



**Postgres Enterprise Manager -
Developer Guide**
Release 8.5.0

EDB

Jul 14, 2022

CONTENTS

1	Using the Query tool	3
1.1	The SQL editor panel	6
1.2	The Data Output panel	7
1.2.1	Data Output tab	7
1.2.2	Explain tab	10
1.2.3	Graphical tab	10
1.2.4	Analysis tab	10
1.2.5	Statistics tab	12
1.2.6	Messages tab	12
1.2.7	Notifications tab	12
1.3	Query History panel	15
1.4	Connection status	16
1.5	Macros	16
2	Using the Schema Diff tool	21
2.1	Schema Diff Object Comparison panel	21
2.2	Schema Diff DDL Comparison panel	23
3	Using the ERD tool	27
3.1	Toolbar	27
3.2	File options	27
3.3	Export options	29
3.4	Editing options	29
3.5	Table relationship options	29
3.6	Utility Options	29
3.7	Zoom options	30
3.8	Table dialog	30
3.9	Table node	30
3.10	One-to-many link dialog box	31
3.11	Many-to-many link dialog box	32
3.12	Table link	32
3.13	Table notes	34

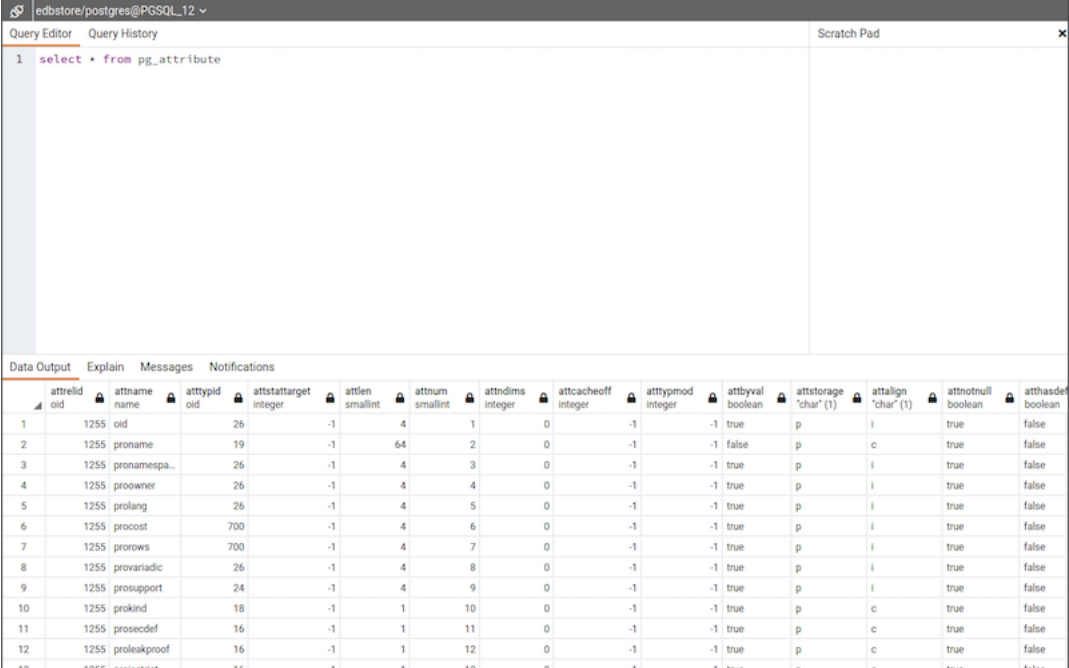
This guide describes the PEM tools for Developers. PEM consists of components that provide the management and analytical functionality for your EDB Postgres Advanced Server or PostgreSQL database. PEM is based on the Open Source pgAdmin 4 project. PEM is a comprehensive database design and management system. PEM is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance, and use of database objects.

USING THE QUERY TOOL

PEM contains a feature-rich Interactive Development Environment (IDE) that allows you to issue ad-hoc SQL queries against Postgres servers.

You can access the Query Tool via the **QueryTool** menu option on the **Tools** menu, or through the context menu of select nodes of the Browser tree control. The Query Tool allows you to:

- Issue ad-hoc SQL queries.
- Execute arbitrary SQL commands.
- Edit the result set of a SELECT query if it is updatable .
- Displays current connection and transaction status as configured by the user.
- Save the data displayed in the output panel to a CSV file.
- Review the execution plan of a SQL statement in either a text, a graphical format or a table format (similar to <https://explain.depesz.com>).
- View analytical information about a SQL statement.



The screenshot shows the Query Tool interface with a query editor at the top containing the SQL statement: `select * from pg_attribute`. Below the editor is a table of results with 13 columns and 13 rows of data. The columns represent various attributes of the pg_attribute table, and the rows show the first 13 records.

	attrelid oid	attname name	atttypid oid	attstattarget integer	attlen smallint	attnum smallint	attnums integer	attcacheoff integer	atttypmod integer	attbyval boolean	attstorage 'char' (1)	attalign 'char' (1)	attnotnull boolean	attisdef boolean
1	1255	oid	26	-1	4	1	0	-1	-1	true	p	i	true	false
2	1255	prname	19	-1	64	2	0	-1	-1	false	p	c	true	false
3	1255	prnamespa...	26	-1	4	3	0	-1	-1	true	p	i	true	false
4	1255	proowner	26	-1	4	4	0	-1	-1	true	p	i	true	false
5	1255	prolang	26	-1	4	5	0	-1	-1	true	p	i	true	false
6	1255	procost	700	-1	4	6	0	-1	-1	true	p	i	true	false
7	1255	prorows	700	-1	4	7	0	-1	-1	true	p	i	true	false
8	1255	provariadic	26	-1	4	8	0	-1	-1	true	p	i	true	false
9	1255	prosupport	24	-1	4	9	0	-1	-1	true	p	i	true	false
10	1255	prokind	18	-1	1	10	0	-1	-1	true	p	c	true	false
11	1255	prosecdef	16	-1	1	11	0	-1	-1	true	p	c	true	false
12	1255	proleakproof	16	-1	1	12	0	-1	-1	true	p	c	true	false
13	1365	proisstrict	16	-1	1	13	0	-1	-1	true	p	c	true	false

Fig. 1: Query Tool tab

You can open multiple copies of the Query tool in individual tabs simultaneously. To close a copy of the Query tool, select the **X** in the upper-right hand corner of the tab bar.

The Query Tool features two panels:

- The upper panel displays the SQL Editor. You can use the panel to enter, edit, or execute a query. It also shows the **History** tab, which you can use to view the queries that executed in the session. You can use the scratch pad to hold text snippets during editing. To reopen a closed scratch pad or open a new one, right-click the SQL Editor and select **AddPanel**. You can also open new scratch pads from other panels using the same technique.
- The lower panel displays the **DataOutput** panel. The tabbed panel displays the result set returned by a query, information about a query's execution plan, server messages related to the query's execution, and any asynchronous notifications received from the server.

TheQueryToolToolbar

The **QueryTool** toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks. If an icon is highlighted, the option is enabled; if the icon is grayed-out, the task is disabled.

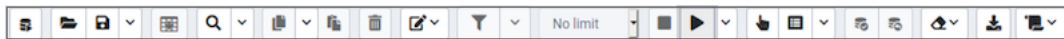


Fig. 2: Query Tool Toolbar

Hover over an icon to display a tool-tip that describes the icon's functionality:

Icon	Behavior	Shortcut
Open File	Display a previously saved query in the SQL Editor.	Accesskey + O
Save	Save a query or access the Save menu: - Select Save to save the selected content of the SQL Editor panel in a file. - Select Save As to open a new browser dialog and specify a new location to save the selected content of the SQL Editor panel.	Accesskey + S
Save Data Changes	Save the data changes (insert, update, or delete) in the Data Output Panel to the server.	F6
Find	Search, replace, or navigate the code displayed in the SQL Editor: - Select Find to provide a search target and search the SQL Editor contents. - Select Find next to locate the next occurrence of the search target. - Select Find previous to move to the last occurrence of the search target. - Select Persistent find to identify all occurrences of the search target within the editor. - Select Replace to locate and replace (with prompting) individual occurrences of the target. - Select Replace all to locate and replace all occurrences of the target within the editor. - Select Jump to navigate to the next occurrence of the search target.	Cmd+F Cmd+G Cmd+Shift+G Cm
Copy	Copy the content that's currently highlighted in the Data Output panel when in View/Edit data mode.	Accesskey + C
Paste	Paste a copied row into a new row when in View/Edit data mode.	Accesskey + P
Delete	Mark selected rows for deletion. Select Save Data Changes to update the content with your deletions.	Accesskey + D

continues on next page

Table 1 – continued from previous page

Icon	Behavior	Shortcut
Edit	Use options on the Edit menu to access text editing tools; the options operate on the text displayed in the SQL Editor panel when in Query Tool mode: - Select Indent Selection to indent the currently selected text. - Select Unindent Selection to remove indentation from the currently selected text. - Select Inline Comment Selection to enclose any lines that contain the selection in SQL style comment notation. - Select Inline Uncomment Selection to remove SQL style comment notation from the selected line. - Select Block Comment to enclose all lines that contain the selection in C style comment notation. This option acts as a toggle.	Tab Shift+Tab Cmd+ Cmd+.
Filter	Set filtering and sorting criteria for the data when in View/Edit data mode. Select the down arrow to access other filtering and sorting options: - Select Sort/Filter to open the sorting and filtering dialog box. - Select Filter by Selection to show only the rows containing the values in the selected cells. - Select Exclude by Selection to show only the rows that do not contain the values in the selected cells. - Select Remove Sort/Filter to remove any previously selected sorting or filtering options.	Accesskey + F
Limit Selector	Set a value for the maximum number of rows in a dataset.	Accesskey + R
Stop	Cancel the execution of the currently running query.	Accesskey + Q
Execute/Refresh	Execute or refresh the query highlighted in the SQL editor panel. Select the down arrow to access other execution options: - Select Auto-Rollback to instruct the server to roll back a transaction in case an error occurs during the transaction. - Select Auto-Commit to instruct the server to commit each transaction. Any changes made by the transaction are visible to others and durable in the event of a crash.	F5
Explain	View an explanation plan for the current query. The result of <i>EXPLAIN</i> is displayed graphically on the Explain tab of the output panel, and in text form on the Data Output tab.	F7

continues on next page

Table 1 – continued from previous page

Icon	Behavior	Shortcut
Explain analyze	Invoke an <i>EXPLAIN ANALYZE</i> command on the current query. Navigate through the Explain Options menu to select options for the <i>EXPLAIN</i> command: - Select Verbose to display additional information regarding the query plan. - Select Costs to include information on the estimated startup and total cost of each plan node as well as the estimated number of rows and the estimated width of each row. - Select Buffers to include information on buffer usage. - Select Timing to include information about the startup time and the amount of time spent in each node of the query. - Select Summary to include the summary information about the query plan.	
Commit	Commit the transaction.	Shift+CTRL+M
Rollback	Rollback the transaction.	Shift+CTRL+R
Clear	Use options on the Clear drop-down menu to erase display contents: - Select Clear Query Window to erase the content of the SQL Editor panel. - Select Clear History to erase the content of the History tab.	Accesskey + L
Download as CSV	Download the result set of the current query to a comma-separated list. You can specify the CSV settings through the Preferences -> SQL Editor -> CSV output dialog box.	F8
Macros	Create, edit, or clear the macros by selecting Manage Macros .	

1.1 The SQL editor panel

The **SQL editor** panel is a workspace where you can manually provide a query, copy a query from another source, or read a query from a file. The SQL editor features syntax coloring and auto-completion.

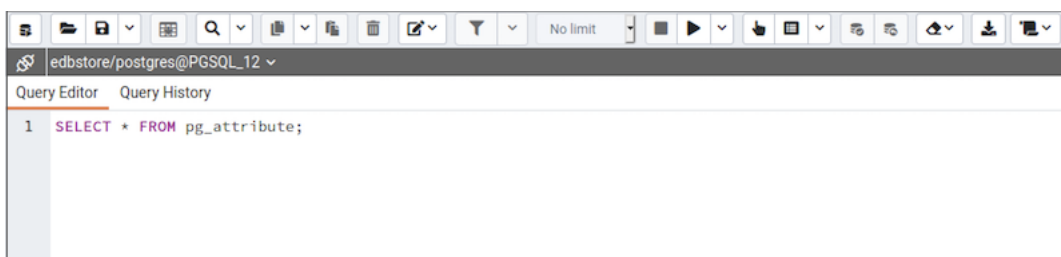


Fig. 3: Query Tool - Query Editor tab

To use auto-complete, begin typing your query; when you would like the Query editor to suggest object names or commands that might be next in your query, press the Control+Space key combination. For example, type *SELECT * FROM* (with a trailing space), and then press the Control+Space key combination to select from a popup menu of auto-complete options.

After entering a query, select **Execute/Refresh** from the toolbar. The database server receives the complete contents of the SQL editor panel for execution. To execute only a section of the code that is displayed in the SQL editor, highlight

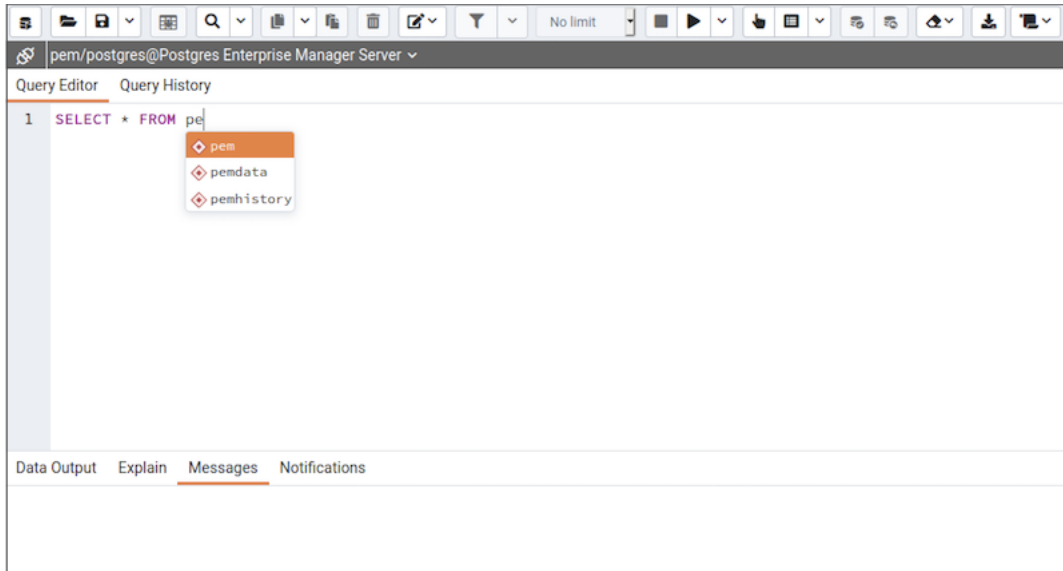


Fig. 4: Query Tool - Query Editor tab

the text that you want the server to execute, and select **Execute/Refresh**.

The message returned by the server when a command executes is displayed on the **Messages** tab. If the command is successful, the **Messages** tab displays execution details.

Options on the **Edit** menu offer functionality that helps with code formatting and commenting:

- Use the auto-indent feature to indent text to the same depth as the previous line by pressing the Return key.
- Block indent text by selecting two or more lines and pressing the Tab key.
- Implement or remove SQL style or toggle C style comment notation within your code.

You can also drag and drop certain objects from the tree-view to save time spent typing long object names. Text containing the object name are fully qualified with the schema name. Double quotes are added if required. For functions and procedures, the function name along with parameter names are pasted into the Query Tool.

1.2 The Data Output panel

The **DataOutput** panel displays data and statistics generated by the most recently executed query.

1.2.1 Data Output tab

The **DataOutput** tab displays the result set of the query in a table format. You can:

- Select and copy from the displayed result set.
- Use the **Execute/Refresh** options to retrieve query execution information and set query execution options.
- Select **Download as CSV** to download the content of the **DataOutput** tab as a comma-delimited file.
- Edit the data in the result set of a SELECT query if it is updatable.

A result set is updatable if:

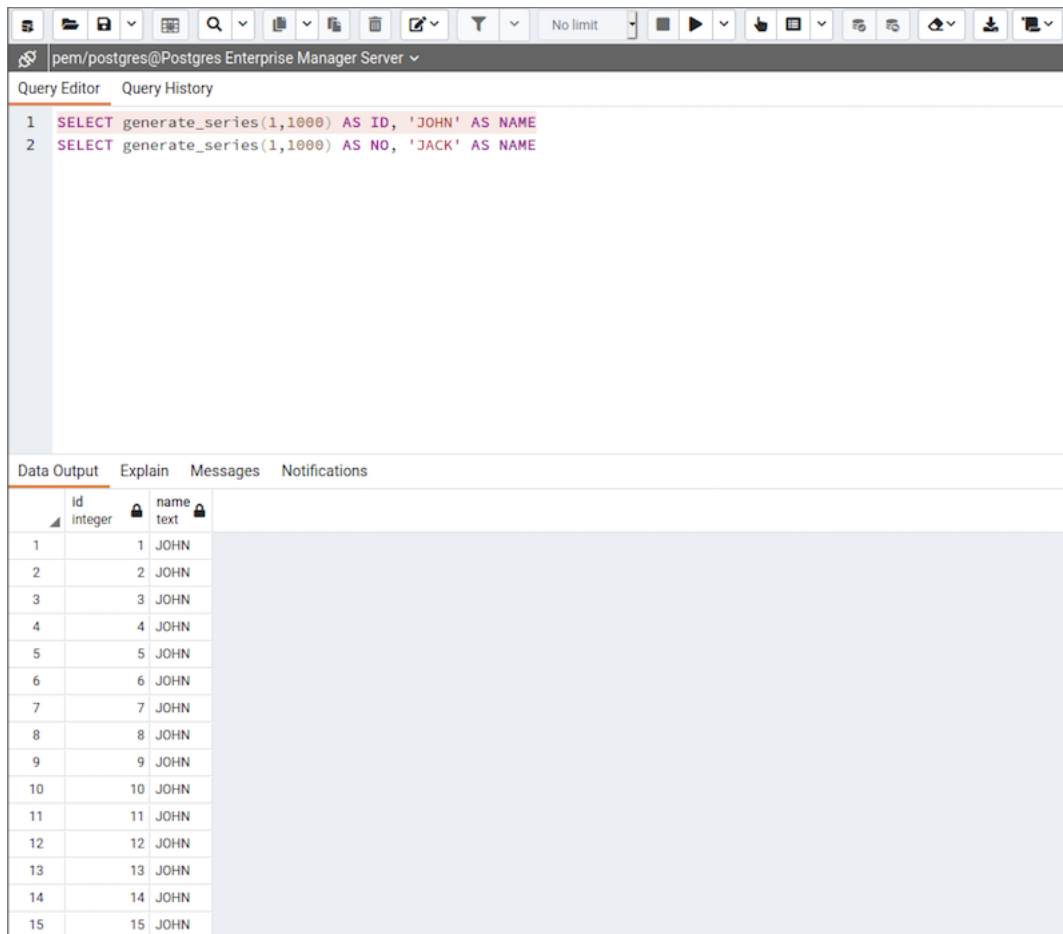


Fig. 5: Query Tool - Query Editor tab - Autocomplete feature

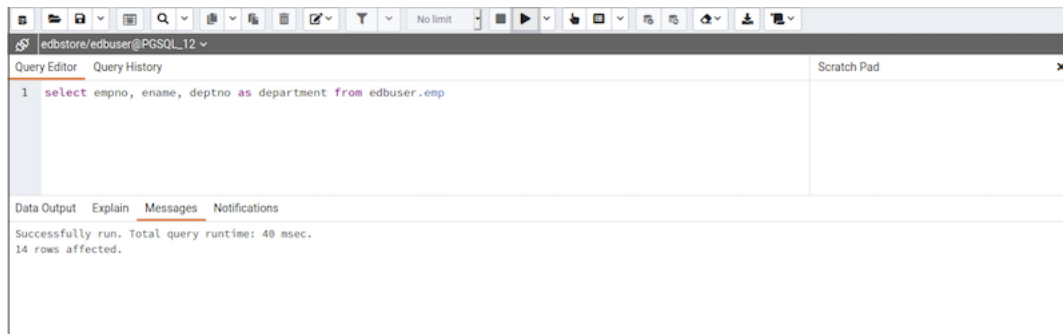
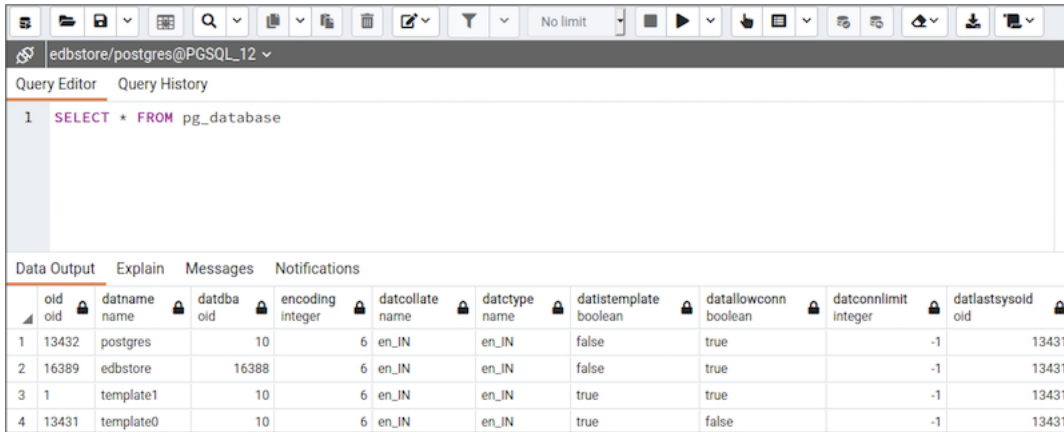


Fig. 6: Query Tool - Query Editor - Message tab



	oid	datname	datdba	encoding	datcollate	datatype	datistemplate	datallowconn	datconnlimit	datlastsysoid
1	13432	postgres	10	6	en_IN	en_IN	false	true	-1	13431
2	16389	edbstore	16388	6	en_IN	en_IN	false	true	-1	13431
3	1	template1	10	6	en_IN	en_IN	true	true	-1	13431
4	13431	template0	10	6	en_IN	en_IN	true	false	-1	13431

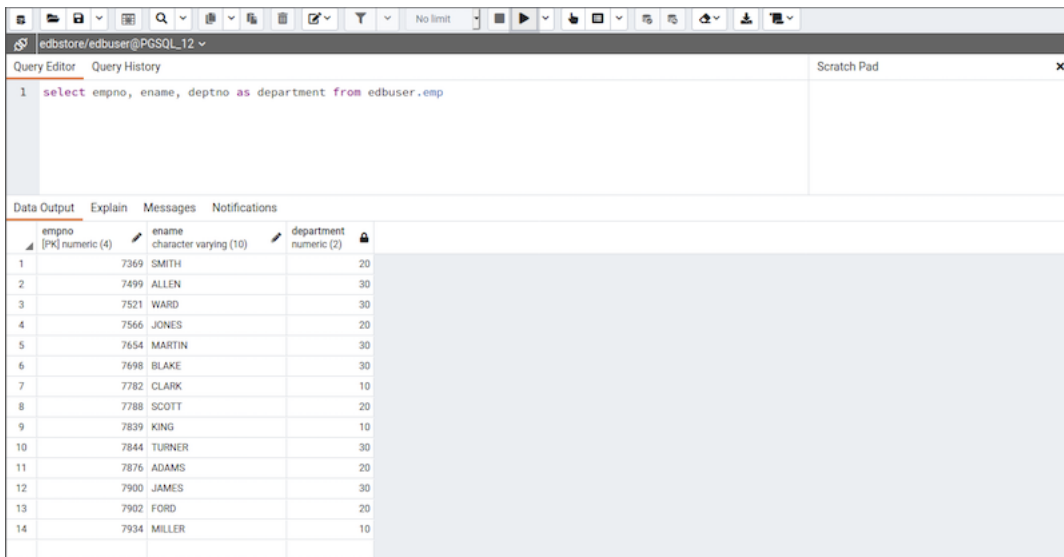
Fig. 7: Query Tool - Data output tab

- All columns are either selected directly from a single table, or they are not actually a table column (for example, the concatenation of two columns). Only columns that are selected directly from the table are editable, other columns are read-only.
- All the primary key columns or OIDs of the table are selected in the result set.

Any columns that are renamed or selected more than once are also read-only.

Note: To work with an updatable query result set, you must have psycopg2 driver version 2.8 or above installed.

Editable and read-only columns are identified using pencil and lock icons (respectively) in the column headers.



	empno	ename	department
1	7369	SMITH	20
2	7499	ALLEN	30
3	7521	WARD	30
4	7566	JONES	20
5	7654	MARTIN	30
6	7698	BLAKE	30
7	7782	CLARK	10
8	7788	SCOTT	20
9	7839	KING	10
10	7844	TURNER	30
11	7876	ADAMS	20
12	7900	JAMES	30
13	7902	FORD	20
14	7934	MILLER	10

Fig. 8: Query Tool - Editable and Read-only columns

An updatable result set is similar to the **DataGrid** in **View/EditData** mode and can be modified in the same way.

If Auto-commit is off, data changes are made as part of the ongoing transaction; if no transaction is ongoing, a new one is initiated. The data changes are not committed to the database unless the transaction is committed.

If any errors occur during saving (for example, trying to save a NULL into a column with a NOT NULL constraint), the data changes are rolled back to an automatically created SAVEPOINT to ensure any previously executed queries in the ongoing transaction are not rolled back.

All rowsets from previous queries or commands that are displayed in the **DataOutput** panel get discarded when you invoke another query; open another query tool browser tab to keep your previous results available.

1.2.2 Explain tab

To generate the **Explain** or **ExplainAnalyze** plan of a query, select **Explain** or **ExplainAnalyze** in the toolbar.

More options related to **Explain** and **ExplainAnalyze** can be selected from the drop-down menu.

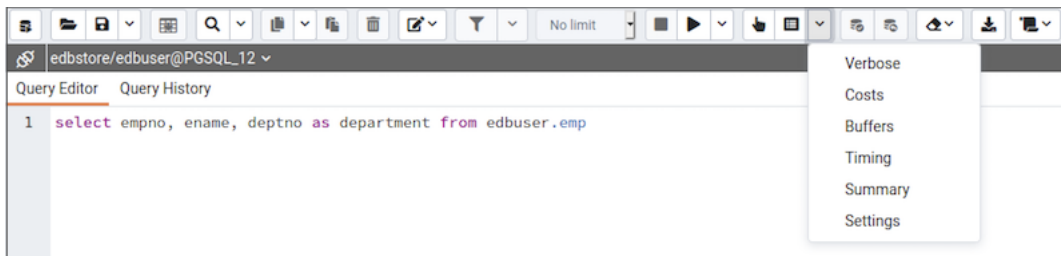


Fig. 9: Query Tool - Toolbar Explain button

Please note that PEM generates the **Explain[Analyze]** plan in JSON format.

On successful generation of **Explain** plan, three tabs/panels generate under the **Explain** panel.

1.2.3 Graphical tab

To download the plan as an SVG file, select **Download** in the top-left corner of the Explain canvas. **DownloadasSVG** isn't supported on Internet Explorer.

The query plan that accompanies **Explainanalyze** is available on the **DataOutput** tab.

1.2.4 Analysis tab

The **Analysis** tab shows the plan details in table format, generating a format similar to the one available at explain.depsez.com. Each row of the table represents the data for an Explain Plan node. The output can contain the node information, exclusive timing, inclusive timing, actual versus planned rows, actual rows, planned rows, or loops. Child rows of the selected row are marked with an orange dot.

If the percentage of the exclusive/inclusive timings of the total query time is:

Greater than 90 → Red

Greater than 50 → Orange (between red and yellow)

Greater than 10 → Yellow

If the planner has misestimated the number of rows (actual vs planned) by:

10 times → Yellow color

100 times → Orange (between Red and Yellow) color

1000 times → Red color

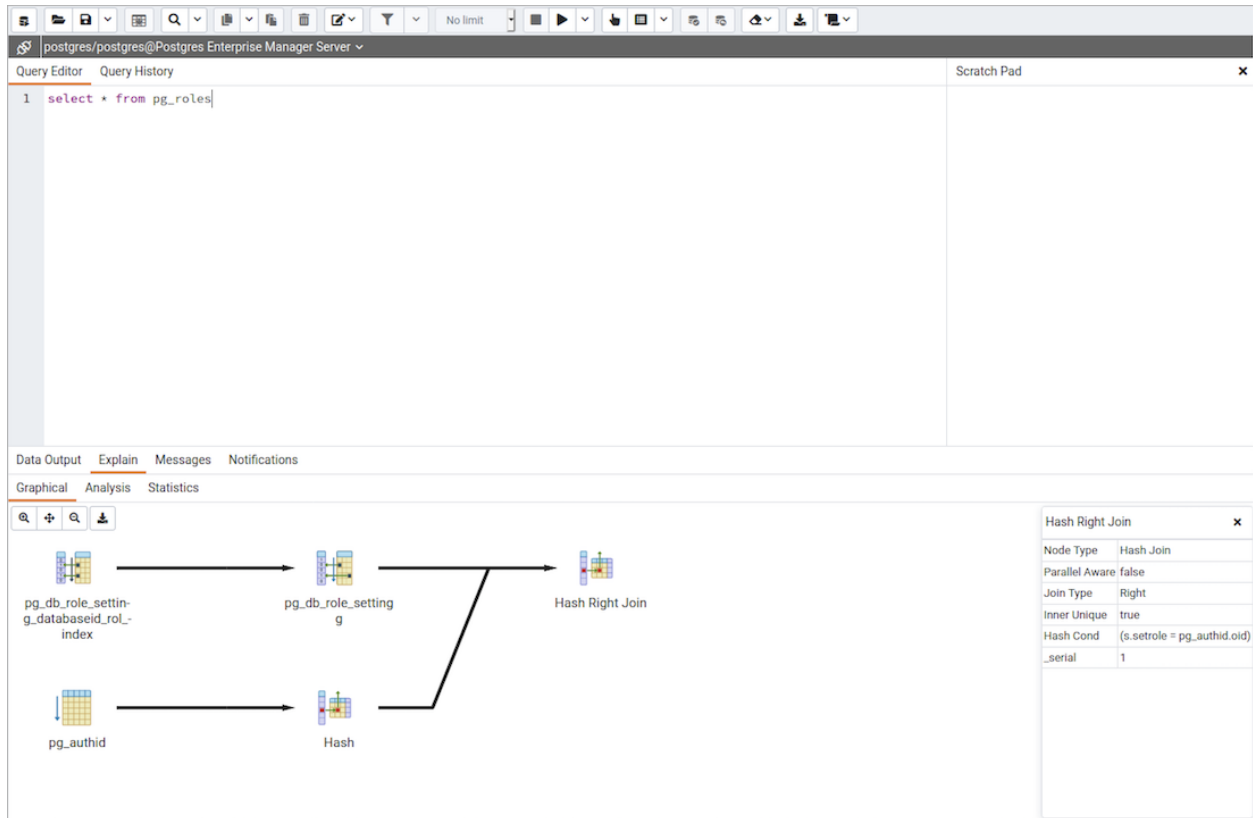


Fig. 10: Query Tool - Explain tab - Graphical plan tab

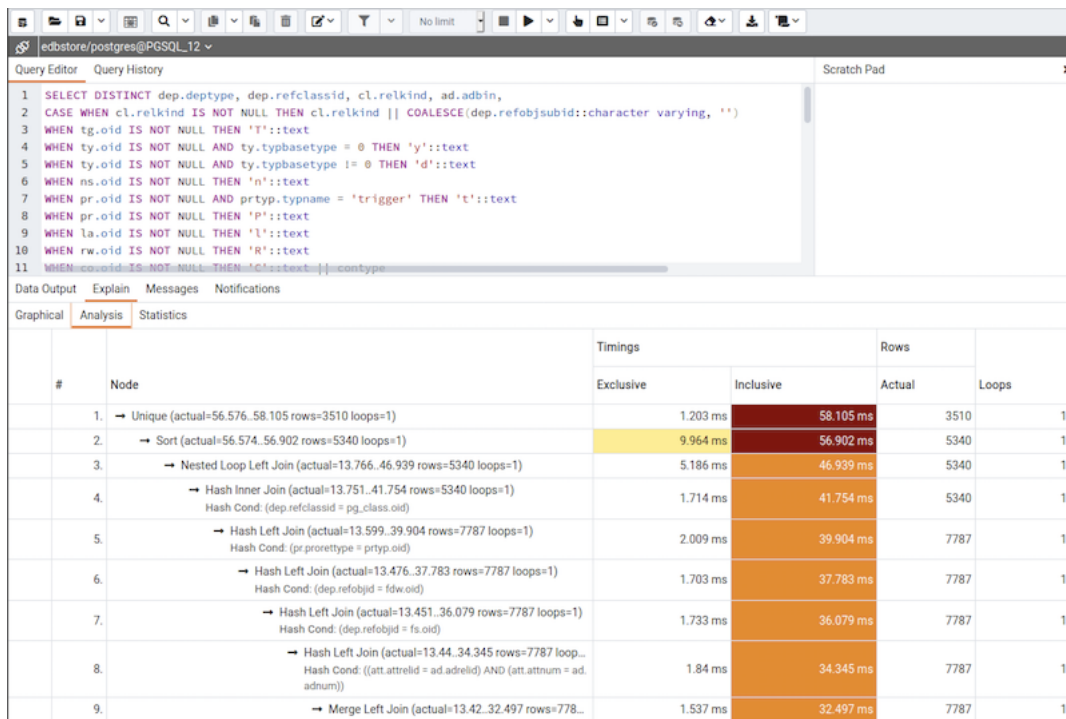


Fig. 11: Query Tool - Explain tab - Analysis tab

1.2.5 Statistics tab

The **Statistics** tab displays information in two tables:

- ****StatisticsperNodeType**** tells you how many times each node **type** was referenced.
- ****StatisticsperTable**** tells you how many times each table was referenced by the **query**.

The screenshot shows the Query Tool interface with the following components:

- Query Editor:** Contains a SQL query for explaining a query plan.
- Scratch Pad:** An empty text area for notes.
- Data Output:** Includes tabs for Explain, Messages, and Notifications.
- Statistics per Node Type:**

Node type	Count	Time spent	% of query
Hash	12	3.164 ms	5.45%
Hash Inner Join	1	1.714 ms	2.95%
Hash Left Join	10	12.938 ms	22.27%
Hash Right Join	1	0.062 ms	0.11%
Index Only Scan	1	0 ms	0%
Index Scan	3	0.065 ms	0.12%
Materialize	4	0.027 ms	0.05%
Merge Left Join	7	11.159 ms	19.21%
Nested Loop Left Join	4	12.214 ms	21.03%
Seq Scan	20	2.215 ms	3.82%
Sort	7	13.36 ms	23%
Unique	1	1.203 ms	2.08%
- Statistics per Table:**

Table name	Scan count	Total time	% of query
pg_attrdef	1	0.004 ms	0.01%
Seq Scan	1	0.004 ms	100%
pg_attribute	1	0.593 ms	1.03%
Seq Scan	1	0.593 ms	100%
pg_class	4	0.235 ms	0.41%
Index Scan	1	0.038 ms	16.18%
Seq Scan	3	0.197 ms	83.83%
pg_constraint	1	0.022 ms	0.04%
Index Scan	1	0.022 ms	100%
pg_depend	1	0.586 ms	1.01%
Seq Scan	1	0.586 ms	100%

Fig. 12: Query Tool - Explain plan tab - Statistics tab

1.2.6 Messages tab

Use the **Messages** tab to view information about the most recently executed query:

If the server returns an error, the error message appears on the **Messages** tab, and the syntax that caused the error is underlined in the SQL editor. If a query succeeds, the **Messages** tab displays how long the query took to complete and how many rows were retrieved:

1.2.7 Notifications tab

Use the **Notifications** tab to view the notifications using the PostgreSQL Listen/Notify feature. For more details, see [PostgreSQL documentation](#).

Example:

1. Execute `LISTEN "foo"` in first Query Tool session.
2. In the another Query Tool session, execute `Notify` command or `pg_notify` function to send the notification of the event together with the payload.

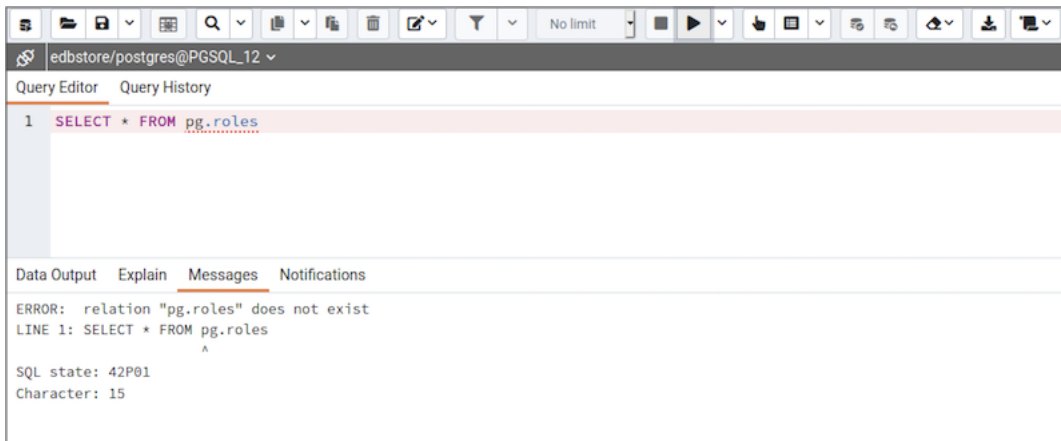


Fig. 13: Query Tool - Output error

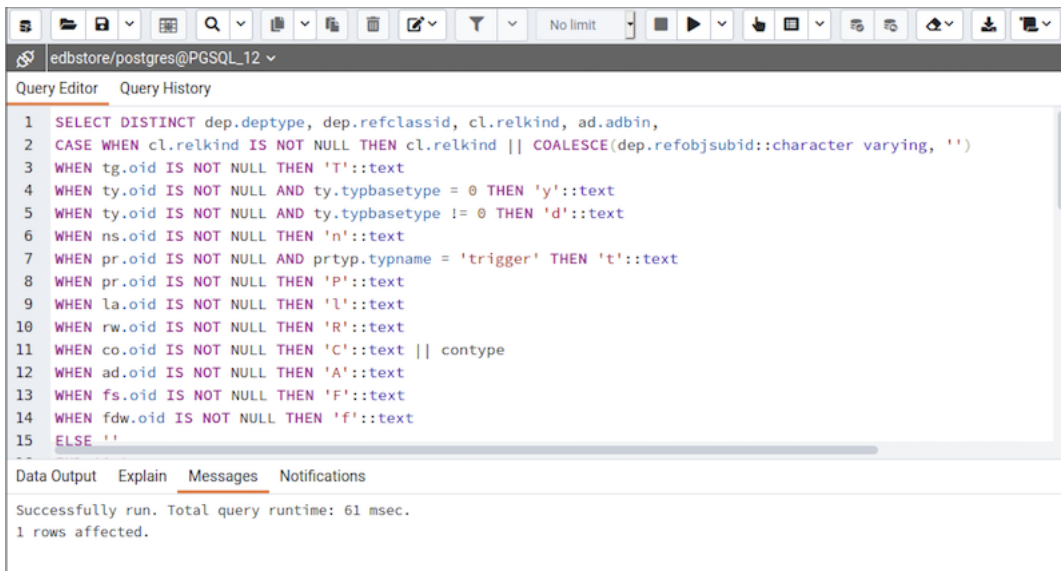


Fig. 14: Query Tool - Messages tab

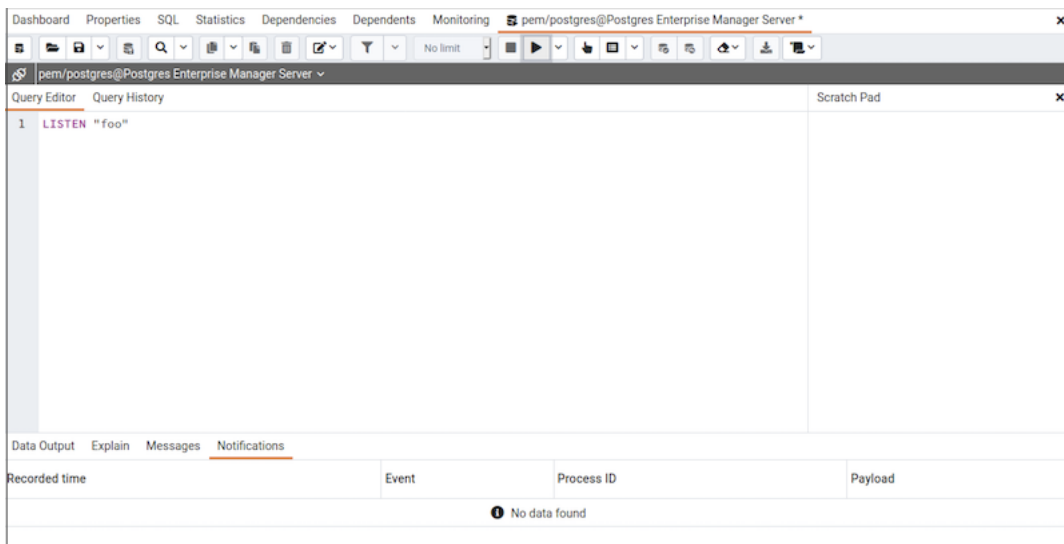


Fig. 15: Query Tool - notifications listen

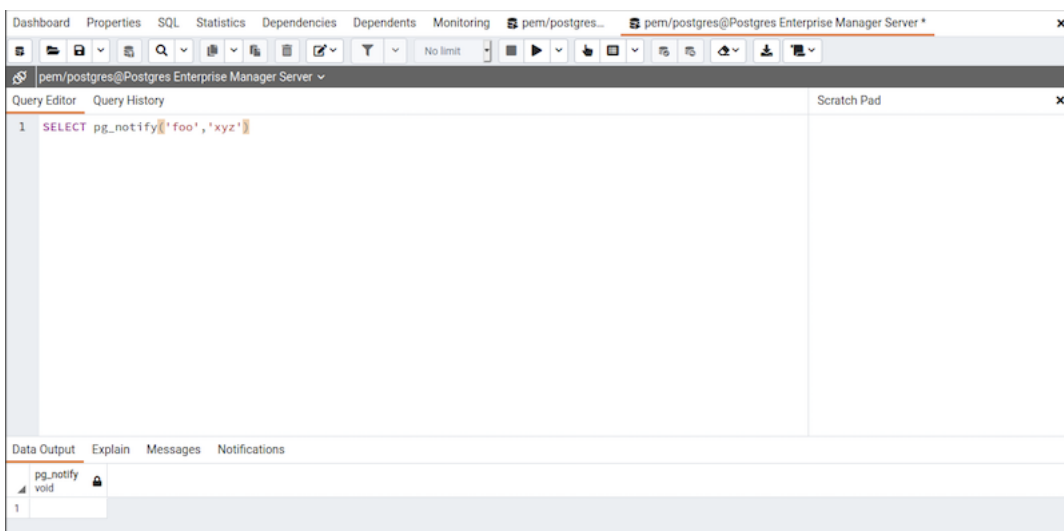
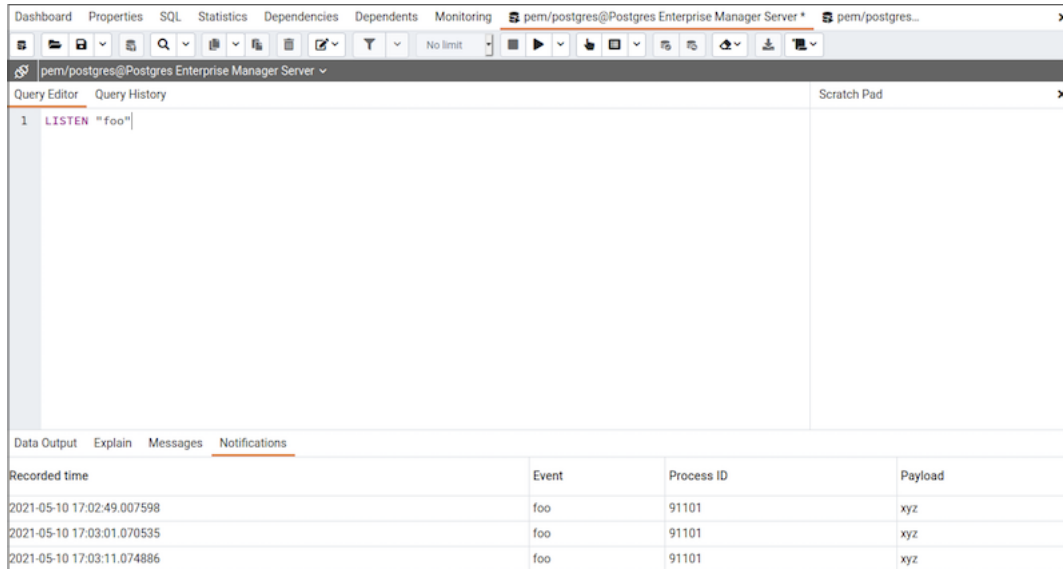


Fig. 16: Query Tool - notifications notify

- You can observe the **Notification** tab in the first Query Tool session where it shows the Recorded time, Event, Process ID, and the Payload of the particular channel.

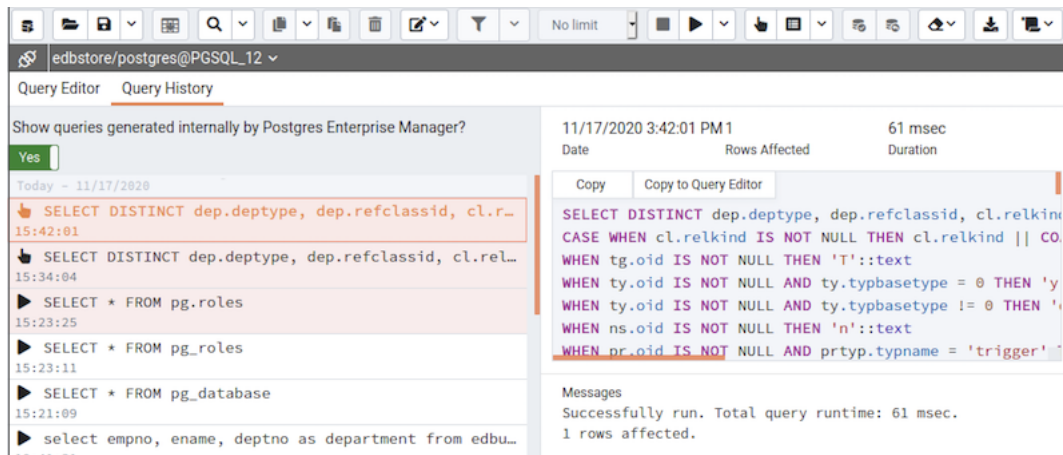


Recorded time	Event	Process ID	Payload
2021-05-10 17:02:49.007598	foo	91101	xyz
2021-05-10 17:03:01.070535	foo	91101	xyz
2021-05-10 17:03:11.074886	foo	91101	xyz

Fig. 17: Query Tool - Output Notifications panel

1.3 Query History panel

Use the **QueryHistory** tab to review activity for the current session:



Date	Rows Affected	Duration
11/17/2020 3:42:01 PM		61 msec

```

SELECT DISTINCT dep.deptype, dep.refclassid, cl.r...
15:42:01
SELECT DISTINCT dep.deptype, dep.refclassid, cl.rel...
15:34:04
SELECT * FROM pg.roles
15:23:25
SELECT * FROM pg_roles
15:23:11
SELECT * FROM pg_database
15:21:09
select empno, ename, deptno as department from edbu...

```

Messages
Successfully run. Total query runtime: 61 msec.
1 rows affected.

Fig. 18: Query Tool - Query History tab

The **QueryHistory** tab displays information about recent commands:

- The date and time that a query was invoked.
- The text of the query.
- The number of rows returned by the query.
- The amount of time it took the server to process the query and return a result set.

- Messages returned by the server (not noted on the **Messages** tab).
- The source of the query (indicated by icons corresponding to the toolbar).

You can show or hide the queries generated internally by pgAdmin (during **View/EditData** or **SaveData** operations).

To erase the content of the **QueryHistory** tab, select **Clearhistory** from the **Clear** drop-down menu.

Query history is maintained across sessions for each database on a per-user basis when running in **QueryTool** mode. In **View/EditData** mode, history is not retained. By default, the last 20 queries are stored for each database. This can be adjusted in `config_local.py` by overriding the `MAX_QUERY_HIST_STORED` value.

1.4 Connection status

Use the **Connectionstatus** feature to view the current connection and transaction status by selecting the status in Query Tool:

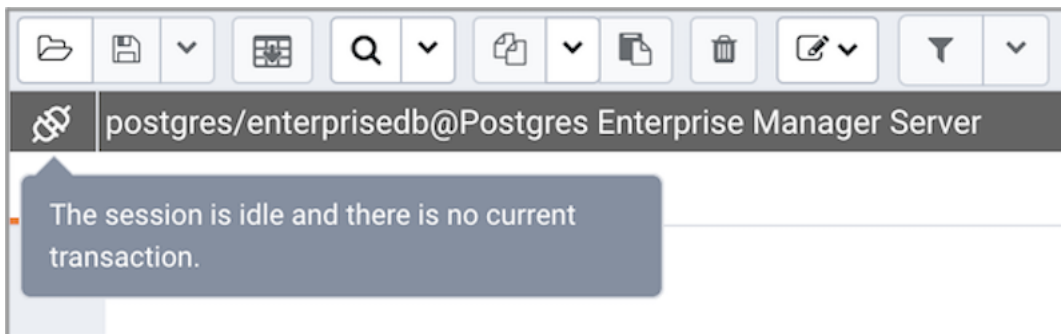


Fig. 19: Query Tool - Connection Status button

1.5 Macros

Query Tool Macros enable you to execute predefined SQL queries by pressing a single key. Predefined queries can contain the placeholder `$$SELECTION$`. When the macro executes, the placeholder is replaced with the currently selected text in the **QueryEditor** pane of the Query Tool.

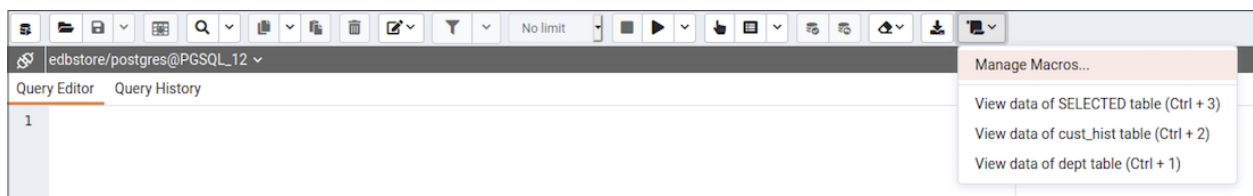


Fig. 20: Query Tool - Connection Status button

To create a macro:

1. In the Query Tool, select **Macros > ManageMacros**.
2. Select the key you want to use, enter the name of the macro and the query.
3. Optionally, include the selection placeholder.
4. Select **Save**.

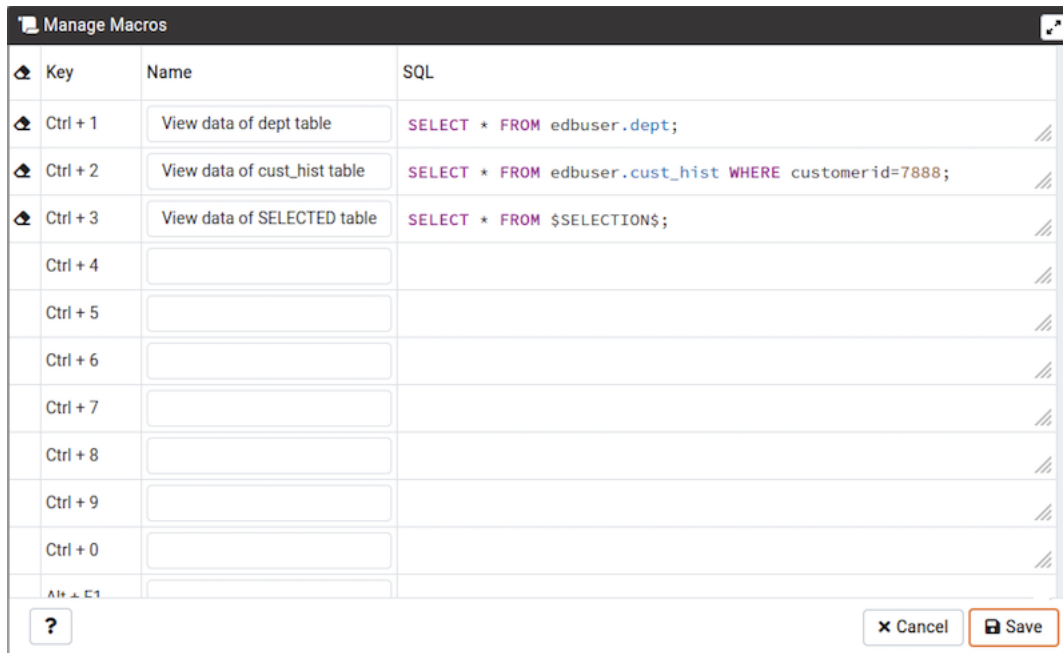


Fig. 21: Query Tool - Connection Status button

To clear a macro, in the Manage Macros dialog box, select the macro and select **Clear**.

To clear the selected macro, respond **Yes** to the prompt.

To clear all macros, select **Clear** next to **Key**. Respond **Yes** to the prompt.

To execute a macro, select the shortcut keys, or select it from the **Macros** menu.

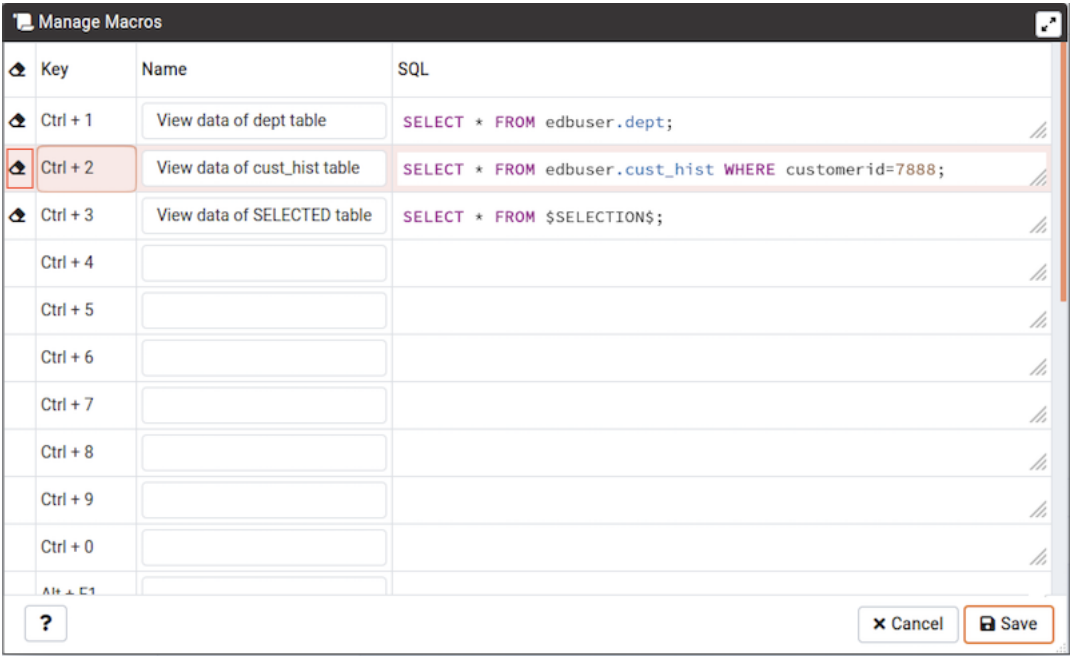


Fig. 22: Query Tool - Connection Status button

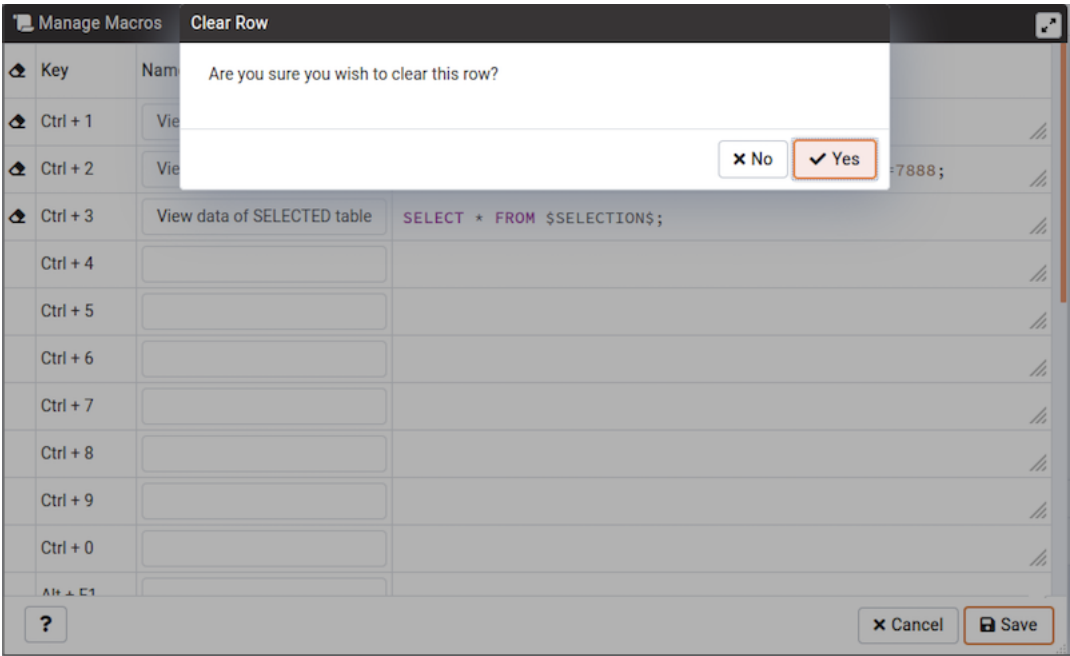


Fig. 23: Query Tool - Connection Status button

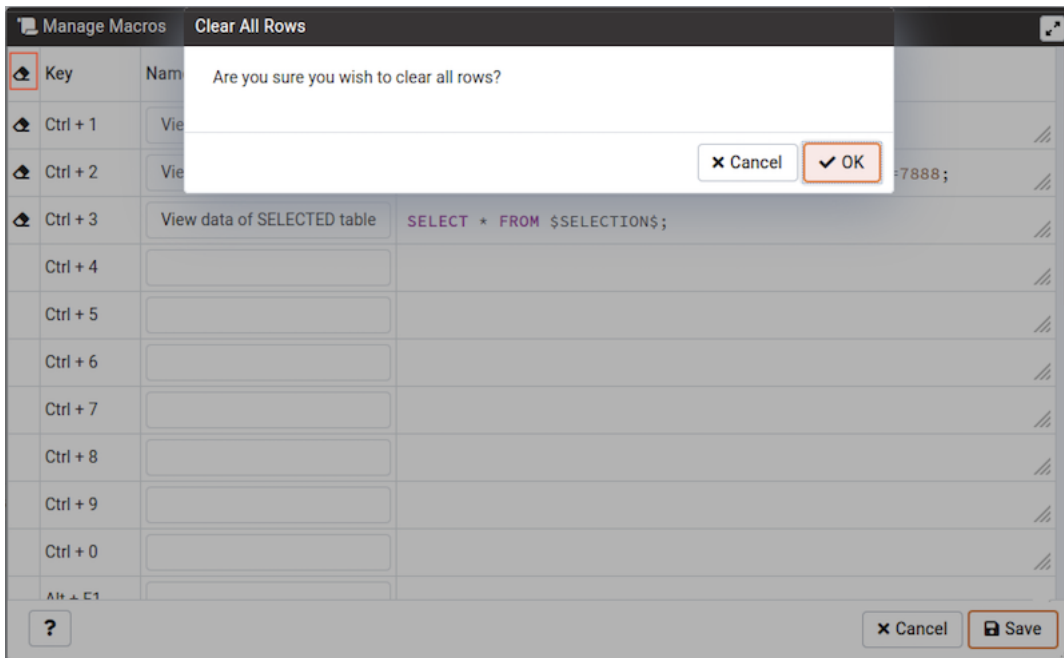


Fig. 24: Query Tool - Connection Status button

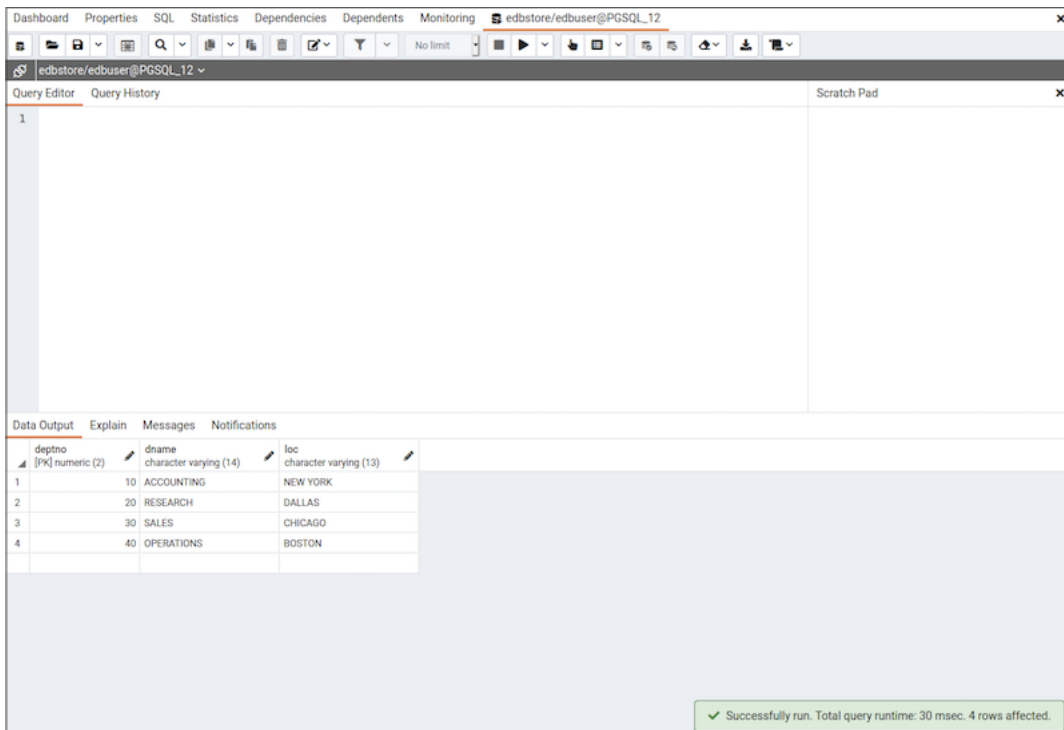


Fig. 25: Query Tool - Connection Status button

USING THE SCHEMA DIFF TOOL

Schema Diff compares schema objects between two database schemas. Use the **Tools** menu to access Schema Diff.

Schema Diff allows you to:

- Compare and synchronize the database schemas (from source to target).
- Visualize the differences between database schemas.
- List the differences in SQL statement for target schema objects.
- Generate synchronization scripts.

To open the selection panel, select **Tools>SchemaDiff** . Select the source and target servers, databases, and schemas to compare. After selecting the objects, select **Compare** .

Note: The source and target databases must use the same major version.

You can open multiple copies of Schema Diff in individual tabs. To close a tab, select **X** in the upper-right corner of the tab bar.

To specify whether to open Schema Diff in a new browser tab, select the Preferences dialog box. Set **Openinnew-browsertab** to **true** .

The Schema Diff panel is divided into two panels: an Object Comparison panel and a DDL Comparison panel.

2.1 Schema Diff Object Comparison panel

In the Object Comparison panel, you can select the source and target servers of the same major version, databases, and schemas to compare. You can select any server listed under the browser tree whether it's connected or disconnected. If you select a server that isn't connected, then you must enter the password before using that server.

Select the databases to compare. The databases can be the same or different, and from the same or different servers.

Select the source and target schemas to compare.

After you select servers, databases, and schemas, select **Compare** to get the comparison results.

Use the lists of **Functions** , **MaterializedViews** , **Tables** , **TriggerFunctions** , **Procedures** , and **Views** to view the DDL statements of all the schema objects.

To filter the schema objects, select **Filter** in the upper-right corner of the Object Comparison panel. Filter the schema objects according to these criteria:

- ****Identical**** — If the **object is** found **in** both schemas **with** the same SQL **↵** statement, then the comparison result **is** identical.

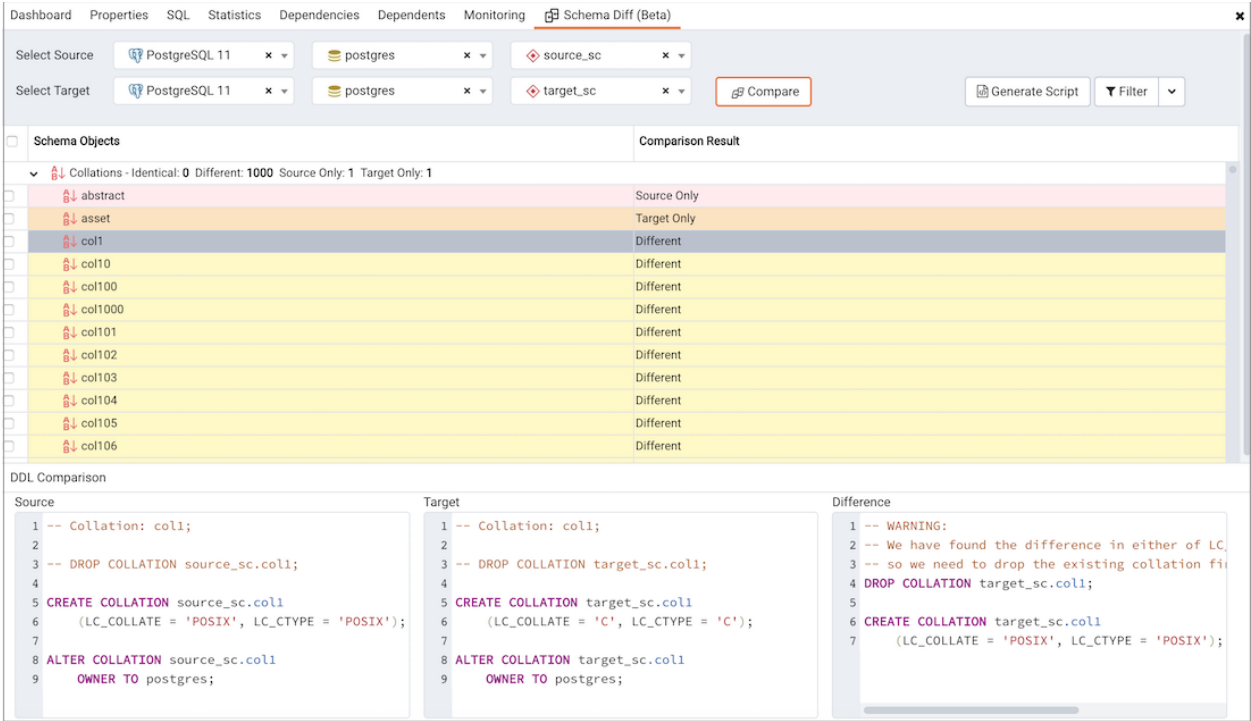


Fig. 1: Schema Diff dialog



Fig. 2: Schema Diff dialog - Compare button

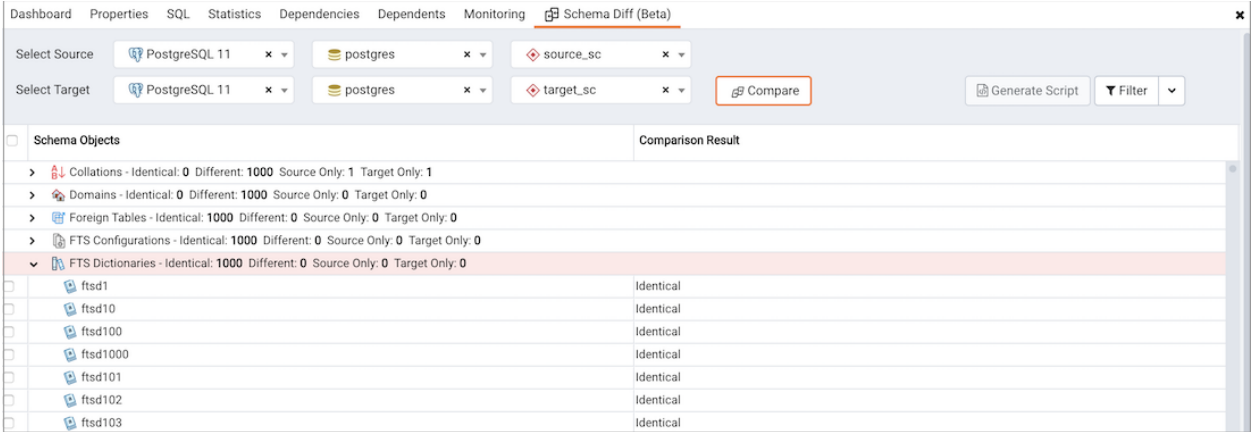


Fig. 3: Schema Diff dialog - Comparison Results

- ****Different**** — If the **object is** found **in** both schemas **with** different SQL statements, then the comparison result **is** different.
- ****SourceOnly**** — If the **object is** found only **in** source schema **and not in** target schema, then the comparison result **is** source only.
- ****TargetOnly**** — If the **object is** found only **in** target schema **and not in** source schema, then the comparison result **is** target only.

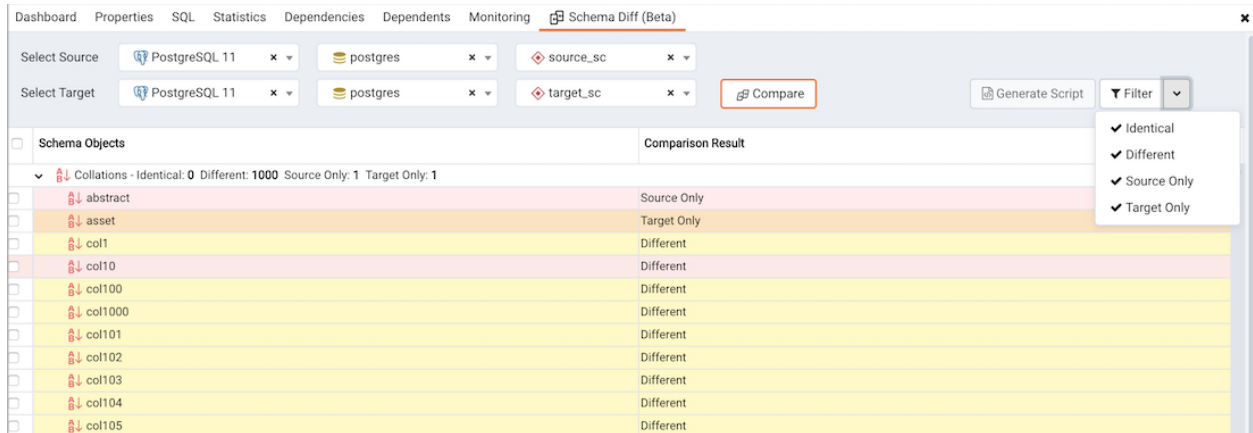


Fig. 4: Schema Diff dialog - Filter option

Select any of the schema objects in the Object Comparison panel to display the DDL statements for that object in the DDL Comparison panel.

2.2 Schema Diff DDL Comparison panel

The DDL Comparison panel displays three columns:

- The first column displays the DDL statement of the object from the source schema.
- The second column displays the DDL statement of the object from the target schema.
- The third column displays the difference in the SQL statement of the target schema object.

To check for differences in the SQL statements, review the DDL statements of all the schema objects.

The Schema Diff tool can generate a SQL script with the differences found in the target schema object. The SQL script compares the target schema object to the SQL statement of the source schema object. To generate the script:

1. In the Object Comparison panel, select the check boxes of the schema objects.
2. Select **GenerateScript**.

To open the Query Tool in a new tab and display the differences in the SQL statement in the Query Editor:

1. Select the schema objects.
2. Select **GenerateScript**.

If you select the schema object to check the difference generated in the DDL Comparison panel but don't select the check box of the schema object, then PEM opens the Query Tool in a new tab. The Query Tool displays the differences in the SQL statements in the Query Editor.

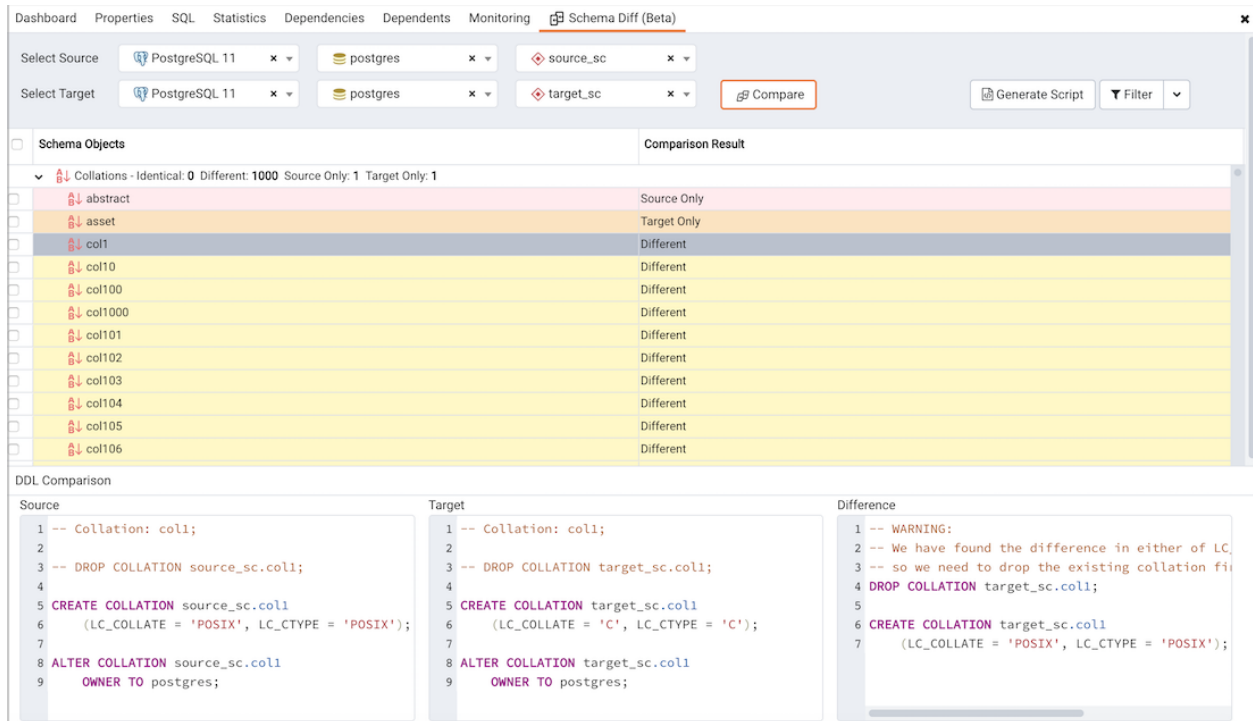


Fig. 5: Schema Diff dialog - DDL Comparison panel

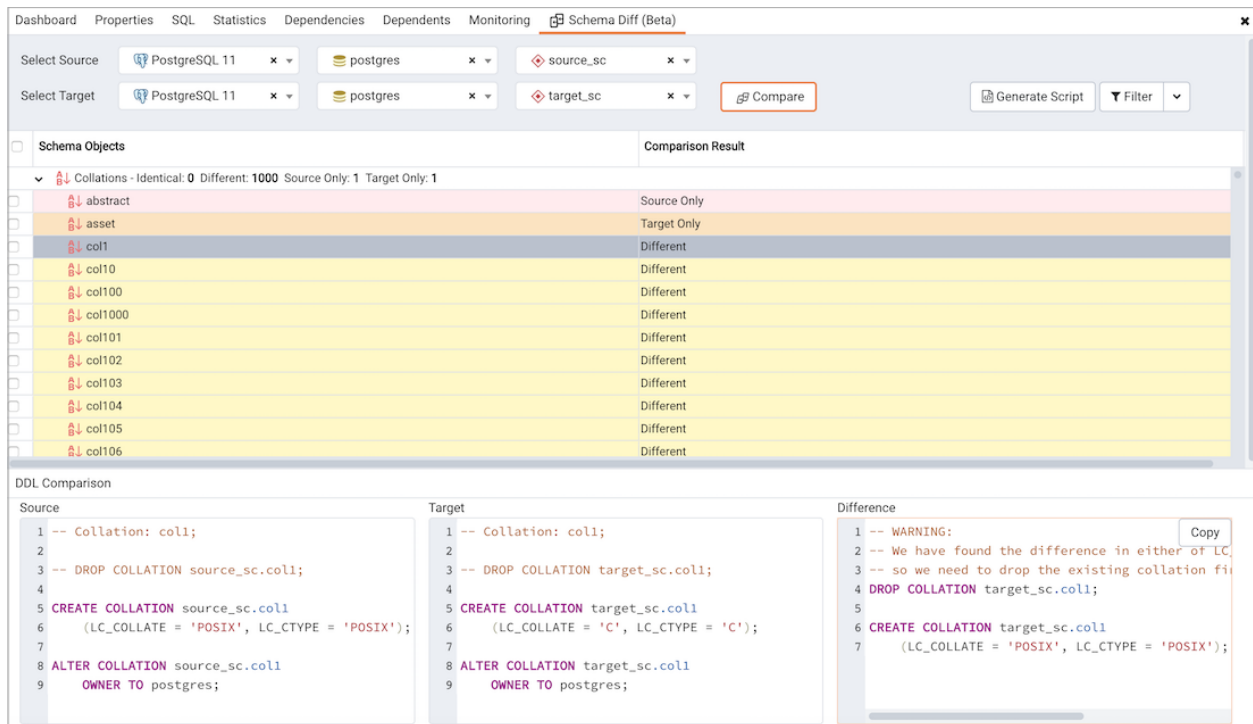


Fig. 6: Schema Diff dialog - Generate Script button

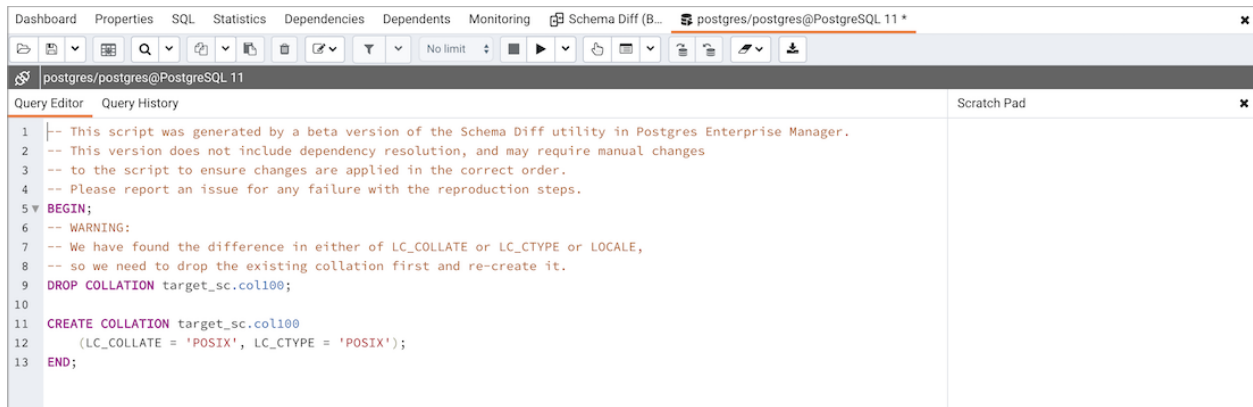


Fig. 7: Schema Diff dialog - Generate Script - Query Editor

To copy the difference generated in the DDL Comparison panel, select **Copy**.

Apply the SQL Statement in the target schema to synchronize the schemas.

USING THE ERD TOOL

The Entity-Relationship Diagram (ERD) tool is a database design tool that provides a graphical representation of database tables, columns, and inter-relationships. ERD can give sufficient information for the database administrator to follow when developing and maintaining the database. To access the ERD tool, select **Tools>ERDTool**. The ERD tool allows you to:

- Design and visualize the database tables and their relationships.
- Add notes to the diagram.
- Align the tables and links for a cleaner look.
- Save the diagram and open it later to continue working on it.
- Generate ready-to-run SQL from the database design.
- Generate the database diagram for an existing database.

You can open multiple copies of the ERD tool in individual tabs simultaneously. To close a copy of the ERD tool, select the **X** in the upper-right corner of the tab bar.

3.1 Toolbar

The ERD Tool toolbar uses context-sensitive icons that provide shortcuts to frequently performed tasks. When an icon is enabled, it appears highlighted. When an icon is disabled, it appears grayed-out.

3.2 File options

Icon	Behavior	Shortcut
Open File	Load a saved diagram.	Ctrl + O
Save	Save changes to a saved diagram or save the diagram to a file.	Ctrl + S
Save as	Open a new dialog box and specify a new location to save the diagram.	Ctrl + Shift + S

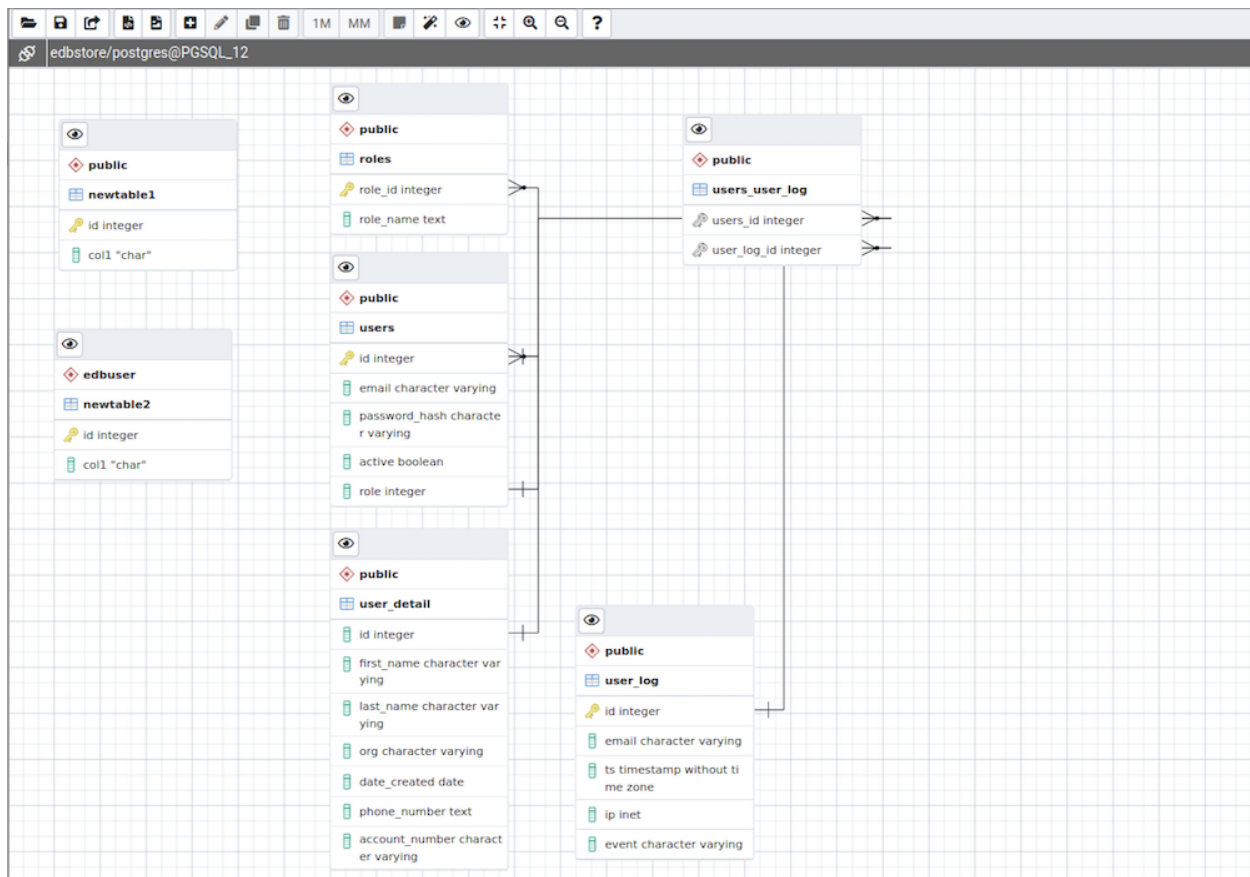


Fig. 1: ERD Tool



Fig. 2: ERD Tool - Toolbar

3.3 Export options

Icon	Behavior	Shortcut
Generate SQL	Generate the DDL SQL for the diagram and open a Query tool with the generated SQL ready for execution.	Option + Ctrl + S
Download image	Save the ERD diagram in an image format.	Option + Ctrl + I

3.4 Editing options

Icon	Behavior	Shortcut
Add table	Add a new table to the diagram. In the table dialog box, enter the table details.	Option/Alt + Ctrl + A
Edit table	Edit a selected table on the diagram. In the table dialog box, enter the table details.	Option/Alt + Ctrl + E
Clone table	Clone the complete table structure, name it with a auto-generated name, and put it in the diagram.	Option/Alt + Ctrl + C
Drop table/link	Drop a selected table or link.	Option/Alt + Ctrl + D

3.5 Table relationship options

Icon	Behavior	Shortcut
1M	Open a one-to-many relationship dialog box to add a relationship between two tables. The selected table is the referencing table and has the <i>many</i> endpoint of the link.	Option/Alt + Ctrl + O
MM	Open a many-to-many relationship dialog box to add a relationship between two tables. This option creates a linked table from the selected columns of the two relating tables.	Option/Alt + Ctrl + M

3.6 Utility Options

Icon	Behavior	Shortcut
Add/Edit note	Make notes on table nodes while designing the database.	Option/Alt + Ctrl + N
Auto align	Align all tables and links for a cleaner look.	Option/Alt + Ctrl + L
Show details	Toggle the column details. You can also choose to show few or more column details.	Option/Alt + Shift + D

3.7 Zoom options

Icon	Behavior	Shortcut
Zoom to fit	Zoom in/out automatically and fit all the tables to the view.	Option/Alt + Shift + F
Zoom in	Zoom in on the diagram.	Option/Alt + Shift + “+”
Zoom out	Zoom out on the diagram.	Option/Alt + Shift + “-”

3.8 Table dialog

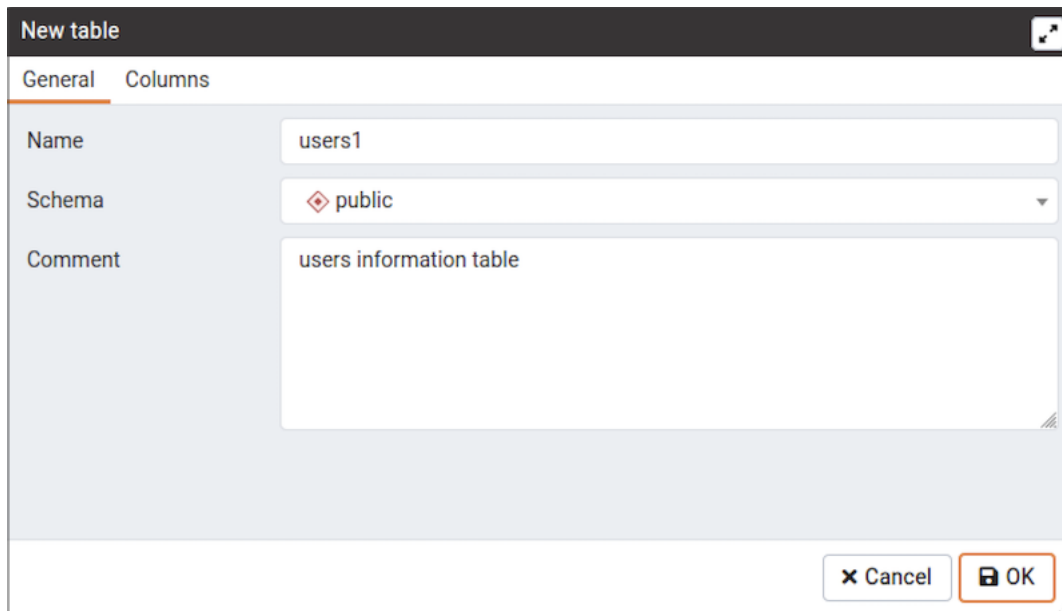


Fig. 3: ERD Tool - Table Dialog

The table dialog box allows you to:

- Change the table structure details.
- Edit an existing table or add a new one.
- See [Table dialog](#) for information on different fields.

3.9 Table node

The table node shows table details in a graphical representation:

- On the top bar: - To show column details, select **Detailstoggle** . - When you add a note, **Note** becomes enabled. Select **Note** to changes notes.
- The first row displays the table’s schema name, like `public` in the example.
- The second row displays the table’s name, like `users` in the example.

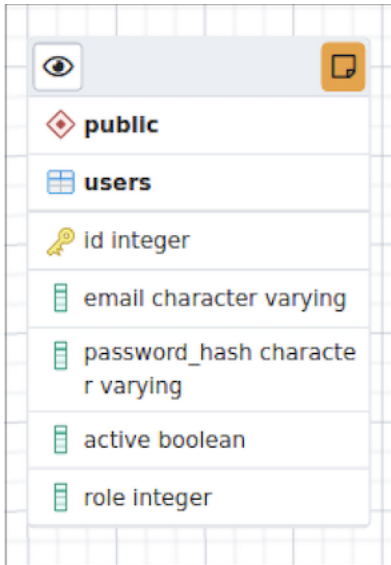


Fig. 4: ERD Tool - Table Node

- The remaining rows display the table's column names and their data types. - A column that is a primary key displays as a lock key, like `id integer` in the example. - Other columns display as a green column, like `email character varying` in the example.
- Select the node to drag it on the canvas.
- To open the table dialog box and change the table details, either double-click the table node or select **Edit** on the toolbar. See

Table dialog for information on different fields.

3.10 One-to-many link dialog box

The one-to-many link dialog box allows you to:

- Add a foreign key relationship between two tables.
- Select a Local Table that references another table. The Local Table has the many end point.
- Select a Local Column that references another column.
- Select the Referenced Table. The Referenced Table has the one end point.
- Select the Referenced Column.

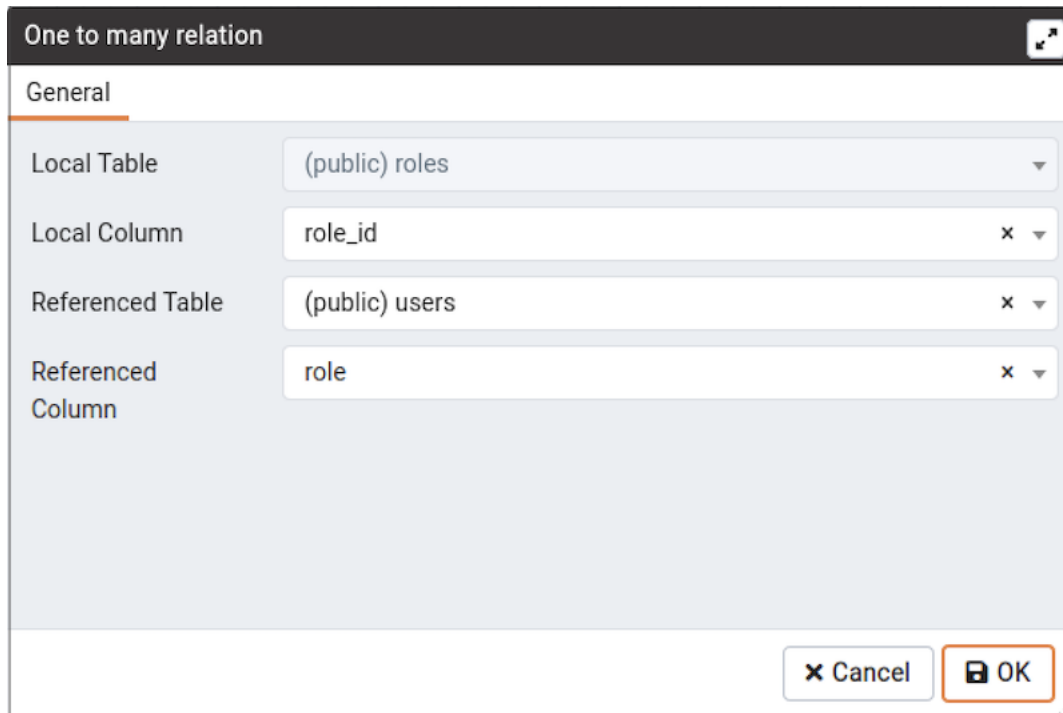


Fig. 5: ERD Tool - One-to-Many Dialog Box

3.11 Many-to-many link dialog box

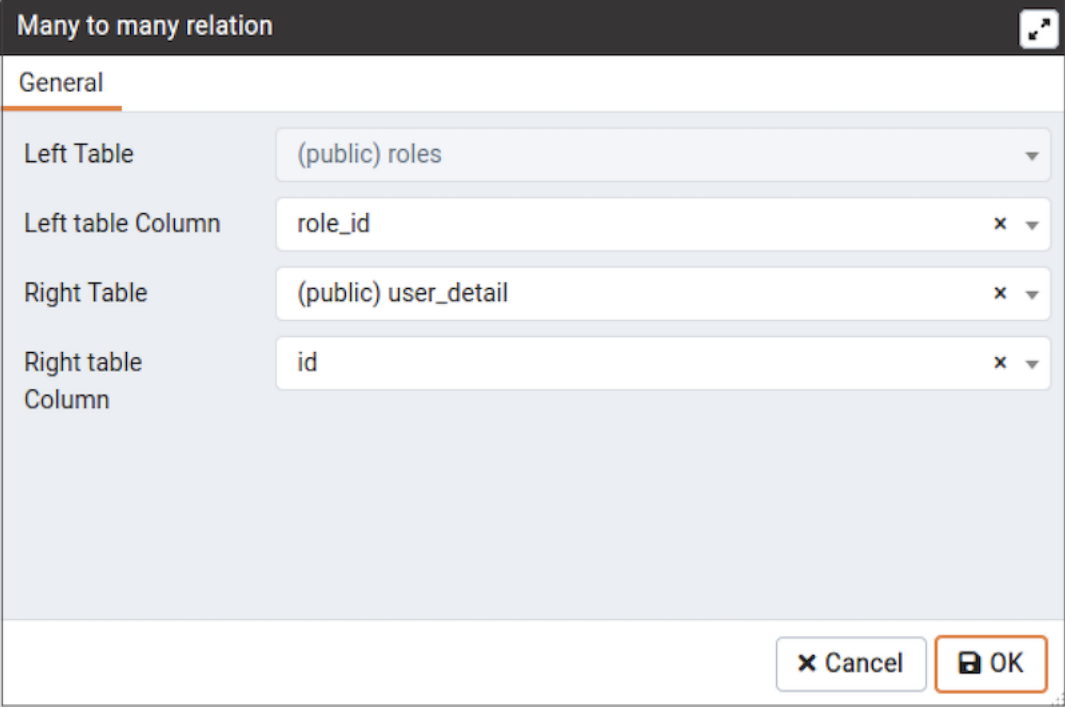
The many-to-many link dialog box allows you to:

- Add a many-to-many relationship between two tables.
- Create a relation table with linked columns that are derived from the two tables.
- Select the Left Table, which is the first table that is linked. The Left Table has the one endpoint of the link with the new relation table.
- Select the Left table Column, which is the first table's column. The Left table Column is a primary key.
- Select the Right Table, which is the second table that is linked. The Right Table has the one endpoint of the link with the new relation table.
- Select the Right table Column, which is the second table's column. The Right table Column is a primary key.

3.12 Table link

The table link shows the relationship between tables:

- The single-line endpoint of the link shows the column that is the reference.
- The three-line endpoint of the link shows the column that references.
- If one of the columns that is a reference or that references is removed from the table, then the link drops.
- Select the link to drag it on the canvas.



The dialog box is titled "Many to many relation" and has a "General" tab. It contains four input fields: "Left Table" with the value "(public) roles", "Left table Column" with the value "role_id", "Right Table" with the value "(public) user_detail", and "Right table Column" with the value "id". Each field has a small "x" icon and a dropdown arrow. At the bottom right, there are "Cancel" and "OK" buttons.

Field	Value
Left Table	(public) roles
Left table Column	role_id
Right Table	(public) user_detail
Right table Column	id

Fig. 6: ERD Tool - Many-to-Many Dialog box

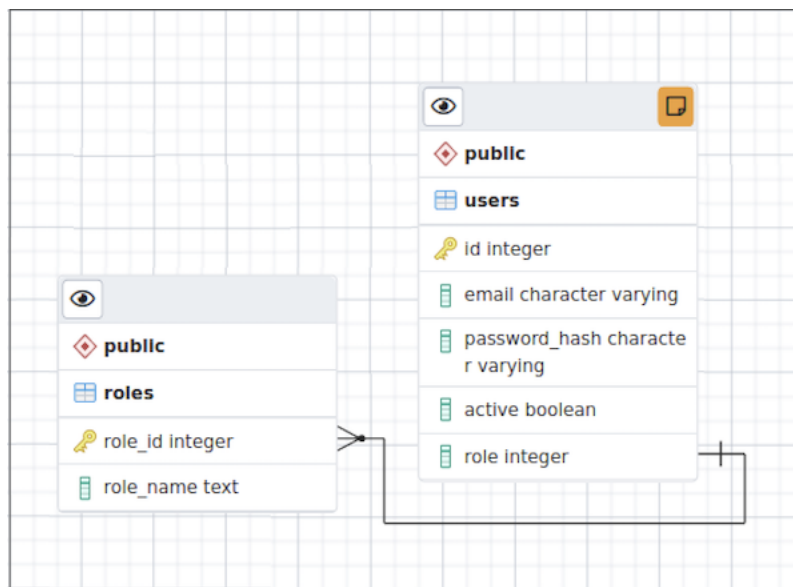


Fig. 7: ERD Tool - Table Link

3.13 Table notes

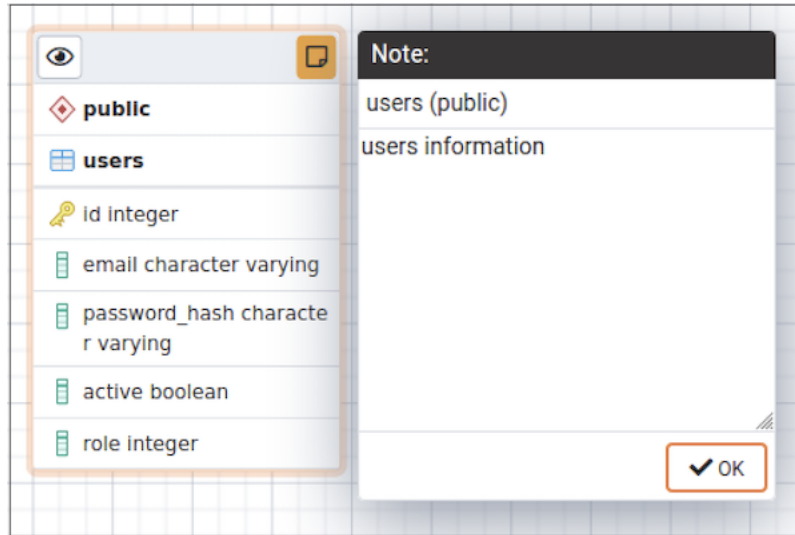


Fig. 8: ERD Tool - Table Notes

- To add a note to a table: 1. On the toolbar, select **Add/Editnote** . 1. In the window, enter your note.
- When a note is added to a table, **Note** becomes enabled on the Table node. To check or update notes, select **Note** .