or the **Table Name** may be empty.
- Step 5** Select **Add** to add foreign key. Select **Delete** to drop foreign key.

----End

## 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 remove unique from the list using **Delete** button.

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

## 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 6-6** Supported Options

Option Name	Row Table	Column Table
-------------	-----------	--------------

Option Name	Row Table	Column Table
Check	✓	✗
Unique	✓	✓
Primary Key	✓	✓
Foreign Key	✓	✗
Partial Cluster 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.

For Partition Table, we can select **Index Type** including Global index and Local index when creating index for it. If no index type is selected, global index is created by default.

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 to minimize the index's physical size. For heavily updated tables, an 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 6-7** Supported Fields/Options

Field/Option Name	Row Table	Column Table
Unique Index	✓	✓
btree	✓	✓
cmtree	✗	✓

Field/Option Name	Row Table	Column Table
cgin	X	✓
gin	✓	✓
gist	✓	X
hash	X	X
psort	X	✓
spgist	X	X
User Defined Expression	✓	✓
Partial Index	✓	X

#### NOTE

- Fill Factor: Index other than the GIN and psort support the FILLFACTOR parameter.
- Tablespace: Specifies the tablespace for an index. If no tablespace is specified, the default tablespace is used.
- Index Type: Index Type only can be selected by partition tables.

## 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 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 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 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 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 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 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.
- If the database doesn't support the XML data type and do the operation to change data type to XML, the error message "unsupported XML feature" will be displayed.

**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 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.
- We can create Partial Cluster Key for Column Tables.

----End

## Renaming a Constraint

Follow the steps 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 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 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 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 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 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 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 cannot be created in SQL Terminal. It can be viewed in the Object browser after refresh after created on the database server.

- 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.

----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 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** Define the column constraints for different constraint types such as primary key, unique, and check. For more details, refer to [Defining Table Constraints](#).
- Step 5** 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 6** 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](#).

- Step 7** To include comments for **Column** in **Create Partition Table**, add column information in **Description of Column (Max 5000 chars)** text box and click **Add** button.

----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 as follows:

- Step 1** Select table orientation from **Table Orientation** drop-down.
- Step 2** After providing the general information about the table, click **Next** to define the columns information for the table.

Following table lists the supported fields for each Partition type tables:

**Table 6-8** Supported Fields

Field Name	Row Partition	Column Partition
Tablespace	Normal	Normal
Table Type	X	X
If Not Exists	✓	✓
With OIDS	X	X
Fill Factor	✓	X

----End

## Defining Columns

Refer to [Defining Columns](#) to define column(s) for the table.

Following table lists the supported fields for each partition type tables:

**Table 6-9** Supported Fields

Field Name	Row Partition	Column Partition
Array Dimensions	✓	X
Data type Schema	✓	X
Not Null	✓	✓
Default	✓	✓
Unique	✓	X
Check	✓	X

## 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**.

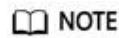
## SQL Preview

Refer to [SQL Preview](#)

## Editing a Partition

Follow the steps 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.



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

----End

## Deleting a Partition

Follow the steps to delete a partition:

- Step 1** Select the required partition.
  - Step 2** Click **Delete**.
- End

## Defining Partitions

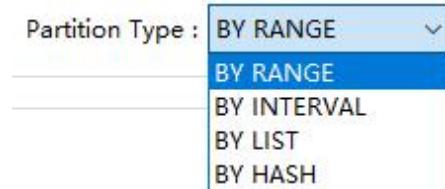
Following table lists the supported fields/options for each Partition type tables:

**Table 6-10** Supported Fields/Options

Field/Option Name	Row Partition	Column Partition
Partition Type	By Range	By Range
	By Interval	✗
	By Hash	✗
	By List	✗
Partition Name	✓	✓
Partition Value	✓	✓
Tablespace	Normal	Normal

Follow the steps to define partition(s) for the table:

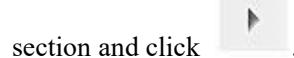
**Step 1** If **Row** is selected as **Table Orientation** in the **General** tab, then **By Range**, **By Interval**, **By Hash**, **By List** are displayed in the **Partition Type** section.



 **NOTE**

- For details about the data types supported by different partition types, see related documents.
- If **COLUMN** is selected as **Table Orientation** in the **General** tab, then **By Range** 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.
- 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.

**Step 6** After you enter all information for partition, click **Add**.

**Step 7** After defining all partitions, click **Next**.

----End

You can perform the following operations on an existing partition for Row or Column Partition table. :

- [Deleting a Partition](#)
- [Editing a Partition](#)

## Defining Indexes

Refer to [Defining Indexes](#) to define table indexes.

**Table 6-11** Supported Options

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

## Defining Table Constraints

Refer to [Defining Table Constraints](#) to define table constraint(s).

**Table 6-12** Supported Options

Option Name	Row Partition	Column Partition
Check	✓	✗
Unique	✓	✓
Primary Key	✓	✓
Foreign Key	✓	✗
Partial Cluster Key	✗	✓

### 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 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 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

## Viewing Partition Data

Follow the steps to viewing partition data:

**Step 1** Right-click the selected partition and select **Viewing Partition Data**.

**Viewing Partition Data** tab is displayed, and show data for the specific partition.

----End

## 6.6.5 Grant/Revoke Privilege - Regular/Partition Table

Follow the steps 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 6-13** Table Context Menu Options

Option	Description
View Table Data	Opens the table data information. Refer to 6.6.7.7 Viewing Table Data
Edit Table Data	Opens the edit table data window. Refer to 6.6.7.8 Editing Table Data
Reindex Table	Performs reindexing a table operation. Refer to 6.6.6.4 Reindexing a Table
Analyze Table	Analyzes a table. Refer to 6.6.6.5 Analyzing a Table
Truncate Table	Truncates table data. Refer to 6.6.6.3 Truncating a Table
Vacuum Table	Vaccum table data. Refer to 6.6.6.6 Vacuuming a Table
Set table Description	Sets the table description for the table. Refer to 6.6.6.7 Setting the Table Description
Set Tablespace	Sets the tablespace for the table. Refer to 6.6.6.8 Setting the Tablespace
Set Schema	Sets the schema for the table. Refer to 6.6.6.9 Setting the Schema
Export Table Data	Exports the table data. Refer to 6.6.7.4 Exporting Table Data
Import Table Data	Imports the table data. Refer to 6.6.7.6 Importing Table Data
Show DDL	Shows the DDL of the table. Refer to 6.6.7.5 Showing DDL
Export DDL	Exports the DDL of the table. Refer to 6.6.7.2 Exporting Table DDL

Option	Description
Export DDL and Data	Exports the DDL and data of the table. Refer to 6.6.7.3 Exporting Table DDL and Data
Rename Table	Renames the table. Refer to 6.6.6.2 Renaming a Table
Drop Table	Drops (deletes) the table. Refer to 6.6.6.10 Dropping a Table
Properties	Displays the properties of the table. Refer to 6.6.6.11 Viewing Table Properties
Grant/Revoke	Grants/Revokes object permissions. For details, see 6.6.6.12 Grant/Revoke Privilege
Refresh	Refreshes a table

### 6.6.6.2 Renaming a Table

Follow the steps 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.

----End

### 6.6.6.3 Truncating a Table

Follow the steps 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 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.

---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 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 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 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 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.

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 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

- 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 6.12.2 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 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 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, Index and Partition) 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
Partition	Copy <b>NOTE</b> Partition tab Only for Partition Table, partition tab cannot be modified.

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 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 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.

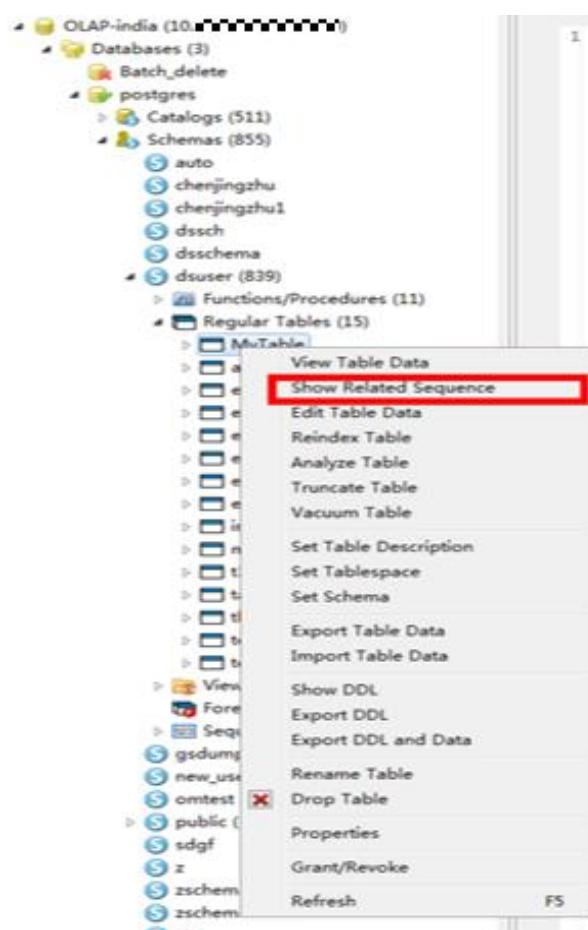
----End

### 6.6.6.13 Show Related Sequences

openGauss support for displaying dependencies between tables and sequences.

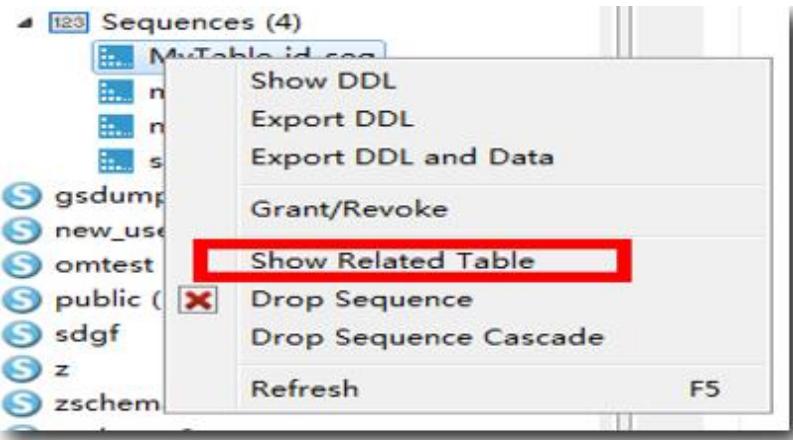
Follow the steps to show related sequences:

**Step 1** Right click menu in table's name.



#### NOTE

You can right click menu in sequence's name.



----End

## 6.6.7 Managing Table Data

### 6.6.7.1 Overview

This section describes how to manage table data.

**NOTE**

Foreign tables support only ER graphs, showing DDLs, and viewing 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 **Export DDL and Data** 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 (msvcrt100.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

Database Encoding	File Encoding	Supports Exporting DDL
	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
- Data 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 **Export DDL and Data** 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

Database Encoding	File Encoding	Supports Exporting DDL
	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 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), Text or binary format. By default Excel (xlsx) is selected.
- **Include Header** - This option is available for Text files. If selected, it will include the column headers. By default, this option is selected when exporting to 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 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 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 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. Database and Data Studio encoding should be compatible with file encoding to export proper data. Refer Table 11-1 and Table 11-2.
- **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 Text format "," will be the default delimiter. Maximum of 10 bytes can only be entered in the **Other** field. For 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.

- **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 Warning** - The security warning 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 Warning 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:
  - Click **OK** to close the connection profile.
  - 1. Clean the disk.
  - 2. 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

## Cancelling the export table data operation

Follow the steps 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 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 (msvcrt100.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 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 EXCEL, CSV, Text or binary format. By default EXCEL 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. Database and Data Studio encoding should be compatible with file encoding to import proper data. Refer Table 11-1 and Table 11-2.
- **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

## Cancelling the import table data operation

Follow the steps 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 <a href="#">View Query Results</a> 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 <a href="#">Execute SQL Queries</a> 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 up to 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.
  - Foreign tables can only be viewed by "select" DDL.
- 

### 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

## View of Unstructured Data for openGauss

Data Studio supports view of following data type in Editor:

- [Bytea](#)

### Bytea

The information about Data tab for bytea type is as follows:

- The display of data has "\x" prepended to the hex string, to know that the format is hexadecimal and in lower case.
- When you enter the data manually, then there are two scenarios:
  - Data can be entered without "\x" at start. This can be a normal ASCII string input. Once OK is pressed, it is converted to hexadecimal first. This can see by viewing the dialog again. There is no key restriction.
  - Data entered with "\x" at start is considered as a hexadecimal. Hence, there will be key restriction [a-fA-F0-9]. Also, the length of input followed by "\x" cannot be odd. If it is odd, then the corresponding message is displayed.

#### NOTE

- If content is of the type other than image, data is shown but image tab displays are not supported.
- The content size cannot exceed 4 MB. If so, then preview will not be available.
- When you click **Export All**, any format other than Excel have bytes instead of watermark.

### 6.6.7.8 Editing Table Data

Follow the steps 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, xml, serial, 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 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 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 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 6.11.12 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

## 6.6.9 Supporting ER for openGauss

Data Studio provides mechanism to view the schema (tables group), table level ER Diagram.

### NOTE

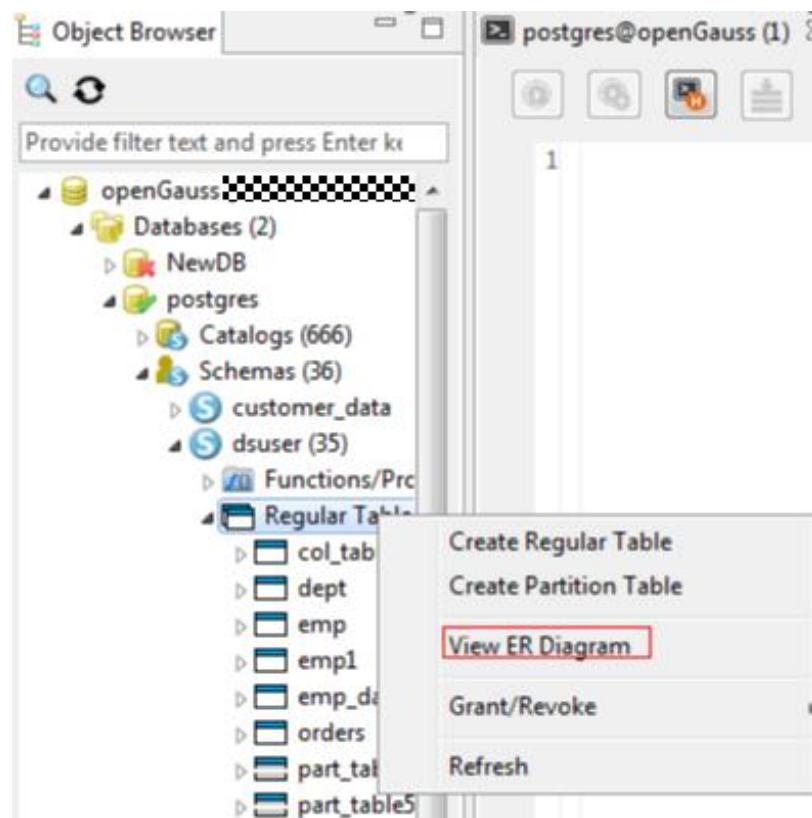
In Linux environment, this feature is available only if Oracle JDK is used.

## Procedure

**Step 1** Go to **Object Browser > Regular Tables**.

**Step 2** Right-click on **Regular Tables**.

**View ER Diagram** context menu is displayed.



### Step 3 Select View ER Diagram.

ER Diagram is displayed.



### View Style

**Show Icons** - Column Names and Icons are displayed by default.

**Show Data Types** - Display Data types and Precision / Size of the column.

**Show Nullability** - Display the Nullability of column values.

**Show Comments** - Display the table comments and column comments if exists.

**Show Fully qualified names** - Display the table name along with the owner name.

### **Show Attributes**

**All** - All attributes are displayed by default.

**Any Keys** - Display the Primary Key and Foreign Key attributes.

**Primary key** - Display the Primary Key of the entity.

**None** - Display only the table name. No other attributes are displayed.

- ToolBar: The toolbar displays various attributes which are used in ER Diagram of table.  
To know about View Style and Show Attribute, right click and select as follows:

Following Icon indicates the Primary Key and Foreign Key in ER Diagram.



- This indicates Primary Key.



- This indicates Foreign Key.

Table header color indicates the Current and Related tables.



- This indicates Current Table.



- This indicates Related Table.

- Display of multiple tables: Display all the tables of that selected schema.



When selecting single table also, it shows the table and its associated linked tables.

----End

## 6.7 Sequences

### 6.7.1 Creating Sequence

Follow the steps 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:

- Enter a name in **Sequence Name** field.



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".

- Enter minimum value in **Minimum Value** field.
- 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.

 **NOTE**

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.

 **NOTE**

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

----End

## 6.7.2 Grant/Revoke Privilege

Follow the steps 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

## 6.7.3 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 6.12.2 Dropping Batch of Objects section for batch drop.

Follow the steps 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 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

## Grant/Revoke Privilege

Follow the steps to grant/revoke privilege:

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

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

**Step 2** Refer to 6.7.2 Grant/Revoke Privilege section to grant/revoke privilege.

----End

# 6.8 Views (Including Materialized Views)

## 6.8.1 Creating a View

Follow the steps 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**.

 **NOTE**

The status bar will not display message on completion of this operation.

----End

## 6.8.2 Grant/Revoke Privilege

Follow the steps 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

## 6.8.3 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 to export view DDL:

**Step 1** Right-click the selected view and select **Export DDL**.

**Export DDL** 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 [6.12.2 Dropping Batch of Objects](#) section for batch drop.

Follow the steps 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 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 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 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 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 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 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 to grant/revoke privilege:

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

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

**Step 2** Refer to 6.8.2 Grant/Revoke Privilege section to grant/revoke privilege.

----End

## 6.9 Tablespaces

### 6.9.1 Creating a Tablespace

You can create tablespaces to optimize performance of database objects.

After creating a new 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 to create a tablespace for openGauss, 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.

- **Relative Path:** The LOCATION directory is relative to the data directory of each database node. Each relative paths contains a maximum of two levels.
  - **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.
  - **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.

---

**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**

## 6.9.2 Working with Tablespaces

You can perform the following operations on an existing tablespace:

- [Set Tablespace Option](#)
- [Set Tablespace Maxsize](#)
- [Show DDL](#)
- [Rename Tablespace](#)
- [Drop Tablespace](#)
- [Refresh](#)

### Set Tablespace Option

Follow the steps to set tablespace option for openGauss database:

**Step 1** Right-click the selected tablespace and select **Set Tablespace Option**.

The **Set Tablespace Option** dialog box is displayed.

**Step 2** Enter the value of **Random Page Cost** and **Sequential Page Cost**.

**Step 3** Click **OK** to save changes.

----End

### Set Tablespace Maxsize

Follow the steps 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

### Show DDL

Follow the steps to show DDL of the tablespace:

**Step 1** Right-click the selected tablespace and select **Show DDL**.

The DDL of tablespace is displayed at **SQL Terminal** tag.

----End

## Rename Tablespace

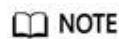
Follow the steps to rename a tablespace for openGauss database:

**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

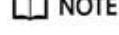
System tablespaces are displayed in the **Object Browser** by default.

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

----End

## Drop Tablespace

Follow the steps to drop the tablespace for openGauss 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

## Refresh

**Step 1** Right-click the selected tablespace and select **Refresh**.

----End

## 6.10 Users/Roles

### 6.10.1 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.

### 6.10.1.1 openGauss

The following description refers to the openGauss database.

#### Adding User/Role

Follow the steps 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 Option](#)

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 Option

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.

## 6.10.2 Working with User/Role

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 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 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 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

## 6.10.3 Revoke User/Role

Follow the steps to revoke user/role:

**Step 1** Right-click the selected user/role and select **Properties**

The user/role Properties is displayed in a new **SQL Terminal** tab

**Step 2** Click **Membership** in the Properties. And double-click **Belong to**, Select or deselect the user and click **Save Changes** button .

The **Update User/Role Properties** tab page is displayed.

**Step 3** Click **Yes**

The user/role properties is changed successfully.



- To cancel the operation, click **Cancel Change** Button .

----End

## 6.11 SQL Terminal

### 6.11.1 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 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 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.

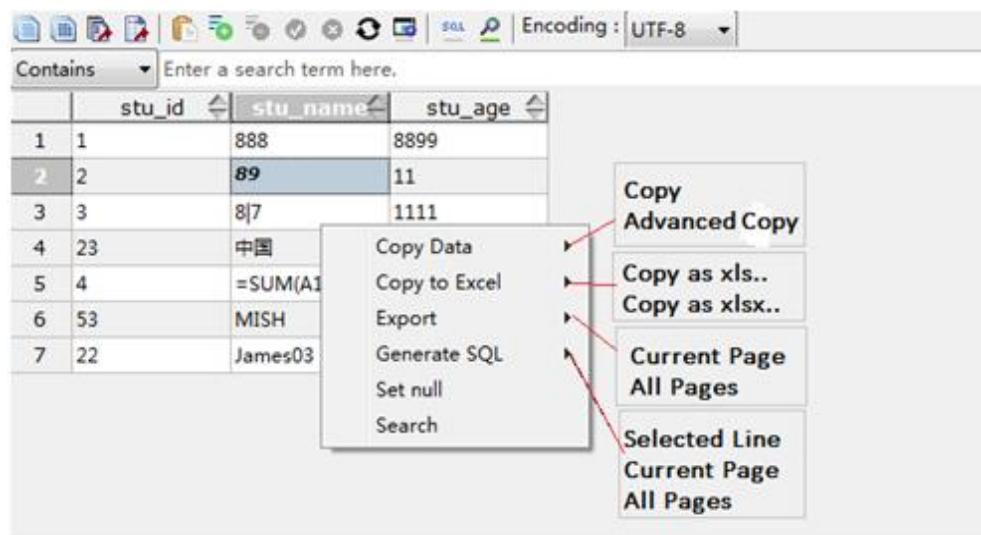
The new **SQL Terminal** tab is named as `<database_name>@<connection_profile>(<tab_number>)`. For example, `postgres@IDG_1(2)`. The tab number is updated for each new **SQL Terminal** tab of the connection profile.

----End

## Managing Right Click Option On Result Window, Edit Table Data, View Table Data, View Data

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 follows:



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 menu is as follows:

**Table 6-14** 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.
Generate SQL	Selected Line	Generate the SQL file of the selected data in logical insert

Menu Option	Sub Menu	Description
		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.  Note: When Data Studio Encoding Type is set to GBK, this button is unavailable.
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.

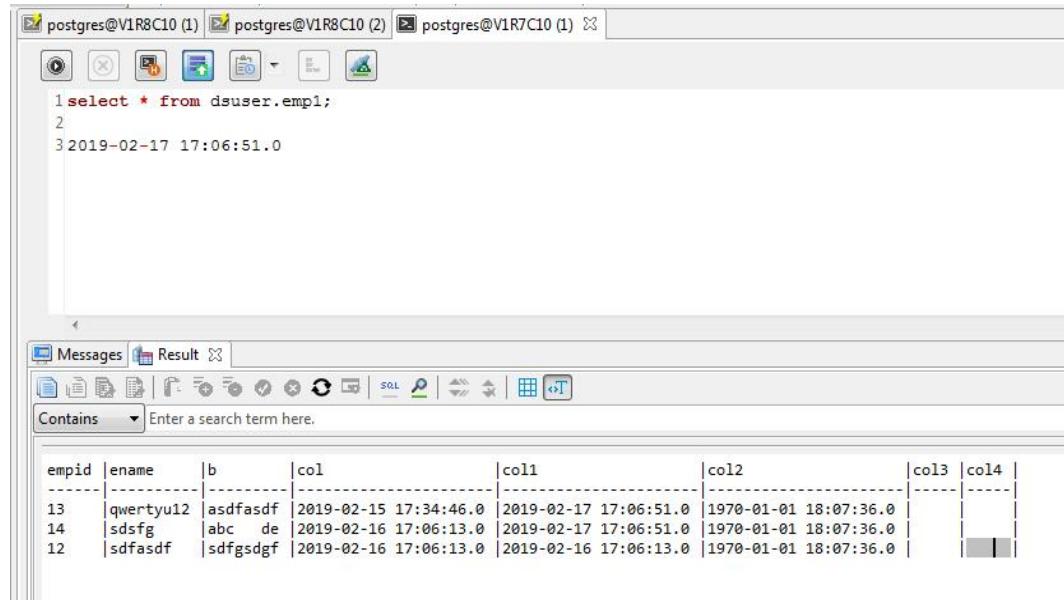
**Step 1** Click **Settings** on Menu bar of Data Studio.

**Step 2** Select **Preferences**.

**Step 3** Expand **Result Management** and select **Query Results**.

**Step 4** Under **Result Data Text Mode** option check **Include result data TextMode**.

**Step 5** Click  icon to obtain the result in text mode.



The screenshot shows the Data Studio interface. At the top, there are three tabs labeled 'postgres@V1R8C10 (1)', 'postgres@V1R8C10 (2)', and 'postgres@V1R7C10 (1)'. Below the tabs is a toolbar with various icons. The main area contains a terminal window with the following text:

```
1 select * from dsuser.emp1;
2
3 2019-02-17 17:06:51.0
```

Below the terminal is a results grid titled 'Result'. It has a search bar at the top labeled 'Contains' with the placeholder 'Enter a search term here.' The grid displays the following data:

empid	ename	b	col	col1	col2	col3	col4
13	qwertyu12	asdfasdf	2019-02-15 17:34:46.0	2019-02-17 17:06:51.0	1970-01-01 18:07:36.0		
14	sdsfg	abc	de	2019-02-16 17:06:13.0	2019-02-17 17:06:51.0	1970-01-01 18:07:36.0	
12	sdfasdf	sdfgsdgf		2019-02-16 17:06:13.0	2019-02-16 17:06:13.0	1970-01-01 18:07:36.0	

----End

#### NOTE

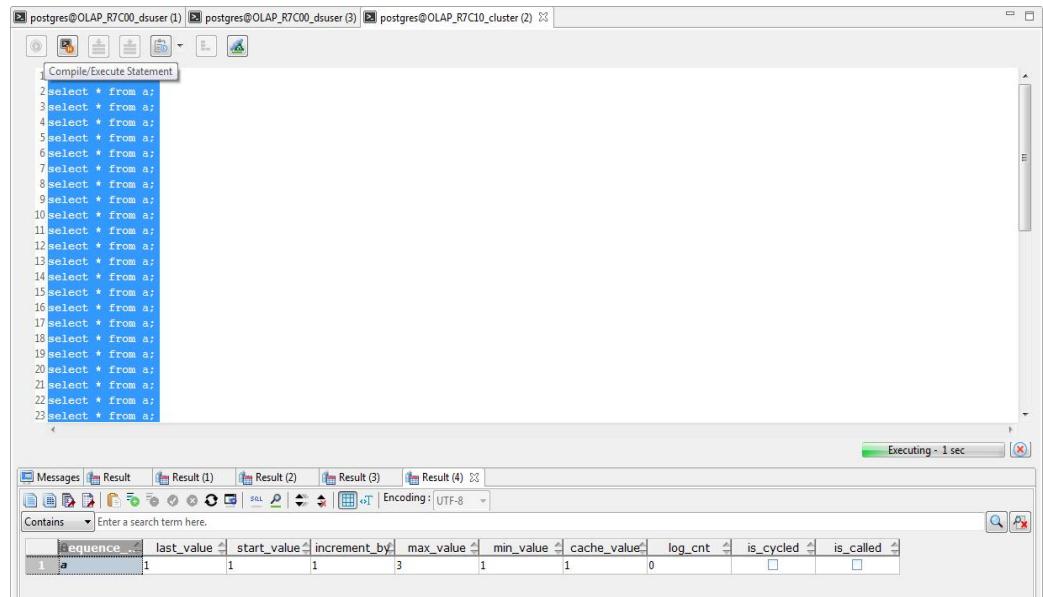
- Selecting multiple cell data and searching may show some incorrect results in text mode as all the information is copied in plain text to the search window.
- In case of Chinese data, width alignment may not be correct due to difference in character width as compared to English.

## 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 following image:



#### NOTE

- The **Cancel** button now has been removed from the toolbar.
- Execution Progress Bar is also shown in compiling 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.

## 6.11.2 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 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 now 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 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 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.

## Loading More Records

Regarding to load more data of result tab, you have to scroll down to bottom in order to load more data, which is inconvenient in some use cases. Currently, DS supports a load more record button which makes easier to trigger load more data action.

Follow the steps to load more records



**Step 1** Select the required queries and click .

**Step 2** List all the required records.

----End



**Load More Record** button is supported for

- Edit Table Data of openGauss.

## Deleting an SQL query

Follow the steps 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 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.



The pinned queries appear on top once the **SQL History** window is closed and re-opened.

Follow the steps 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 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

## 6.11.3 Opening And Saving SQL Scripts

### Opening an SQL Script

Follow the steps 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

#### NOTE

- The encoding type of the SQL file must match the encoding type specified in 7.5 Environment.
- 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 to save an SQL script:

**Step 1** Perform any of the following operations:

- Choose **File > Save** from the main menu.
- Press "Ctrl + s" to save the SQL terminal content.
- Click **Save** on the toolbar or right-click the **SQL Terminal** and select **Save**.

The **Security Warning** dialog box is displayed.

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

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

 **NOTE**

- 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 (that is 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 to save an SQL script:

**Step 1** Perform any of the following operations:

- 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 **Security Warning** 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**

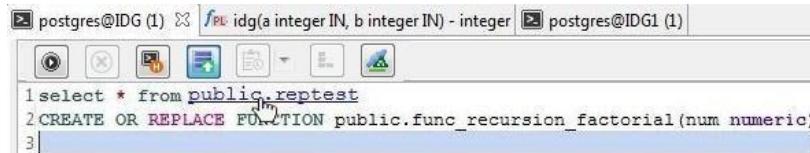
When there are unsaved changes in **File Terminals**, then user will be given an option to save or cancel on graceful exit of data studio.

## 6.11.4 Viewing Object Properties in the SQL Terminal

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

Follow the steps to view table properties:

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



```
postgres@IDG (1) [f] idg(a integer IN, b integer IN) - integer postgres@IDG1 (1)
1 select * from public.reptest
2 CREATE OR REPLACE FUNCTION public.func_recursion_factorial(num numeric)
3
```

**Step 2** Click the highlighted table name. The properties of the selected table is displayed.

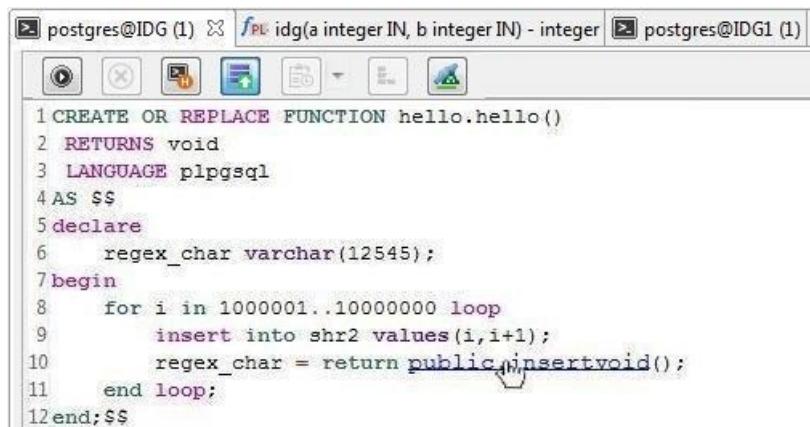
 **NOTE**

The table properties are read-only.

---End

Follow the steps to view functions/procedures:

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



```
postgres@IDG (1) [f] idg(a integer IN, b integer IN) - integer postgres@IDG1 (1)
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. The function/procedure is displayed in a new **PL/SQL Viewer** tab.

---End

Follow the steps 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 terminal before exiting the application.

Follow the steps to save the content of the terminal:

**Step 1** Click on close button of the application. **Exit Application** dialog box will appear.

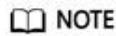
**Step 2** Click **Graceful Exit**.

**Saving File Terminal** dialog box appears.Unsaved dirty file terminal is displayed.

**Step 3** Select the terminal to save.

**Step 4** Click **OK**.

----End



Saving File Terminal dialog box will not appear in case of Force Exit.

## 6.11.5 Canceling Execution of SQL Queries

Data Studio allows you to cancel execution of an SQL query executing in the **SQL Terminal**.

Follow the steps 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.

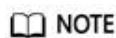
Database changes made by the canceled query are rolled back and the queries following the canceled query are not executed.

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 6.11.8 Viewing the Query Execution Plan and Cost.

The **Messages** tab shows the query cancelation message.



The **Cancel** button is enabled only during query execution.

## 6.11.6 Formatting of SQL Queries

Data Studio supports formatting and highlighting of SQL queries and PL/SQL statements.

### PL/SQL Formatting

Follow the steps 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.

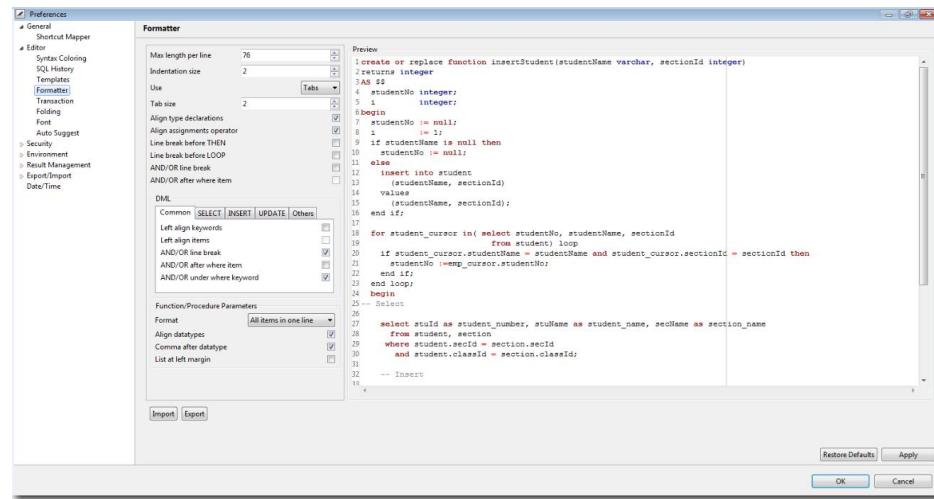
Alternatively, use the key combination **Ctrl+Shift+F** or choose **Edit > Format** from the main menu.

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. It also supports Create, Drop, Truncate. Now, you are able to configure the rules in the preferences section for the formatter. Also you can Export/Import the rules to/from Disk.



### NOTE

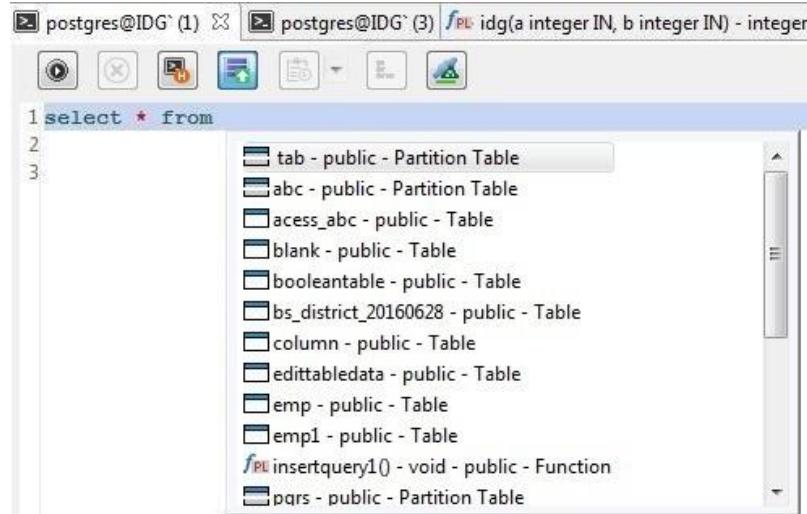
The other tab pages contain rules that apply to specific aspects of PL/SQL code. These rules are self-explanatory in the preview pane.

## 6.11.7 Selecting a DB Object in the SQL Terminal

Data Studio suggests a list of possible schema names, table names, column names, views, synonym, and sequence in the **SQL Terminal**.

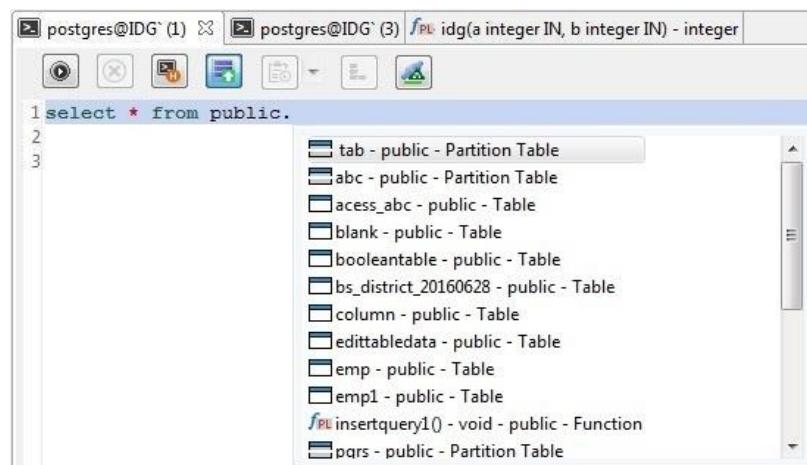
Follow the steps to select a DB object:

- 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 is appended to the parent DB object (with a period '.').

#### NOTE

- Auto-suggest also works on keywords, data types, schema names, table names, views, and table name aliases in the same way as shown above for all schema objects that you have access.

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

## 6.11.8 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 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	f <sup>∞</sup>
Hash	#
Hash Join	⋈
Nested Loop	⟳
Nested Loop Join	⟳
Modify Table	✎
Partition Iterator	🌀
Row Adapter	☰
Seq Scan on	⌚
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:

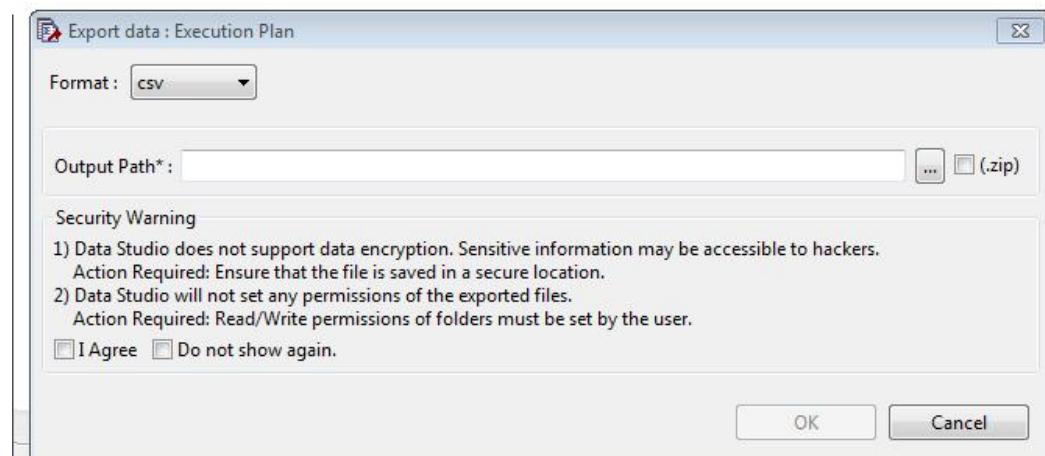
Toolbar Name	Toolbar Icon	Description
Tree Format		This icon is used view explain plan in tree format.
Text Format		This icon is used view explain plan in text format.
Copy		This icon is used to copy selected content from result window to clipboard. Shortcut key - <b>Ctrl+C</b> .
Export Execution Plan		This icon is used to save the explain plan in text, Excel(xlsx), and Excel(xls) 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.

On clicking the image of **Export Execution Plan** button, Data Studio should export the whole data present in the **Execution Plan** tab. This option is available in both Text and tree format of data.



Support of formats are as follows:

1. Excel(xlsx)
2. Excel(xls)
3. Text

----End

## 6.11.9 Viewing the Query Execution Plan and Cost Graphically

Visual Explain plan displays a graphical representation of the sql query using information from the extended JSON format. This helps to refine query to enhance query and server performance. It helps to analyze the query path taken by the database and identifies heaviest, costliest and slowest node.

The graphical 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).

You can open multiple result tabs when you run multiple queries for visual explain plan. Initially it used to load the first tab depends upon selecting other tab, data correspond to particular tab can render. Each tab contains the Visual Explain Plan Diagram, reset button, corresponding query in the tool bar, properties (General properties / Specific properties) correspond to particular query.

**Costliest** - Highest **Self Cost** plan node.

**Heaviest** - Maximum number of rows output by a plan node is considered heaviest node.

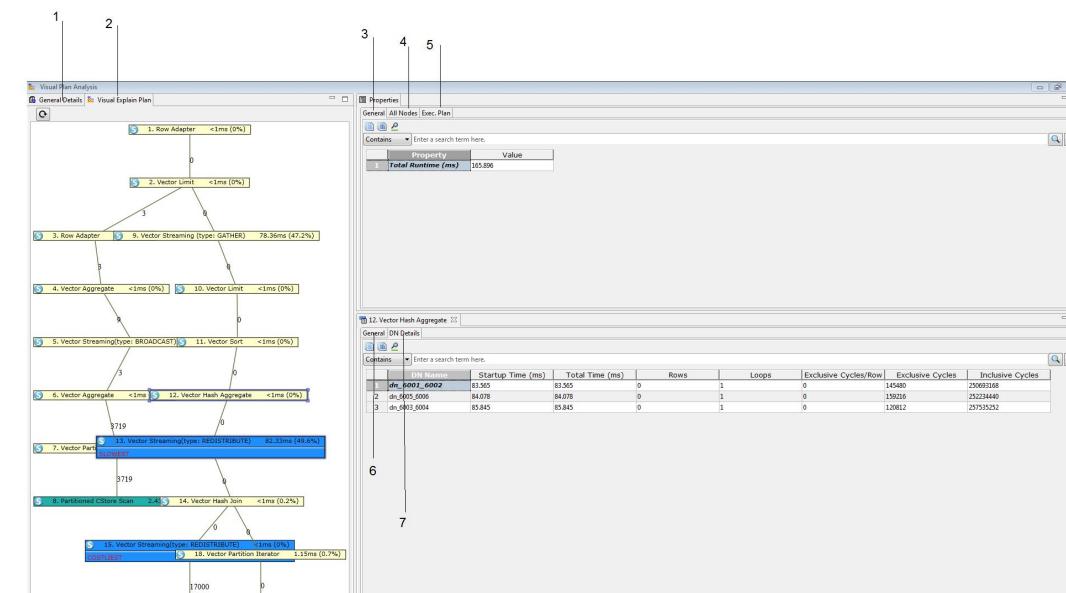
**Slowest** - Highest execution time by a plan node.

Follow the steps to view the graphical representation of 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. Alternatively, press **ALT+CTRL+X** together.

**Visual Plan Analysis** window is displayed.

Refer to 6.11.8 Viewing the Query Execution Plan and Cost section for information on reconnect option in case connection is lost while retrieving the execution plan and cost.



- 1 - **General Detail** tab - This tab displays the query.
- 2 - **Visual Explain Plan** tab - This tab displays a graphical representation of all nodes such as execution time, costliest, heaviest, and slowest node. Click each node to view the node details.
- 3 - **Properties - General** tab - Provides the execution time of the query in ms.
- 4 - **Properties - All Nodes** tab - Provides all node information.
- 5 - **Properties - Exec. Plan** tab - Provides the execution information of all nodes.
- 6 - **Plan Node - General** tab - Provides the node information for each node.
- 7 - **Plan Node - DN Details** tab - Provides detailed data node information for each node. DN Details are available only if data is collected from data node.  
Refer to 6.6.7.7 Viewing Table Data section for description on copy and search toolbar options.

----End

### 6.11.10 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/Un-indent 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

- Server will open a transaction for all the SQL statements. (For Example: select statement, explain select statement, set parameter)

#### 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 database 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 to execute function/procedure(s) or SQL queries:

1. 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.
2. Alternatively, you can right-click in the **SQL Terminal** tab and select **Execute Statement**.



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.

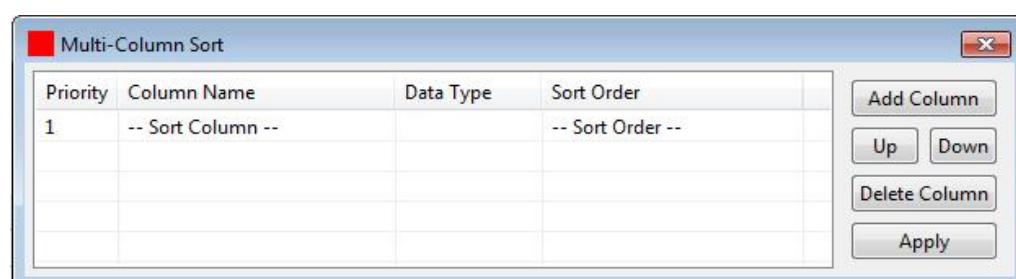
The feature is available for the following pages:

- Result Set Tab
- Edit Table Data Window
- View Table Data Window
- Batch Drop Result Window

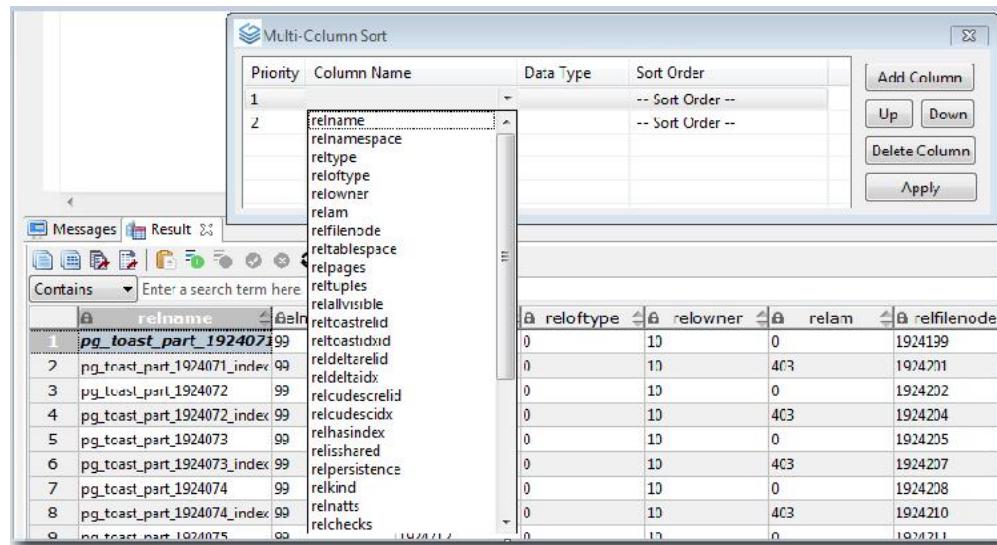
Follow the step to access Multi-column sort:

**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 {sortAscending, sortDescending}	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.
Down	Button	Moves selected column down by 1 step, this

Attribute Name	UI Element Type	Description/Action
		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, XML, SERIAL, 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</b> header
	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</b> header
	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</b> header

Icons for the sort priority are as follows:



: 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		<p>This icon is used to export all data either in excel (xlsx/xls), Text or Binary format. Refer to 6.6.7.4 Exporting Table Data.</p> <p><b>NOTE</b></p> <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) .
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 6.6.7.8 Editing Table Data section for more information.
Rollback		This button is used to roll back the changes made to the result set. Refer to 6.6.7.8 Editing Table Data section for more information.
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,

Toolbar Name	Toolbar Icon	Description
		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 6.6.7.8 Editing Table Data 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	 Encoding :   UTF-8 ▾	<p>This field will be available based on the <b>Preference &gt; Result Management &gt; Query Result &gt; Result Advanced Copy</b> settings. This drop-down is used to select the appropriate encoding to view the data accurately. The default encoding is UTF-8.</p> <p>Refer to <a href="#">Result Data Encoding</a> section to set the encoding preference.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• Data editing except for data insertion is restricted once the default encoding is modified.</li> </ul>
Multi Sort		This button brings up multi-sort pop up.
Clear Sort		This button is used to reset all the sorted column.
Text		This button is used to display texts.
Load More Records		This button is to fetch more records.

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.

Option	Description
Close Tabs to the Right	Closes only the right 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 a good 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 setRefer to [Query Results](#) for more information
  - If any column of resultset tab has Lock Image icon in it, then values are not editable.

## 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\<user name>\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 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 the **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 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 to comment/uncomment lines in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to comment/uncomment.

**Step 2** Choose **Edit** option. Choose **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 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** option. Choose **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/Un-indent Lines

The indent/un-indent option is used to shift lines as per the indent size defined in the **Preferences** tab.

Follow the steps 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 to un-indent lines in PL/SQL Viewer or SQL Terminal:

**Step 1** Select the lines to un-indent.

**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 un-indent. For example, if multiple lines are selected, and one of the selected line starts at position 1, then pressing **Shift+Tab** will un-indent 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 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.

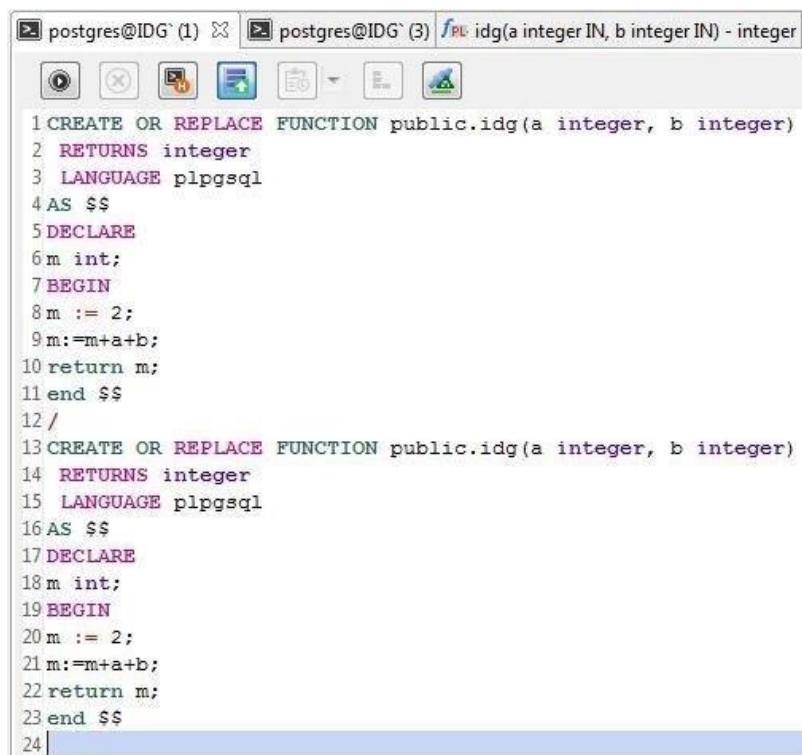
---End

## Execute Multiple Functions/Procedures or Queries

Follow the steps to execute multiple functions/procedures:

Insert **'/'** in a new line after the function/procedure in the **SQL Terminal**.

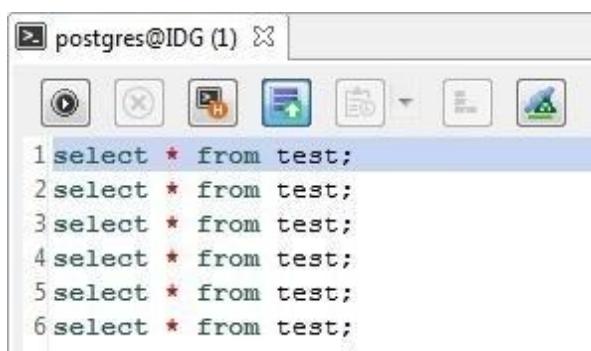
Add the new function/procedure in the next line.



```
1 CREATE OR REPLACE FUNCTION public.idg(a integer IN, b integer IN) - 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 to execute multiple SQL queries:

**Step 1** Enter multiple SQL queries in the **SQL Terminal** tab as follows:



```
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 to execute an SQL query after a function/procedure:

**Step 1** Insert '/' in a new line after the function/procedure and click  in the **SQL Terminal** tab.

**Step 2** Follow the steps to [Execute Multiple Functions/Procedures or Queries](#) on different connections.

**Step 3** In the toolbar, select the required connection from the connection profiles drop-down list and click  in the **SQL Terminal** tab.

----End

## Rename SQL Terminal

Follow the steps 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 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.

## 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.

**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.
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 is closed for all the terminals unless you click **Open SQL Assistant** button again.

----End

### 6.11.11 Exporting Query Results

You can export the results of an SQL query into a 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

Follow the steps 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.



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.



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 to export the current page:

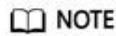
**Step 1** Select the **Result** tab.

**Step 2** Click  to export the current page.

The **Security Warning** dialog box is displayed.

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

**Step 4** Select the location to save the current page.



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.



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

## 6.11.12 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.



Use the existing SQL Terminal connection to edit temporary tables.

----End

## 6.12 Batch Operation

### 6.12.1 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
-------------	--

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.

#### NOTE

The Object Browser filter option opens a new tab, where you can enter search scope. After providing text and press **Enter** to start search. For usability improvement, a search bar is provided on the object browser component and on entering the object name of interest, the tree (if expanded) shall display only the objects that match the filter criteria.

For the nodes that are not expanded the filter rules will be applied when the node is expanded.

## 6.12.2 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	Displays the status of the drop operation. <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.
No selection	Un-selection 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 objects that are 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 menu bar. If you disconnect the database, the icons will remain disabled and will not be enabled even after reconnecting. You need to re-select the objects to drop and the selected objects will be available in the new terminal window.

----End

### 6.12.3 Exporting DDL Batch of Objects

The objects of Regular Tables, columns, and views support to batch export DDL in Data Studio.

#### Regular Tables-Batch Export DDL

Follow the steps to export DDL of Regular Tables in a batch:

**Step 1** In **Regular Tables** drop-down menu, Press **Ctrl+left-click** (select objects one by one) or **Shift+left-click** (select objects in a bunch) to select the objects to be exported DDL.

**Step 2** Right-click and select **Export DDL**.

**Step 3** The **Export DDL** dialog box is displayed. Enter the **Output Path**.

Select the option **I agree** in the safety warning, then click **OK**.

----End

#### Columns-Batch Export DDL

Follow the steps to export DDL of Columns in a batch:

**Step 1** In **Columns** drop-down menu, Press **Ctrl+left-click** (select objects one by one) or **Shift+left-click** (select objects in a bunch) to select the objects to be exported DDL.

**Step 2** Right-click and select **Export DDL**.

**Step 3** The **Export DDL** dialog box is displayed. Enter the **Output Path**.

Select the option **I agree** in the safety warning, then click **OK**.

----End

#### Views-Batch Export DDL

Follow the steps to export DDL of Views in a batch:

**Step 1** In Views drop-down menu, Press **Ctrl+left-click** (select objects one by one) or **Shift+left-click** (select objects in a bunch) to select the objects to be exported DDL.

**Step 2** Right-click and select **Export DDL**.

**Step 3** The **Export DDL** dialog box is displayed. Enter the **Output Path**.

Select the option **I agree** in the safety warning, then click **OK**.

----End

## 6.12.4 Exporting DDL and Data Batch of Objects

The objects of Regular Tables, columns, and views support to batch export DDL and Data in Data Studio.

The operation steps are similar to Export DDL in a batch, please refer to 6.12.3 Exporting DDL Batch of Objects.

## 6.12.5 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.

### 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 Security](#)
- [7.5 Environment](#)
- [7.6 Result Management](#)
- [7.7 Export/Import](#)

## 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 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.

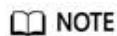
**Step 3** Select the required shortcut key and click **Modify**.

**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.



- Multiple shortcut keys can be modified before restarting Data Studio.

**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 to remove the shortcut keys:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

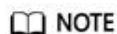
**Step 2** Choose **General > Shortcut Mapper**.

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.



- 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 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 5.6 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
Sorts the result sets of visual charts, edit tables, and queries in ascending, descending, or server receiving order	Alt+Click
Help menu	Alt+H
Save the SQL script	Ctrl+S
Edit menu	Alt+E
Compiling/Executing SQL Terminal Statements	Ctrl+Enter
Search and Replace	Ctrl+F
Search for the previous one	Shift+F3
Search for the next one	F3
Redoing	Ctrl+Y
On the Edit Table Data tab page, copy Execution Time and Status	Ctrl+Shift+K
Copy the database object from the automatic recommendation list	Alt+U
Open the SQL script	Ctrl+O
Comment out or cancel the comment line	Ctrl+/
Locate the first element in the Object Browser	Alt+Page Up or Alt+Home
Locate the last element in the Object Browser	Alt+Page Down or Alt+End
Locate to row	Ctrl+G
Disconnect the connection	Ctrl+Shift+D
Formatting (SQL and PL/SQL)	Ctrl+Shift+F
Change the value to uppercase	Ctrl+Shift+U
Change the value to lowercase	Ctrl+Shift+L
Updates the cells or columns in the Edit Table Data, Properties, and Results windows. Click the cell or column header to enable this option	F2

Function	Shortcut Key
Close the PL/SQL Viewer tab page, Table Data View tab page, Execute Query tab page, or Properties tab page	Shift+F4
Shearing	Ctrl+X
Copy Object Browser or the name of the object modified in the terminal. Copy the selected data from the Terminal, Result, Table Data, or Edit Table Data tab page	Ctrl+C
Copy the data on the Result, Table Data, or Edit Table Data tab page. The data contains/does not contain the column title and row number	Ctrl+Shift+C
Copy the query result on the Edit Table Data tab page	Ctrl+Alt+C
Visualized interpretation plan	Alt+Ctrl+X
Online help (displaying the user manual)	F1
Template	Alt+Ctrl+Space
Switch to the first SQL Terminal tab page	Alt+S
Select All	Ctrl+A
Setting menu	Alt+G
Refresh (in the Object Browser area)	F5
Search Object	Ctrl+Shift+S
Highlight Object Browser	Alt+X
File menu	Alt+F
Creating a connection	Ctrl+N
Running menu	Alt+R
Switch between the SQL Terminal tab page	Ctrl+Page Up or Ctrl+Page Down
Expand/Collapse All Objects	Ctrl+M
Pastes	Ctrl+V
Collapsible object browsing navigation tree	Alt+Q
Execute	Ctrl+E
Execution plan and expense	Ctrl+Shift+X
Stop the query in the running state	Shift+Esc
Comment/Cancel the comment line or the entire segment	Ctrl+Shift+/-

Function	Shortcut Key
List of automatically recommended database objects	Ctrl+Space

## Setting Filter Timeout

You can set timeout period for filtering object browser. Object browser filtering can be stopped after timeout period.

Follow the steps to set timeout period for filtering object browser:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **General > Object Browser > Filter Timeout**.

The **Filter Timeout** pane is displayed.

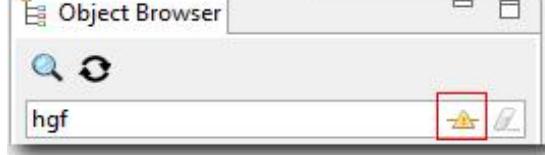
**Step 3** Set timeout period for object browser filtering. For example, 1 sec

Default value is 2 sec. Value range is from 1 to 10 sec.



### NOTE

After timeout, you can point over the following highlighted icon using computer mouse to check which server is not filtered completely.



----End

## 7.3 Editor

This section provides details on how to personalize syntax coloring, SQL history information, templates, and formatter.

### Syntax Coloring

Follow the steps 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.pref* 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.

 **NOTE**

Syntax coloring is supported till 40MB file size.

----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 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/Un-indent 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 to edit Transaction settings:

**Step 1** Choose **Settings > Preferences** from the main menu.

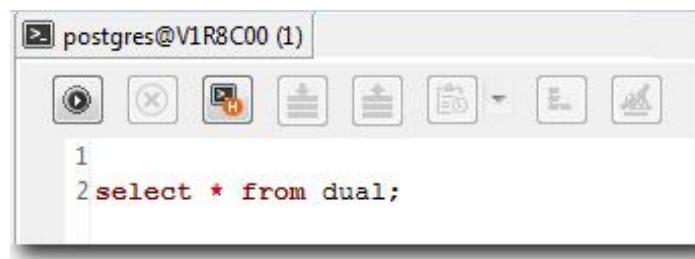
The **Preferences** dialog box is displayed.

**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.



- 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

## Folding

Follow the steps for Folding:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Folding**.

The Folding pane is displayed.

**Step 3** Select **Enable** or **Disable**. By default, **Enable** is selected.

- **Enable:** This indicates enable SQL folding feature. Supported SQL statements can be folded or unfolded.
- **Disable:** This indicates disable SQL folding feature.

 **NOTE**

Modification in settings reflects in newly opened editor. The editor which is already opened will remain with previous settings until restart.

----End

## Font

Follow the steps for Font set up:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Font**.

The Font pane is displayed.

**Step 3** Provide required font size within range from 1 to 50. By default, font size is 10.

----End

## Auto Suggest

Follow the steps for Auto Suggest:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Editor > Auto Suggest**.

The **Auto Suggest** pane is displayed.

**Step 3** In **Auto Suggest** pane, provide required number of characters in **Auto Suggest Min Character**. Default value is 2. Range of number of Auto Suggest minimum characters are within 2 to 10.

For auto suggest, sorting can be as follows:

1. Keywords
2. Data types
3. Loaded Database Objects

 **NOTE**

- Each group should be in sorted order.
- Keywords/Data types should be as per database type (openGauss)
- If database is not connected, then default keywords must be displayed.
- When you press dot (.) then only respective database objects should be displayed. Keywords/Data types should not be displayed.
- Auto suggest should be triggered by shortcuts.

----End

## 7.4 Security

This section provides details on how to personalize password and security warning display.

### Save Password Permanently

You can enable/disable to display the permanent option to save password in the connection window.

Follow the steps 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 the following table 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.

Follow the steps 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 the following table to understand the customization options available:

Option	Outcome
Yes	Selecting this option allows you to log in to Data Studio after the password has expired. <b>NOTE</b> A displayed message informs you that the password has expired and some operations may not work as expected in the following scenarios: <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	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.

**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 Warning

You can enable/disable to display the security warning for any unsecured connection/file operations.

Follow the steps to modify the display of security warning:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Security > Security Warning**.

The **Security Warning** pane is displayed.

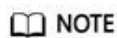
**Step 3** Select the required option. Refer the following table to understand the customization options available:

Option	Outcome
Enable	Selecting this option displays the security warning each time you try to establish an unsecure connection or perform a file operation.
Disable	Selecting this option will not display the security warning 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.



Click **Restore Defaults** from **Security Warning** pane to reset to default values. The default value is **Enable**.

----End

## 7.5 Environment

### Session Setting

Follow the steps 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, GBK and LATIN1 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**.

----End

## SQL Assistant

Follow the steps 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**.
- SQL Assistant is not supporting for Linux operating system.

----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 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.

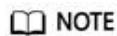
**Step 2** Choose **Environment > Session Setting**.

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 to enable/disable data encryption of saved data:

**Step 1** Choose **Settings > Preferences** from the main menu.

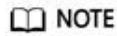
The **Preferences** dialog box is displayed.

**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**.



Click **Restore Defaults** from **Session Setting** pane to reset to default value. Encryption will be enabled by default.

----End

Follow the steps to set the size of **Import Table Data Limit/Import File Data Limit**.

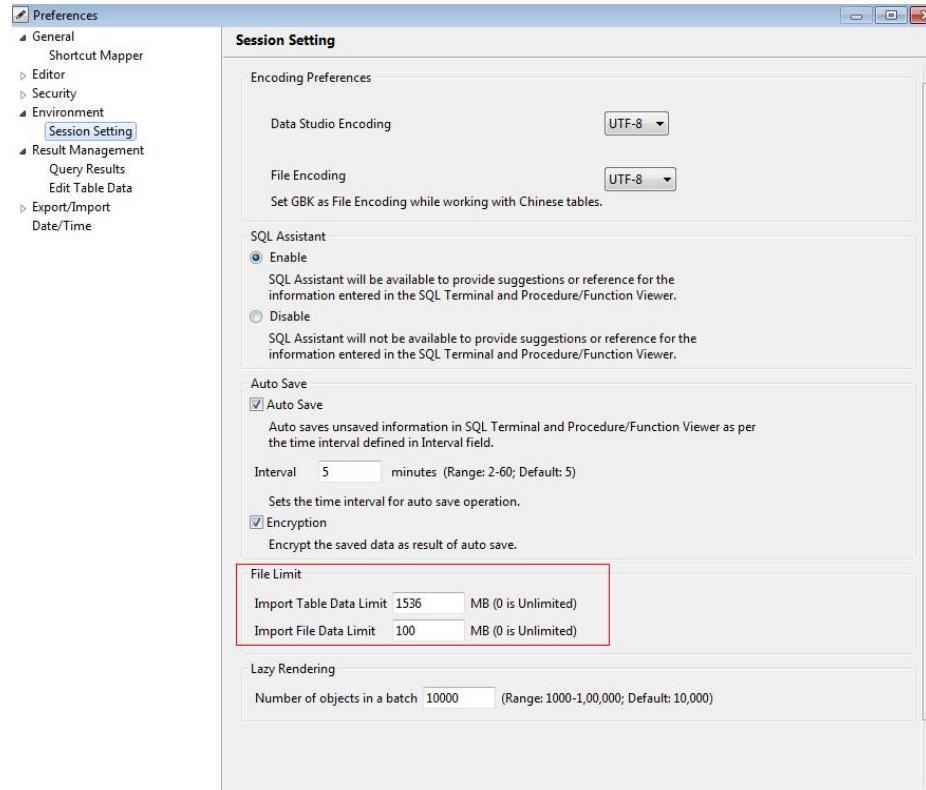
**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Environment > Session Setting**.

The **Session Setting** pane is displayed.

In File Limit section **Import Table Data Limit** and **Import File Data Limit** parameters are displayed.



**Import Table Data Limit** value defines the maximum size of the table data to be imported.

**Import File Data Limit** value defines the maximum size of the file to be imported.

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

**NOTE**

Mentioned values in the above screenshot are the default values.

---End

Follow the steps to perform rendering

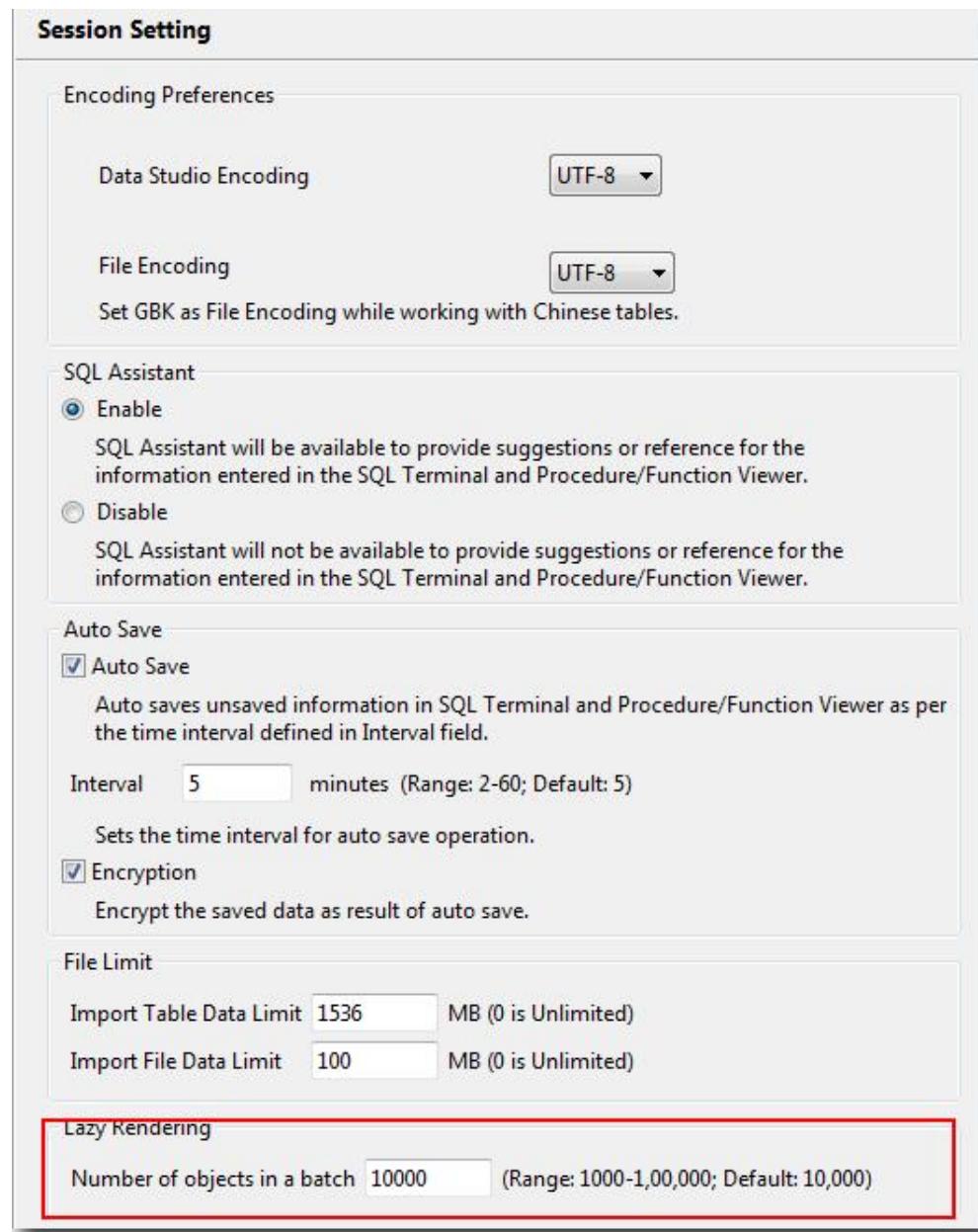
**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Environment > Session Setting**.

The **Session Setting** pane is displayed.

In Lazy Rendering section, **Number of objects in a batch** parameter is displayed.



**Step 3** Provide required number of objects in a batch, want to be rendered. Range is from 1000 to 100000. Default value is **10000**.

If you provide any value which is less than 100 or more than 1000, then **Invalid Range, (1000 - 100000)** error message is displayed.

**Step 4** Click **OK**.

----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**.

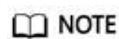
The **Query Results** pane is displayed.

**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**.



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>

Option	Outcome
	This column accepts value between 100 and 5000.

**Step 4** Click **OK**.

 **NOTE**

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 the selected content along with the row number from the query results.

**Step 4** Click **OK**.

 **NOTE**

Click **Restore Defaults** from **Query Results** pane to reset to default values. The default value is **Include column header**.

----End

Set preference to decide the behavior of opening up result set window/s:

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Go to **Result Management > Result Window**.

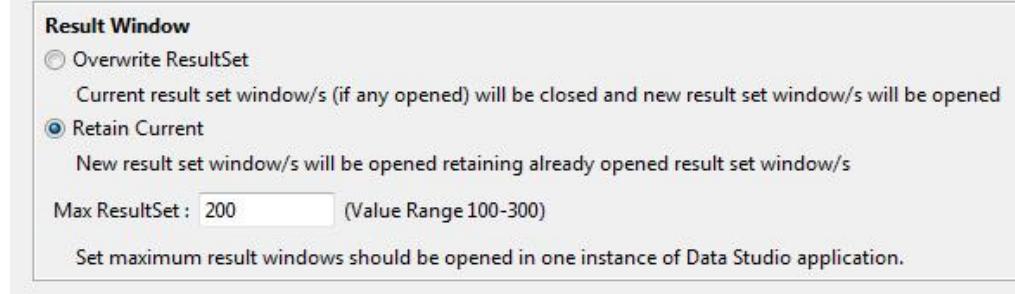
**Step 3** Select the required option.

Option	Outcome
Overwrite Resultset	Current result set opened window/s are closed and new result set window is opened.

Option	Outcome
Retain Current	New result set window/s are opened retaining already opened result set window/s.

**NOTE**

You can configure **Max Result Window** count by providing the value in **Max ResultSet**. Default value is 200 and range is from 100 to 300.



**Step 4** Click **OK**.

----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.

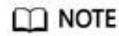
**Step 2** Choose **Result Management > Edit Table Data**.

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
openGauss	ON	ON	Save Valid Data	All the valid data will be saved and committed. Incorrect data will be omitted.
openGauss	ON	ON	Do Not Save	If an error occurs, no data will be saved.
openGauss	ON	OFF	Save Valid Data	All the valid data will be saved and committed.

Server Type	Auto Commit	Reuse Connection	Table Data Save Option	Behavior
				Incorrect data will be omitted.
openGauss	ON	OFF	Do Not Save	If an error occurs, no data will be saved.
openGauss	OFF	ON	Save Valid Data	If an error occurs, no data will be saved. Perform Commit/Rollback to proceed further.
openGauss	OFF	ON	Do Not Save	If an error occurs, no data will be saved. Perform Commit/Rollback to proceed further.

**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 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** is unselected by default.

- Edit table, view table properties and query execution must be performed again to apply the changes.

----End

## Result Data Text Mode

You can enable/disable to display the data by Text format in query results window.

Follow the steps to modify display by Text mode 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 TextMode** to include the text mode option in query results table, and it supports maximum to 30,000,000 characters of data.

**Step 4** Click **OK**.

 **NOTE**

- Click **Restore Defaults** from **Result Management** pane to reset to default values. **Include result data TextMode** is unselected by default.
- Edit table, view table properties and query execution must be performed again to apply the changes.

----End

## 7.7 Export/Import

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**.

----End

## Export Data

You can configure for whether to export data or not under **Settings > Preference > Export/Import > Export setting**.

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Export/Import > Export Data**.

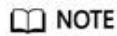
**Step 3** Enable the **Enable this option to allow Export Data** option to export data.

**Encoding** field's default value is decided from **File Encoding** preference setup under **Environment > Session Setting**.

Default value for File Encoding is **UTF-8**.

**Step 4** Click **OK**.

----End



### NOTE

Based on enable this option to allow Export Data check box, you can enable/disable following export scenarios:

- Copy Data, Copy to Excel in resultset
- Export DDL and Data option in Schema
- Export DDL and Data option in Sequence
- Export DDL and Data, Export Table Data in Table
- Export all data, Export current page data in result tab
- Right click options (export and generate sql) in result tab

## Export Timeout

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Export/Import > Export >Export Timeout**.

**Default Timeout [86400 Sec]** is selected by default.

**Step 3** Select **Custom Timeout** and provide required value in Sec.

----End

## Parallel Import/Export Limit

**Step 1** Choose **Settings > Preferences** from the main menu.

The **Preferences** dialog box is displayed.

**Step 2** Choose **Export/Import > Export > Parallel Import/Export Limit**.

**Step 3** Provide value for **Parallel Import/Export Limit** option.

Maximum number of Export/import files can be done parallely. Default value is 10. Value range is 0 to 20. If user select 0, then unlimited number of files can be exported.

Following file operations are included:

- Batch Export
- Export DDL
- Export DDL and Data
- Export Table Data
- Import Table Data
- Load sql
- Result window - Export All
- Export current
- Generate insert queries (Result window right click)

Any of these file operation can be counted.

**----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. Table 8-1 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
View	Select, Insert, Update, Delete, Truncate,	Select

Object Type	Privilege Types	Object Browser - Minimum Privilege Type
	References, All	
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.

# 9 Troubleshooting

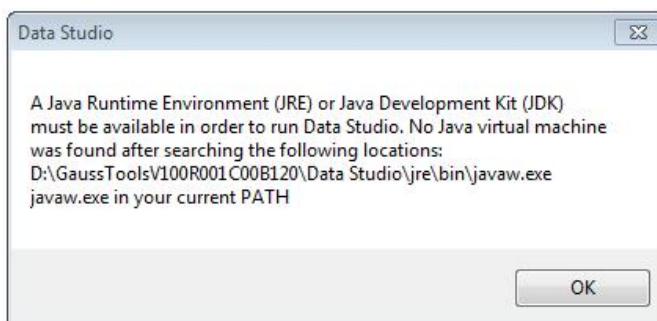
## 1. Data Studio does not open. What is the solution?

**Solution:** Check and 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 Data Studio.exe file is double clicked. What is the solution?

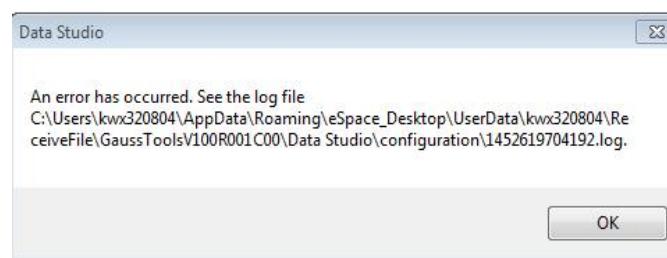
**Solution:**

### - For no JRE:



Check whether the **Java Development Kit (JDK)** version **11** 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 Development Kit (JDK)** that is installed on the system. An older version installed on the system causes this error.

Update the JDK to version **11** or above with appropriate bit number.

### - Java Incompatibility:



Check the version of **JDK** that is installed on the system. Incompatible Java bit version installed on the system causes this error. Update the JDK version to **11** 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 messages are displayed.  
What is the solution?**

**Solution:**

Message	Solution
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 11 -bit JDK (Incompatible)</li> </ul> Install Java 11 64-bit	Install Java 11 64-bit
Data Studio is supported with minimum Java Version of 11  Install Java version 11 in order to use Data Studio	Install Java version 11 with appropriate bit number

4. **Why Data Studio does 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**. The **pg\_hba.conf** and **postgresql.conf** files must be configured properly.

5. **What to do for connection issues while using Data Studio?**

**Solution:** A connection issue that may occur while using Data Studio is explained with an example:

- a. Establish a database connection.
- b. 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 is displayed and the **Object Browser** shows the status of the database.

 **NOTE**

Only the current database will be disconnected. Other databases will remain connected and re-connection is possible.

- c. 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 is the solution?**

**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 1200 MB 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**

- Based on available free memory you can enable/disable syntax color to support loading of large sql file into the terminal.
- As an example for 64-bit Data Studio and 8 GB RAM the value of the Xmx parameter must not cross 2044 MB and for 64-bit Data Studio and 8 GB RAM the value of the Xmx parameter must not cross 6000 MB. 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 is the solution?**

**Solution:** Data Studio will disconnect the connection profile. Re-establish the connection and continue execution.

**9. While exporting DDL or data, why export failed message is displayed?**

**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. 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.

**12. 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.

**13. 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.

**14. What do me do when Data Studio becomes idle and *Data Studio.log* file states "No more handles"?**

**Solution:** Restart Data Studio.

**15. 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.

**16. 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.

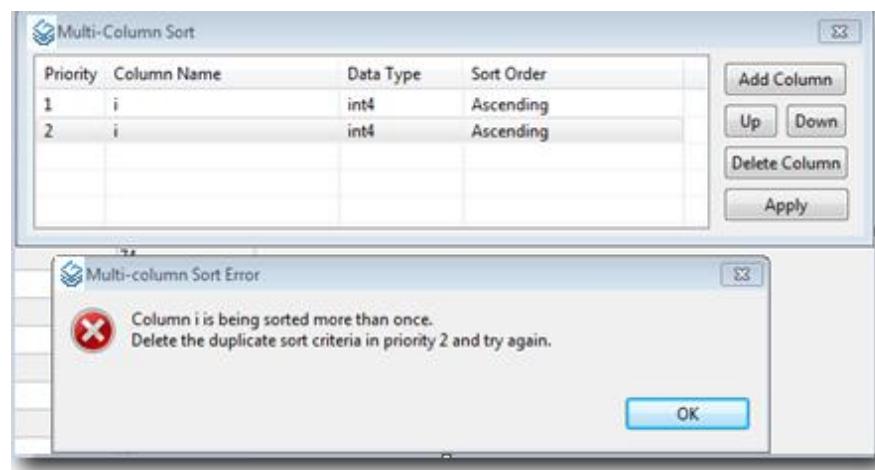
**17. Why am I not able to edit the temp table with Reuse Connection option off?**

**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 6.11.12 Managing SQL Terminal Connections section for more information.

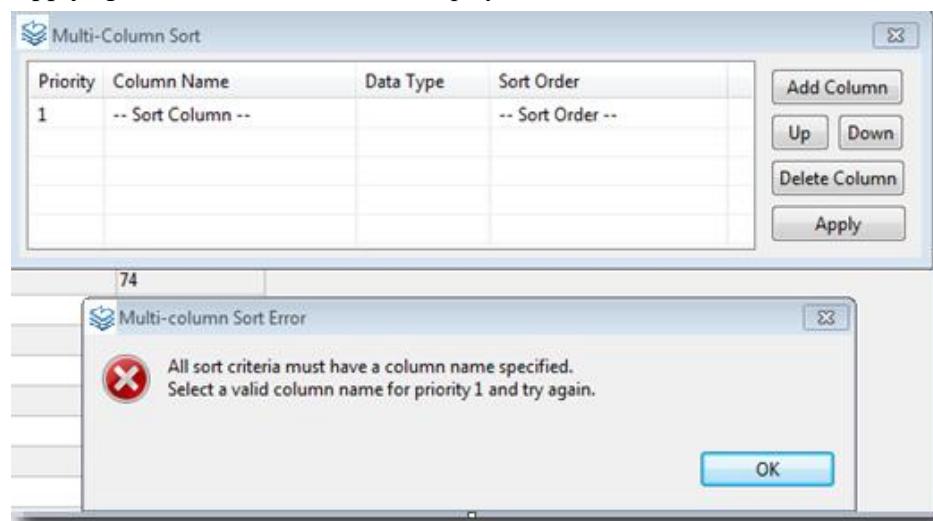
18. **What happens if same column is added more than once in Multi-Column sort pop-up?**

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.



19. **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.



20. **What happens when you click on cancel during multiple create table queries run in SQL terminal window?**

A: Canceling 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.

21. **When the user not able to log in to DS due to security keys are compromised?**

A: Follow the steps to generate the new security keys.

- Delete the config folder from **Datastudio folder > Userdata >config folder**.

- b. Restart Data Studio.
- c. New config folder is created and the keys will be regenerated.
- d. Saved password will be lost and user should re-enter the password to log in to Data Studio.

# 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](#)

## 10.1 Overview

### NOTICE

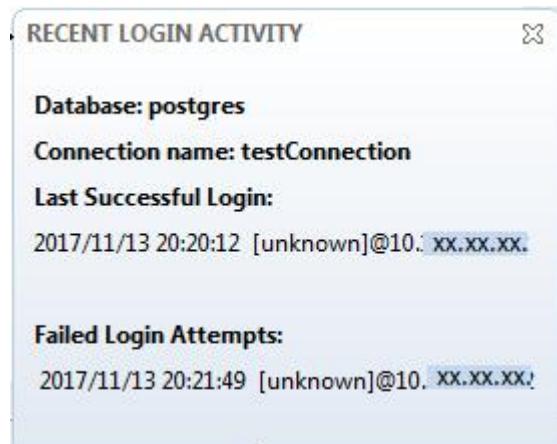
Ensure that the operating system and the required software's (refer to 3.5 System Requirements for more details) are updated with the latest patches to prevent vulnerabilities and other security issues.

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.

## 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.
- SSL connection is recommended when high security data transmission is required. Using non-ssl connection may have security risks.

Data Studio can connect to the database using the Secure Sockets Layer [SSL] option. The following files are required to 6.2.2 Adding 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** is 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/cakey.pem 2048
```

Generating RSA private key, 2048 bit long modulus.

.....+++

.....+++

e is 65537 (0x10001)

and set the protection password of the root private key. The password must contain at least four characters. Assume that the password is **Test@123**.

Enter pass phrase for **demoCA/private/cakey.pem**.

and the private key password is **Test@123**.

Verifying - Enter pass phrase for **demoCA/private/cakey.pem**.

## 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/cakey.pem -out  
demoCA/careq.pem
```

Enter pass phrase for **demoCA/private/cakey.pem**

Enter the root private key password **Test@123**.

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. The information entered when generating the server certificate and client certificate must be consistent with the following names.

```
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 can be entered as required.  
Email Address []:Please enter following'extra' attributes to be sent with your  
certificate request  
A challenge password []:  
An optional company name []:
```

#### NOTE

Name of the host where the database resides is the same as that on the server certificate.

### Step 4 Generate a self-signed root certificate.

When generating the root certificate, modify the **openssl.cnf** file. Set **basicConstraints=CA:TRUE**.

Command:

```
vi openssl.cnf
```

To generate the CA self-issued root certificate **openssl ca -config**

Command:

```
openssl ca -config openssl.cnf -out demoCA/cacert.pem -keyfile  
demoCA/private/cakey.pem -selfsign -infiles demoCA/careq.pem
```

Using configuration from **openssl.cnf**

Enter pass phrase for **demoCA/private/cakey.pem**

Enter the root private key password **Test@123**.

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
```

Until the CA root certificate is generated and signed. The root certificate is **demoCA/cacert.pem**.

**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
```

Generating a 2048 bit RSA private key

.....+++++

..+++++

e is 65537 (0x10001)

Enter pass phrase for **server.key**.

The password must contain at least four characters, for example, **Test@123**.

Enter pass phrase for **server.key**:

Confirm the protection password of the server private key again, that is, **Test@123**.

**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.

When generating the server/client certificate, modify the **openssl.cnf** file and set **basicConstraints=CA:FALSE**.

```
vi openssl.cnf
```

To no in the **index.txt.attr** file in the **demoCA** directory.

```
vi demoCA/index.txt.attr
```

Issue the generated server certificate request file. After the file is issued, a formal server certificate is generated.

```
Select y for server.crt
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 singing 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
```

Perform the following operations to disable password protection for the private key:

Remove the password protection of the server private key.

```
openssl rsa -in server.key -out server.key
```

If the password protection of the server private key is not removed, use the **gs\_guc** tool to encrypt the storage password. After the password is encrypted using

```
gs_guc encrypt -M server -K Test@123 -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 -

The method and requirements for generating the client certificate and client private key are the same as those for generating the server certificate and server 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 formal client certificate **client.crt** is generated.

```
openssl ca -config openssl.cnf -in client.req -out client.crt -days 3650 -md sha256
```



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.

Perform the following operations to disable password protection for the private key:

Remove the password of the client private key.

```
openssl rsa -in client.key -out client.key
```

If password protection for a client private key is not removed, you need to use **gs\_guc** tool to encrypt the storage password.

```
gs_guc encrypt -M client -K Test@123 -D ./
```

After the password is encrypted using **gs\_guc**, two private key password protection, files **client.key.cipher** and **client.key.rand** are generated.

To convert the client key to the DER format, perform the following steps:

```
openssl pkcs8 -topk8 -outform DER -in client.key -out client.key.pk8 -nocrypt
```

**Step 9** Generate a CRL.

To generate a CRL, perform the following steps:

Create the crlnumber file **echo'00'>./demoCA/crlnumber**

Revoke the server certificate.

```
openssl ca -config openssl.cnf -revoke server.crt
```

Generate a certificate revocation list

```
sslcrl-file.crl openssl ca -config openssl.cnf -gencrl -out sslcrl-file.crl
```

**----End**

## Replacing Certificates

By default, the security certificate and private key required by the SSL connection are configured for OpenGauss. You can replace them with your own certificate and private key.

### Prerequisites

You need to apply for formal certificates and keys of the server and client from the CA.

### Precautions

The openGauss database supports only X509v3 certificates in PEM format.

### Procedure

**Step 1** Prepare for a certificate and a key.

The naming conventions for configuration file names on the server are as follows:

- Certificate name: server.crt
- Key name: server.key
- Key password and encrypted file: server.key.cipher and server.key.rand

The naming conventions of the configuration files on the client are as follows:

- Certificate name: client.crt
- Key name: client.key
- Key password and encrypted file: client.key.cipher and client.key.rand
- Certificate name: cacert.pem
- Names of files 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 **sslcrl-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.

1. Upload the prepared package db-cert-replacement.zip to any path of a cluster user. For example: /home/gaussdba/test/db-cert-replacement.zip

2. 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**

**Step 4** Restart the openGauss database.

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

 **NOTE**

The certificate has the rollback function. You can roll back the certificate before the last certificate replacement. You can run the **gs\_om -t cert --rollback** command to remotely invoke this interface. Use **gs\_om -t cert --rollback -L** to locally invoke this interface. After the certificate is successfully replaced, roll back the certificate based on the version of the certificate to be replaced.

----End

## Server Configuration

After the SSL mode is enabled, the root certificate, server certificate, and private key must be provided.

The configuration procedure is as follows (assuming that the user certificate file is stored in the data directory **/gaussdb/data/datanode** and the default file name is used):

**Step 1** Log in to master database node as the OS user omm.

**Step 2** Generate and configure a Certificate.

Generate an SSL certificate. Copy the generated **server.crt**, **server.key**, and **cacert.pem** files to the data directory on the server.

Run the following command to query the data directory of the database node. The instance column indicates the data directory.

```
gs_om -t status --detail
```

In the Unix system, the permission setting of **server.crt** and **server.key** must prevent any external or group access. Run the following command to implement this:

```
chmod 0600 server.key
```

**Step 3** Enable the SSL authentication mode.

```
gs_guc set -D /gaussdb/data/datanode -c "ssl=on"
```

**Step 4** Set client access authentication parameters. The IP address is the IP address of the host to be connected.

```
gs_guc reload -D /gaussdb/data/datanode -h "hostssl all all 127.0.0.1/32 cert"  
gs_guc reload -D /gaussdb/data/datanode -h "hostssl all all IP/32 cert"
```

Indicates that clients in the 127.0.0.1/32 network segment are allowed to connect to the openGauss server in SSL authentication mode.

### NOTICE

If Method is set to cert in the pg\_hba.conf file on the server, only the client using the user name (common name) set in the certificate (client.crt) can connect to the database. If this parameter is set to md5 or sha256, there is no restriction on the user who connects to the database.

**Step 5** Configure digital certificate parameters related to SSL authentication.

The following information indicates that the setting is successful.

```
gs_guc set -D /gaussdb/data/datanode -c "ssl_cert_file= 'server.crt' " gs_guc set:  
ssl_cert_file= 'server.crt'  
gs_guc set -D /gaussdb/data/datanode -c "ssl_key_file= 'server.key' " gs_guc set:  
ssl_key_file= 'server.key'  
gs_guc set -D /gaussdb/data/datanode -c "ssl_ca_file= 'cacert.pem' " gs_guc set:  
ssl_ca_file= 'cacert.pem'
```

**Step 6** Restart the database.

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

**Step 7** Generate and upload a Certificate File.

---End

## Configuration For Client

**Step 1** Copy the **client.pk8**, **client.crt**, **cacert.pem** that were created above to the client machine.

### NOTE

When the DataStudio tool selects the client SSL key, the key file cannot be selected, and the \*.pk8 file needs to be selected. However, the downloaded certificate does not contain the pk8 file.

**Step 2** During login to Data Studio, password is not validated for **Two Way** SSL authentication.

The screenshot shows two configuration tabs: 'General' and 'SSL'. The 'General' tab is active, displaying fields for Database Type (openGauss), Connection Name (openGauss), Host (redacted), Host Port (24600), Database Name (postgres), User Name (dsuser), Password (empty field), Save Password (Current Session Only), and Enable SSL (unchecked). The 'SSL' tab is visible below it. The 'SSL' tab displays fields for Client SSL Certificate (C:\SSL\client.crt), Client SSL Key (C:\SSL\client.pk8, highlighted with a red box), Root Certificate (C:\SSL\cacert.pem), SSL Password (empty field), and SSL Mode (verify-ca).

General	
Database Type	openGauss
Connection Name*	openGauss
Host*	██████████
Host Port*	24600 Max Value 65535
Database Name*	postgres
User Name*	dsuser
Password*	
Save Password	Current Session Only
Enable SSL	<input type="checkbox"/>

SSL	
Client SSL Certificate	C:\SSL\client.crt
Client SSL Key	C:\SSL\client.pk8
Root Certificate	C:\SSL\cacert.pem
SSL Password	
SSL Mode	verify-ca

SSL password needs to be entered.

---End

**NOTE**

SSL password should same while generating the password.

# 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\postgres.conf** file is configured properly. Refer to the server manual for more information.
- 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 displays 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 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 > 7.5 Environment and set the encoding to GBK.

The supported combinations of Database and Data Studio encoding for export operation are shown in Table 11-1.

**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

Data Studio Encoding	Database Encoding	Compatible
	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, index, constraint, partition, function/procedure group, schemas group, and catalogs group.

**Export DDL and Data:**

Connection, database, tablespace, foreign table, sequence, index, constraint, partition, function/procedure, 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
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
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	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. 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.
Database (DB)	Database is a collection of related information, typically organized to make common retrievals easy and efficient.  Properties of a database: <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>
Database Administrator (DBA)	A database administrator (short form DBA) is a person responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization.  The role includes the development and design of database strategies, monitoring and

Term	Definition
	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.
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.
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>
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>
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.

Term	Definition
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	The object browser gives you access to all information that is relevant to PL/SQL development: <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>
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 a openGauss database, its primary function is to store data, securely and

Term	Definition
	<p>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).</p> <p>It is free and open source software, released under the terms of the PostgreSQL License, a permissive free software license.</p>
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.
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.
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

Term	Definition
	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.
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.