

Query Tool

The Query Tool is a powerful, feature-rich environment that allows you to execute arbitrary SQL commands and review the result set. You can access the Query Tool via the *Query Tool* menu option on the *Tools* menu, or through the context menu of select nodes of the Object explorer 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 displays the Query Tool interface. At the top, there's a toolbar with various icons for file operations, execution, and settings. Below the toolbar, the 'Query' tab is active, showing a SQL query: `SELECT * FROM pg_class;`. To the right of the query editor is a 'Scratch Pad' tab. Below the query editor, the 'Data Output' tab is active, displaying a table of results. The table has 19 columns: `oid`, `[PK] oid`, `relname`, `relnamespace`, `reltype`, `reltoastrelid`, `relowner`, `relam`, `relfilenode`, `reltablespace`, `relpages`, `reltuples`, `relisvisible`, `reltoastrelid`, `relhasindex`, `relisshared`, and `relperstates`. The table contains 19 rows of data, including system catalogs like `users_id_seq`, `users`, `users_pkey`, `users_email_key`, `test1`, `pg_statistic`, `pg_type`, `pg_toast_1255`, `pg_toast_1255_index`, `pg_toast_1247`, `pg_toast_1247_index`, `pg_toast_2604`, `pg_toast_2604_index`, `pg_toast_2606`, `pg_toast_2606_index`, `pg_toast_2612`, `pg_toast_2612_index`, `pg_toast_2600`, and `pg_toast_2600_index`. The status bar at the bottom indicates 'Total rows: 403' and 'Query complete 00:00:00.237'.

You can open multiple copies of the Query tool in individual tabs simultaneously. To close a copy of the Query tool, click the X of the tab.

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 or a script. It also shows the *History* tab which can be used to view the queries that have been executed in the session, and a *Scratch Pad* which can be used to hold text snippets during editing. If the Scratch Pad is closed, it

can be re-opened (or additional ones opened) by right-clicking in the SQL Editor and other panels and adding a new panel.

- The lower panel displays the *Data Output* 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.

Query Tool in Workspace Layout

The workspace layout offers a distraction-free, dedicated area for the Query Tool. When the Query Tool workspace is accessed, the Welcome page opens by default.

Note: In the Workspace layout, all Query Tool and View/Edit Data tabs open within the Query Tool workspace.

In the classic UI, users must connect to a database server and navigate to the database node before using the Query Tool. However, with the introduction of the Workspace layout and Welcome page, users can seamlessly connect to any ad-hoc server, even if it is not registered in the Object Explorer.

The screenshot shows the PgAdmin interface with the Query Tool workspace. The top menu bar includes File, Object, Tools, and Help. The main area is divided into two sections. On the left, a welcome message states: 'Welcome to the Query Tool Workspace! The Query Tool is a robust and versatile environment designed for executing SQL commands and reviewing result sets efficiently. In this workspace, you can seamlessly open and manage multiple query tabs, making it easier to organize your work. You can select the existing servers or create a completely new ad-hoc connection to any database server as needed.' On the right, a form titled 'Let's connect to the server' is displayed. It contains the following fields: 'Existing Server (Optional)' with a dropdown menu, 'Server Name' with a text input (containing 'PG16-Cloud'), 'Host name/address' with a text input (containing '100.27.21.193'), 'Port' with a text input (containing '5439'), 'Database' with a text input (containing 'postgres'), 'User' with a text input (containing 'postgres'), 'Password' with a masked text input (containing '...'), 'Role' with a dropdown menu, and 'Service' with a text input. Below these fields is a 'Connection Parameters' section with a table that has three columns: 'Name', 'Keyword', and 'Value'. At the bottom of the form, there are two buttons: 'Reset' and 'Connect & Open Query Tool'.

- Select *Existing Server* from the dropdown to connect to a server already listed in the Object Explorer. It is optional.
- Provide the *Server Name* for ad-hoc servers.
- Specify the IP address of the server host, or the fully qualified domain name in the *Host name/address* field.

- Enter the listener port number of the server host in the *Port* field.
- Use the *Database* field to specify the name of the database to which the client will connect.
- Use the *User* field to specify the name of a user that will be used when authenticating with the server.
- Use the *Password* field to provide a password that will be supplied when authenticating with the server.
- Use the *Role* field to specify the name of a role that has privileges that will be conveyed to the client after authentication with the server.
- Use the *Service* field to specify the service name. For more information, see [Section 33.16 of the Postgres documentation](#).
- Use the fields in the *Connection Parameters* to configure the connection parameters.

After filling in all the required fields, click the Connect & Open Query Tool button to launch the Query Tool with the provided server details. If the password is not supplied, you will be prompted to enter it.

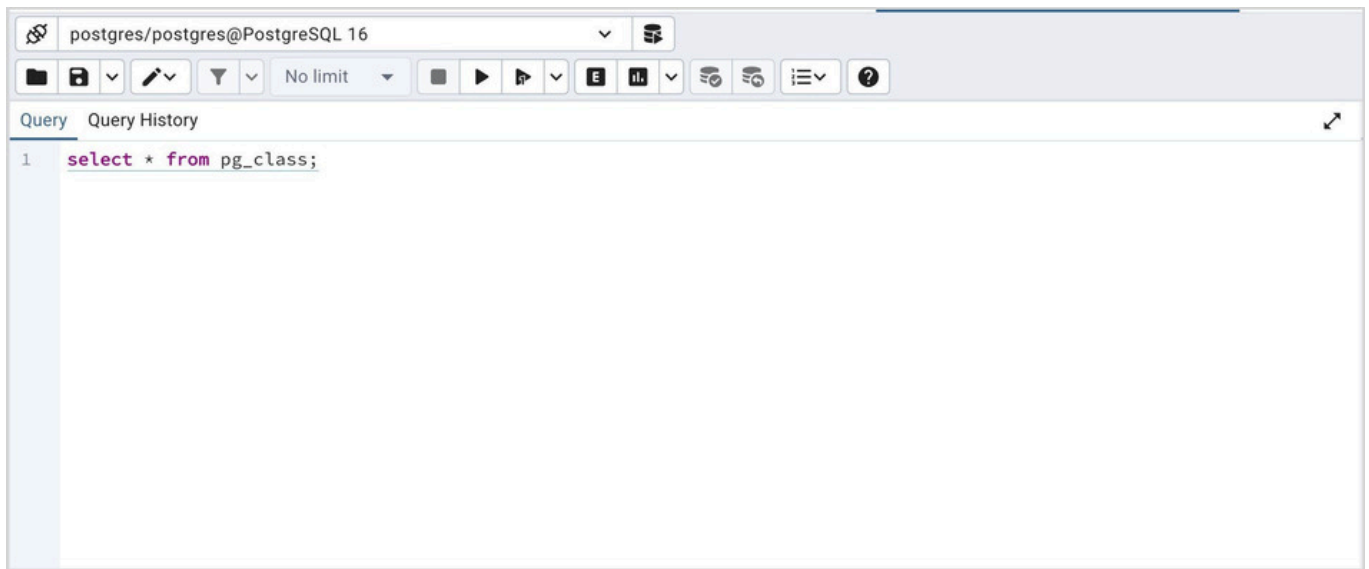
Toolbar

The toolbar is described in the following subsections.

- [Query Tool Toolbar](#)
 - [File Options](#)
 - [Filter/Limit Options](#)
 - [Query Editing Options](#)
 - [Query Execution](#)
 - [Data Editing Options](#)
 - [Pagination Options](#)
 - [Status Bar](#)

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 autocompletion.



To use autocomplete, 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`" (without quotes, but with a trailing space), and then press the Control+Space key combination to select from a popup menu of autocomplete options.



After entering a query, select the *Execute script* icon from the toolbar. The complete contents of the SQL editor panel will be sent to the database server for execution. To execute only a section of the code that is displayed in the SQL editor, highlight the text that you want the server to execute, and click the *Execute script* icon.

postgres/postgres@test1

Query Query History

1 SELECT * FROM pg_class;

Execute script FS

Scratch Pad X

Data Output Messages Notifications

Showing rows: 1 to 403 Page No: 1 of 1

	oid [PK] oid	relname name	relnamespace oid	reltype oid	relsubtype oid	relowner oid	relam oid	reftablespace oid	reftablespace oid	relpages integer	reltuples real	relisvisible integer	reltoastrelid oid	relhasindex boolean	relisshared boolean	relpersister 'char'
1	16386	users_id_seq	2200	0	0	16384	0	16386	0	1	1	0	0	false	false	p
2	16387	users	2200	16389	0	16384	2	16387	0	0	-1	0	0	true	false	p
3	16392	users_pkey	2200	0	0	16384	403	16392	0	1	0	0	0	false	false	p
4	16394	users_email_key	2200	0	0	16384	403	16394	0	1	0	0	0	false	false	p
5	16414	test1	2200	16416	0	16385	2	16414	0	0	-1	0	0	false	false	p
6	2619	pg_statistic	11	12029	0	10	2	2619	0	19	406	19	2840	true	false	p
7	1247	pg_type	11	71	0	10	2	0	0	14	601	14	4171	true	false	p
8	2836	pg_toast_1255	99	0	0	10	2	0	0	1	3	1	0	true	false	p
9	2837	pg_toast_1255_index	99	0	0	10	403	0	0	1	0	0	0	false	false	p
10	4171	pg_toast_1247	99	0	0	10	2	0	0	0	0	0	0	true	false	p
11	4172	pg_toast_1247_index	99	0	0	10	403	0	0	1	0	0	0	false	false	p
12	2830	pg_toast_2604	99	0	0	10	2	2830	0	0	0	0	0	true	false	p
13	2831	pg_toast_2604_index	99	0	0	10	403	2831	0	1	0	0	0	false	false	p
14	2832	pg_toast_2606	99	0	0	10	2	2832	0	0	0	0	0	true	false	p
15	2833	pg_toast_2606_index	99	0	0	10	403	2833	0	1	0	0	0	false	false	p
16	4157	pg_toast_2612	99	0	0	10	2	4157	0	0	0	0	0	true	false	p
17	4158	pg_toast_2612_index	99	0	0	10	403	4158	0	1	0	0	0	false	false	p
18	4159	pg_toast_2600	99	0	0	10	2	4159	0	0	0	0	0	true	false	p
19	4160	pg_toast_2600_index	99	0	0	10	403	4160	0	1	0	0	0	false	false	p

Total rows: 403 Query complete 00:00:00.237 LF Ln 1, Col 24

You can also execute a query based on cursor position. Query tool will detect a query and underline it when cursor position changes. Now, to execute the current underlined query, hit the *Execute query* button on the toolbar. If a section is highlighted then it will behave like normal execute.

postgres/postgres@test1

Query Query History

1 SELECT * FROM pg_class;

2

3 SELECT * FROM pg_attribute;

Execute query Option FS

Scratch Pad X

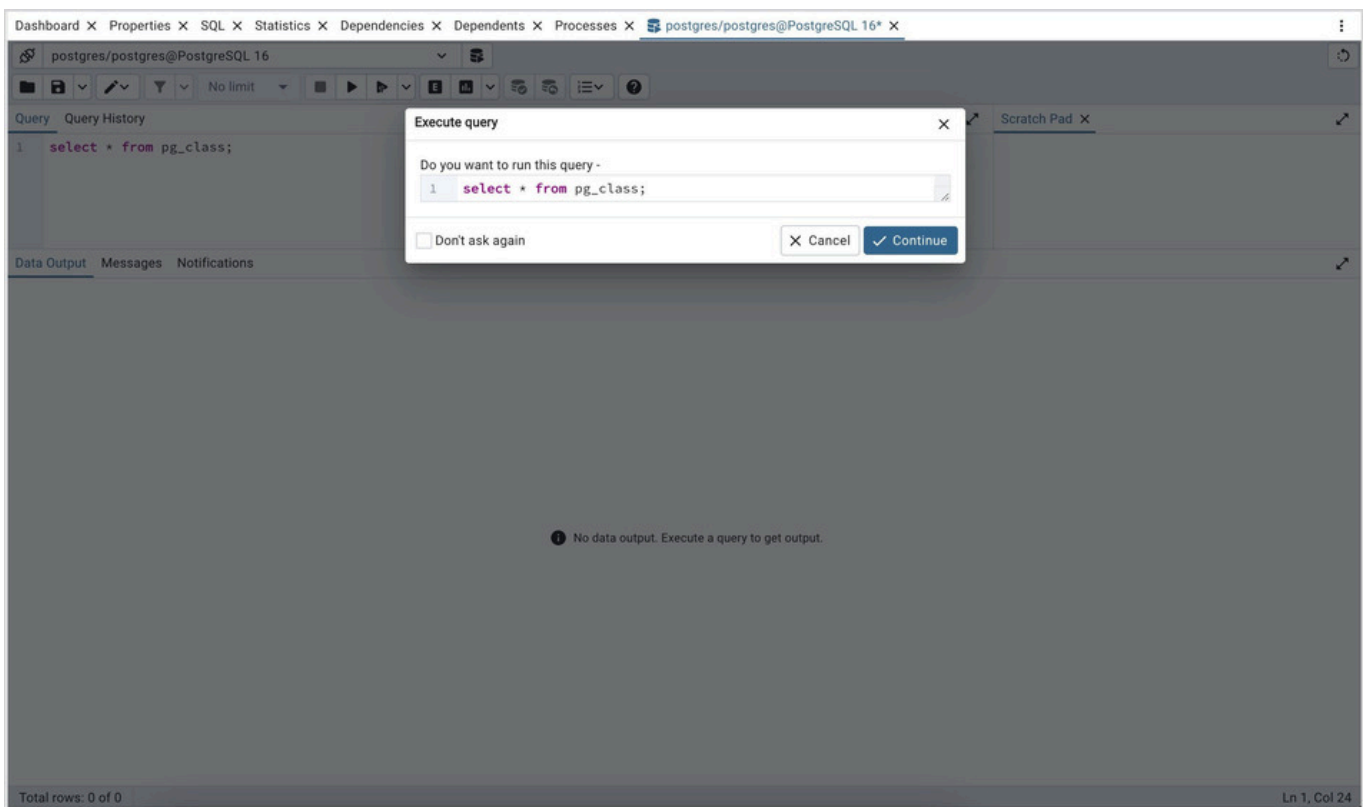
Data Output Messages Notifications

Showing rows: 1 to 403 Page No: 1 of 1

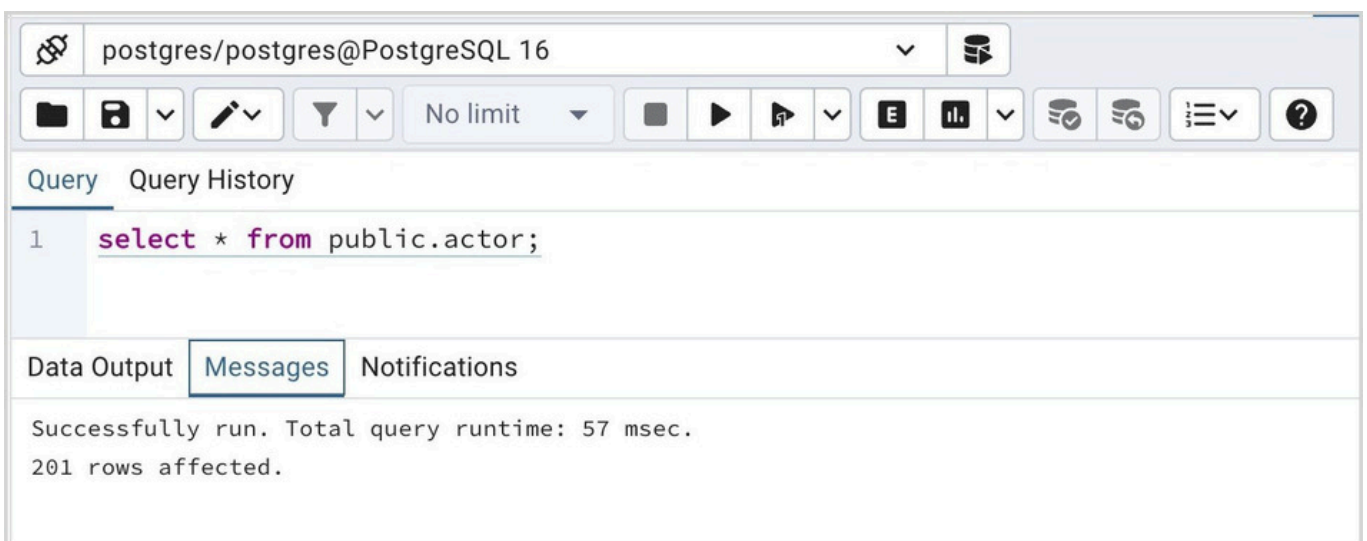
	oid [PK] oid	relname name	relnamespace oid	reltype oid	relsubtype oid	relowner oid	relam oid	reftablespace oid	reftablespace oid	relpages integer	reltuples real	relisvisible integer	reltoastrelid oid	relhasindex boolean	relisshared boolean	relpersister 'char'
1	112	pg_foreign_data_wrapper_oid_index	11	0	0	10	403	112	0	1	0	0	0	false	false	p
2	113	pg_foreign_server_oid_index	11	0	0	10	403	113	0	1	0	0	0	false	false	p
3	174	pg_user_mapping_oid_index	11	0	0	10	403	174	0	1	0	0	0	false	false	p
4	175	pg_user_mapping_user_server_index	11	0	0	10	403	175	0	1	0	0	0	false	false	p
5	548	pg_foreign_data_wrapper_name_index	11	0	0	10	403	548	0	1	0	0	0	false	false	p
6	549	pg_foreign_server_name_index	11	0	0	10	403	549	0	1	0	0	0	false	false	p
7	826	pg_default_acl	11	12088	0	10	2	826	0	0	0	0	4143	true	false	p
8	827	pg_default_acl_role_nsp_obj_index	11	0	0	10	403	827	0	1	0	0	0	false	false	p
9	828	pg_default_acl_oid_index	11	0	0	10	403	828	0	1	0	0	0	false	false	p
10	1213	pg_tablespace	11	12056	0	10	2	0	1664	1	2	1	4185	true	true	p
11	1214	pg_shdepend	11	12060	0	10	2	0	1664	1	37	1	0	true	true	p
12	1232	pg_shdepend_depender_index	11	0	0	10	403	0	1664	14	37	0	0	false	true	p
13	1233	pg_shdepend_reference_index	11	0	0	10	403	0	1664	14	37	0	0	false	true	p
14	1247	pg_type	11	71	0	10	2	0	0	14	601	14	4171	true	false	p
15	1249	pg_attribute	11	75	0	10	2	0	0	55	2971	55	0	true	false	p
16	1255	pg_proc	11	81	0	10	2	0	0	94	3202	94	2836	true	false	p
17	1259	pg_class	11	83	0	10	2	0	0	13	396	13	0	true	false	p
18	1260	pg_authid	11	2842	0	10	2	0	1664	1	21	1	4175	true	true	p
19	1261	pg_auth_members	11	2843	0	10	2	0	1664	1	8	0	0	true	true	p

Total rows: 403 Query complete 00:00:00.150 LF Ln 1, Col 17

The warning will appear only if *Underline query at cursor?* is set to *False* and the *Underlined query execute warning?* switch is set to *True* Preferences Query tool's Options.



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.



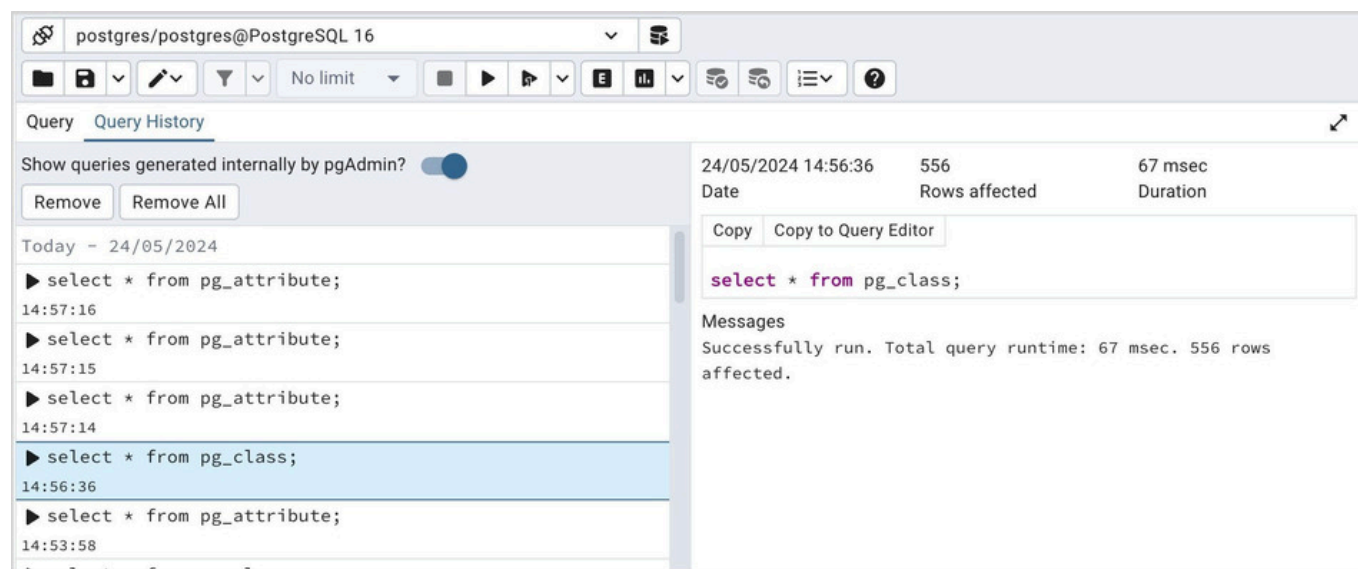
Options on the *Edit* menu offer functionality that helps with code formatting and commenting:

- The auto-indent feature will automatically indent text to the same depth as the previous line when you press 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 treeview which can save time in typing long object names. Text containing the object name will be fully qualified with schema. Double quotes will be added if required. For functions and procedures, the function name along with parameter names will be pasted in the Query Tool.

Query History Panel

Use the *Query History* tab to review activity for the current session:



The Query History 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/Edit Data' or 'Save Data' operations).

You can remove a single query by selecting it and clicking on the *Remove* button. If you would like to remove all of the histories from the *Query History* tab, then click on the *Remove All* button.

By using the *Copy* button, you can copy a particular query to the clipboard, and with the *Copy to Query Editor* button, you can copy a specific query to the Query Editor tab. During this operation, all existing content in the Query Editor is erased.

Query History is maintained across sessions for each database on a per-user basis when running in Query Tool mode. In View/Edit Data mode, history is not retained. By default, the last 20 queries are stored for each database. This can be adjusted in

`config_local.py` or `config_system.py` (see the `config.py` documentation) by overriding the `MAX_QUERY_HIST_STORED` value. See the [Deployment](#) section for more information.

The Data Output Panel

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

Data Output			Messages		Notifications																		
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>			Showing rows: 1 to 403															Page No: 1		of 1		<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	
	oid [PK] oid	relname name	relnamespace oid	reltype oid	relsubtype oid	relowner oid	relam oid	relfilenode oid	reltablespace oid	relpages integer	reltuples real	relisvisible integer	reltoastrelid oid	relhasindex boolean	relisshared boolean	relpersister 'char'							
1	16386	users_id_seq	2200	0	0	16384	0	16386	0	1	1	0	0	false	false	p							
2	16387	users	2200	16389	0	16384	2	16387	0	0	-1	0	0	true	false	p							
3	16392	users_pkey	2200	0	0	16384	403	16392	0	1	0	0	0	false	false	p							
4	16394	users_email_key	2200	0	0	16384	403	16394	0	1	0	0	0	false	false	p							
5	16414	test1	2200	16416	0	16385	2	16414	0	0	-1	0	0	false	false	p							
6	2619	pg_statistic	11	12029	0	10	2	2619	0	19	406	19	2840	true	false	p							
7	1247	pg_type	11	71	0	10	2	0	0	14	601	14	4171	true	false	p							
8	2836	pg_toast_1255	99	0	0	10	2	0	0	1	3	1	0	true	false	p							
9	2837	pg_toast_1255_index	99	0	0	10	403	0	0	1	0	0	0	false	false	p							
10	4171	pg_toast_1247	99	0	0	10	2	0	0	0	0	0	0	true	false	p							
11	4172	pg_toast_1247_index	99	0	0	10	403	0	0	1	0	0	0	false	false	p							
12	2830	pg_toast_2604	99	0	0	10	2	2830	0	0	0	0	0	true	false	p							
13	2831	pg_toast_2604_index	99	0	0	10	403	2831	0	1	0	0	0	false	false	p							
14	2832	pg_toast_2606	99	0	0	10	2	2832	0	0	0	0	0	true	false	p							
15	2833	pg_toast_2606_index	99	0	0	10	403	2833	0	1	0	0	0	false	false	p							
16	4157	pg_toast_2612	99	0	0	10	2	4157	0	0	0	0	0	true	false	p							
17	4158	pg_toast_2612_index	99	0	0	10	403	4158	0	1	0	0	0	false	false	p							
18	4159	pg_toast_2600	99	0	0	10	2	4159	0	0	0	0	0	true	false	p							
19	4160	pg_toast_2600_index	99	0	0	10	403	4160	0	1	0	0	0	false	false	p							
Total rows: 403 Query complete 00:00:00.099																							
LF Ln 1, Col 21																							

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

- Select and copy from the displayed result set.
- Use the *Save results to file* icon to save the content of the *Data Output* tab as a comma-delimited file.
- Edit the data in the result set of a SELECT query if it is updatable.
- Move between pages of data result.

A result set is updatable if:

- All columns are either selected directly from a single table, or are not table columns at all (e.g. concatenation of 2 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.

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

test2/postgres@postgresql

Query Query History

```
1 select * from pg_class;
```

Data Output Messages Notifications

	oid [PK] oid	relname name	relnamespace oid	reltype oid	reloftype oid	relowner oid
1	2619	pg_statistic	11	12029	0	10
2	1247	pg_type	11	71	0	10
3	2836	pg_toast_1255	99	0	0	10
4	2837	pg_toast_1255_index	99	0	0	10
5	4171	pg_toast_1247	99	0	0	10

The psycopg2 driver version should be equal to or above 2.8 for updatable query result sets to work.

An updatable result set is identical to the [Data Grid](#) in View/Edit Data mode, and can be modified in the same way.

If Auto-commit is off, the 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.

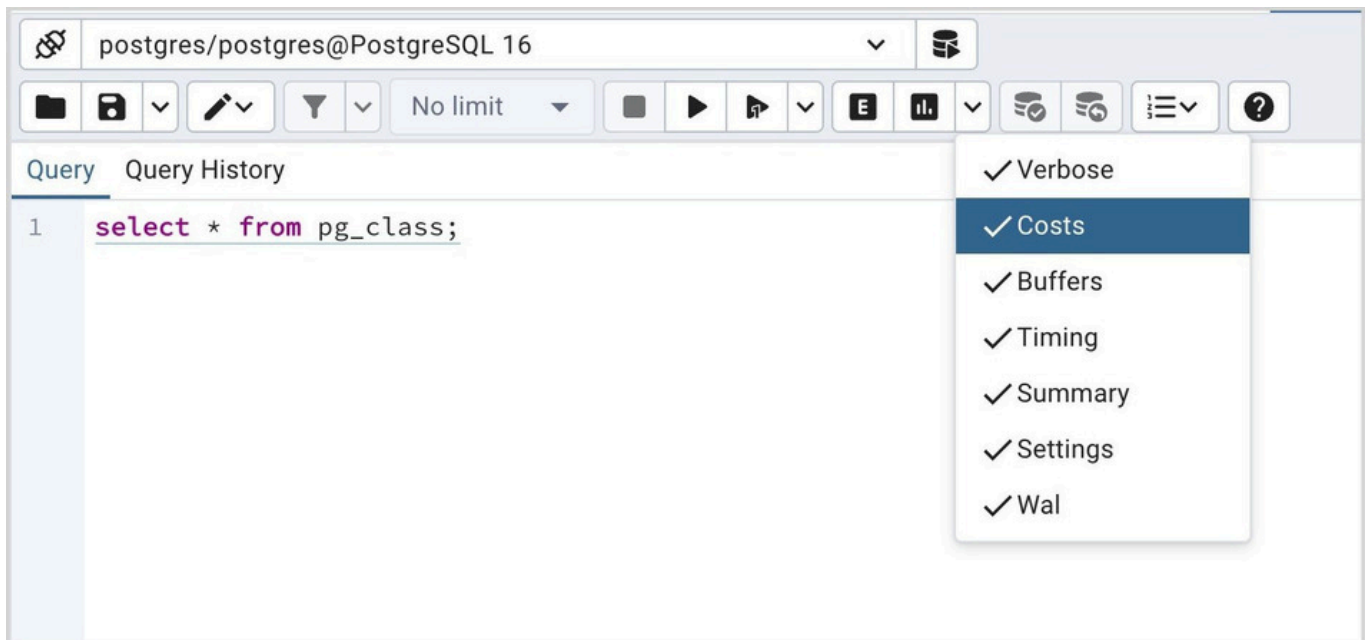
If any errors occur during saving (for example, trying to save NULL into a column with NOT NULL constraint) the data changes are rolled back to an automatically created SAVEPOINT to ensure any previously executed queries in the ongoing transaction are not rolled back.

All rowsets from previous queries or commands that are displayed in the *Data Output* panel will be discarded when you invoke another query; open another Query Tool tab to keep your previous results available.

Explain Panel

To generate the *Explain* or *Explain Analyze* plan of a query, click on *Explain* or *Explain Analyze* button in the toolbar.

More options related to *Explain* and *Explain Analyze* can be selected from the drop down on the right side of *Explain Analyze* button in the toolbar.



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

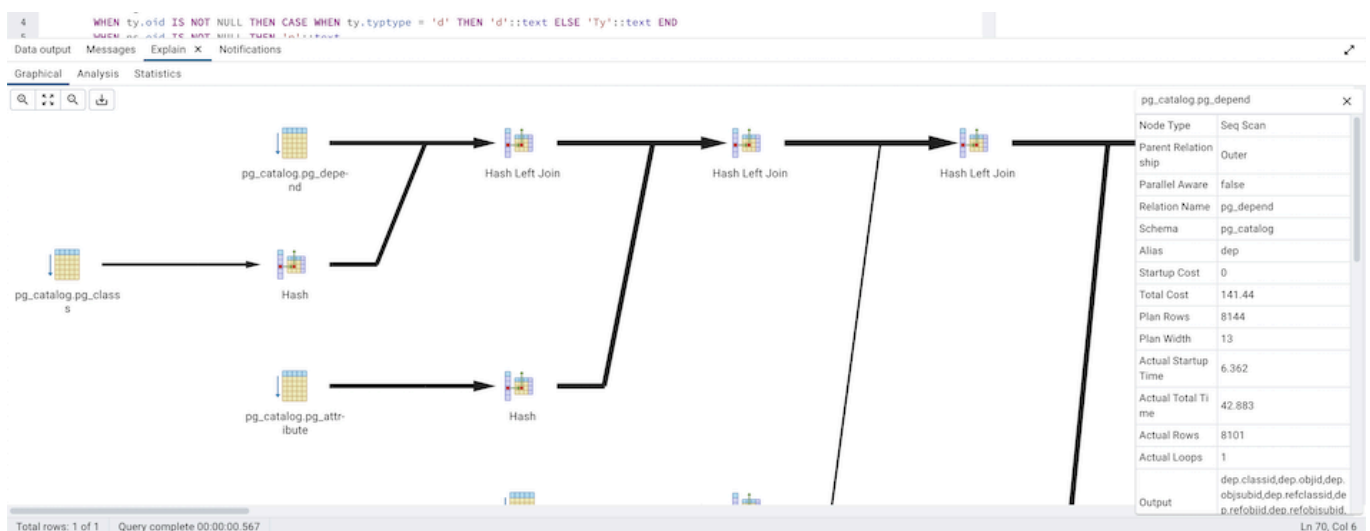
On successful generation of *Explain* plan, it will create three tabs/panels under the Explain panel.

- Graphical

Please note that *EXPLAIN VERBOSE* cannot be displayed graphically. Click on a node icon on the *Graphical* tab to review information about that item; a popup window will display on the right side with the information about the selected object. For information on JIT statistics, triggers and a summary, click on the button on top-right corner; a similar popup window will be displayed when appropriate.

Use the download button on top left corner of the *Explain* canvas to download the plan as an SVG file.

Note: Download as SVG is not supported on Internet Explorer.



Note that the query plan that accompanies the *Explain analyze* is available on the *Data Output* tab.

- Table

Table tab shows the plan details in table format, it generates table format similar to *explain.depesz.com*. Each row of the table represent the data for a *Explain Plan Node*. It may contain the node information, exclusive timing, inclusive timing, actual vs planned rows differences, actual rows, planned rows, loops.

background color of the exclusive, inclusive, and Rows X columns may vary based on the difference between actual vs planned.

If percentage of the exclusive/inclusive timings of the total query time is: > 90 - Red color > 50 - Orange (between red and yellow) color > 10 - Yellow color

If planner mis-estimated number of rows (actual vs planned) by 10 times - Yellow color
100 times - Orange (between Red and Yellow) color 1000 times - Red color

4

WHEN ty_oid IS NOT NULL THEN CASE WHEN ty_type = 'd' THEN 'd':text ELSE 'ty':text END

WHEN ty_oid IS NOT NULL THEN CASE WHEN ty_type = 'd' THEN 'd':text ELSE 'ty':text END

Messages

Explain X

Notifications

Graphical

Analysis

Statistics

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Unique (cost=14859.48..15109.23 rows=8325 width=358) (actual=300.754..301.706 rows=1547 loops=1)	0.805 ms	301.706 ms	↑ 5.39	1547	8325	1
2.	→ Sort (cost=14859.48..14880.29 rows=8325 width=358) (actual=300.753..300.902 rows=2115 loops=1)	4.758 ms	300.902 ms	↑ 3.94	2115	8325	1
3.	→ Hash Inner Join (cost=1299.3..14317.39 rows=8325 width=358) (actual=241.743..296.144 rows=2115 loops=1) Hash Cond: (dep_classid = pg_class.oid)	2.784 ms	296.144 ms	↑ 3.94	2115	8325	1
4.	→ Hash Left Join (cost=1269.74..13378.58 rows=193976 width=2181) (actual=178.755..293.152 rows=8101 loops=1) Hash Cond: (dep_objid = pl.oid)	2.823 ms	293.152 ms	↑ 23.95	8101	193976	1
5.	→ Hash Left Join (cost=1250.51..12845 rows=193976 width=2117) (actual=178.715..290.32 rows=8101 loops=1) Hash Cond: (ftsd_dictname = ftsdn.oid)	2.928 ms	290.32 ms	↑ 23.95	8101	193976	1
6.	→ Merge Left Join (cost=1249.38..11975.02 rows=193976 width=2057) (actual=178.612..287.341 rows=8101 loops=1)	3.021 ms	287.341 ms	↑ 23.95	8101	193976	1
7.	→ Merge Left Join (cost=1248.36..11484.41 rows=193976 width=1989) (actual=177.395..283.121 rows=8101 loops=1)	2.662 ms	283.121 ms	↑ 23.95	8101	193976	1
8.	→ Merge Left Join (cost=1248.36..11484.41 rows=193976 width=1989) (actual=177.395..283.121 rows=8101 loops=1)	19.125 ms	279.859 ms	↑ 23.95	8101	193976	1
9.	→ Merge Left Join (cost=1248.36..11484.41 rows=193976 width=1989) (actual=177.395..283.121 rows=8101 loops=1)	2.601 ms	259.338 ms	↑ 23.95	8101	193976	1
10.	→ Merge Left Join (cost=1243.81..7384.62 rows=39746 width=1721) (actual=175.342..256.711 rows=8101 loops=1)	2.494 ms	256.711 ms	↑ 4.91	8101	39746	1
11.	→ Merge Left Join (cost=1242.1..7262.51 rows=39746 width=1649) (actual=158.523..237.408 rows=8101 loops=1)	2.474 ms	237.408 ms	↑ 4.91	8101	39746	1
12.	→ Merge Left Join (cost=1241.95..6731.66 rows=39746 width=1581) (actual=158.516..234.931 rows=8101 loops=1)	2.566 ms	234.931 ms	↑ 4.91	8101	39746	1
13.	→ Merge Left Join (cost=1241.68..5889.42 rows=39746 width=1449) (actual=157.234..194.683 rows=8101 loops=1)	2.34 ms	194.683 ms	↑ 4.91	8101	39746	1
14.	→ Merge Left Join (cost=1238.73..5755.73 rows=39746 width=1317) (actual=155.435..190.549 rows=8101 loops=1)	2.147 ms	190.549 ms	↑ 4.91	8101	39746	1
15.	→ Nested Loop Left Join (cost=1238.58..5064.43 rows=8144 width=1313) (actual=155.421..188.393 rows=8101 loops=1)	4.223 ms	188.393 ms	↑ 1.01	8101	8144	1

Total rows: 1 of 1 Query complete 00:00:00.567

Ln 70, Col 6

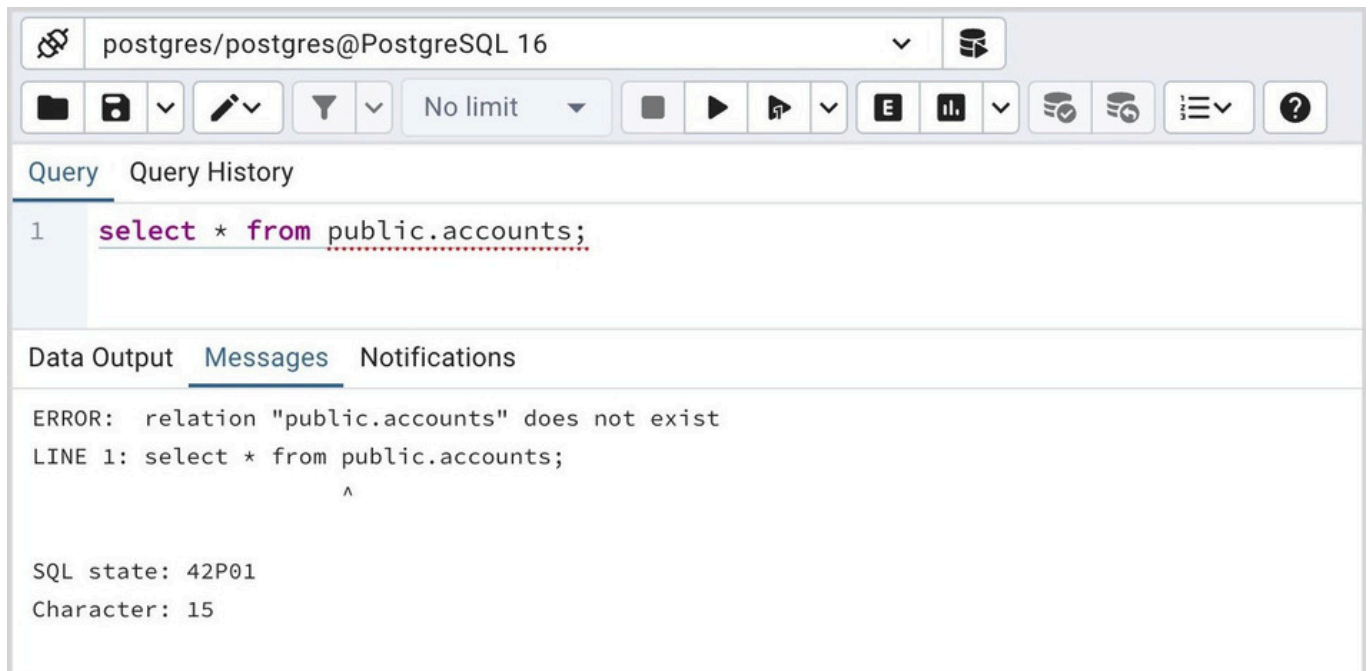
- Statistics

Statistics tab shows two tables: 1. Statistics per Plan Node Type 2. Statistics per Table

Statistics per Node Type				Statistics per Relation			
Node type	Count	Time spent	% of query	Relation name	Scan count	Total time	% of query
Hash	14	20.198 ms	6.7%	Node type	Count	Sum of times	% of relation
Hash Inner Join	1	2.784 ms	0.93%	pg_catalog.pg_attrdef	1	0.057 ms	0.02%
Hash Left Join	11	24.555 ms	8.14%	Index Scan	1	0.057 ms	100%
Hash Right Join	2	0.166 ms	0.06%	pg_catalog.pg_attribute	1	0.974 ms	0.33%
Index Only Scan	1	0.016 ms	0.01%	Seq Scan	1	0.974 ms	100%
Index Scan	9	37.858 ms	12.55%	pg_catalog.pg_class	4	0.48 ms	0.16%
Materialize	8	0.309 ms	0.11%	Index Scan	1	0.002 ms	0.42%
Merge Left Join	17	56.481 ms	18.73%	Seq Scan	3	0.478 ms	99.59%
Nested Loop Left Join	5	14.584 ms	4.84%	pg_catalog.pg_collation	1	36.745 ms	12.18%
Seq Scan	27	133.022 ms	44.09%	Index Scan	1	36.745 ms	100%
Sort	11	10.959 ms	3.64%	pg_catalog.pg_constraint	1	0.033 ms	0.02%
Unique	1	0.805 ms	0.27%	Index Scan	1	0.033 ms	100%
				pg_catalog.pg_depend	1	42.883 ms	14.22%
				Seq Scan	1	42.883 ms	100%
				pg_catalog.pg_event_trigger	1	0.003 ms	0.01%
				Index Scan	1	0.003 ms	100%
				pg_catalog.pg_extension	1	1.169 ms	0.39%

Messages Panel

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



The screenshot shows a database client interface for PostgreSQL 16. The top toolbar includes icons for file operations, query execution, and settings. The main area is divided into three tabs: Query, Query History, and Messages. The Messages tab is active, displaying an error message: "ERROR: relation \"public.accounts\" does not exist". The error message is followed by the SQL query: "LINE 1: select * from public.accounts;". The SQL editor shows the query "select * from public.accounts;" with the word "from" underlined in red. The Messages panel also displays the SQL state "42P01" and the character position "15".

postgres/postgres@PostgreSQL 16

Query Query History

```
1 select * from public.accounts;
```

Data Output Messages Notifications

ERROR: relation "public.accounts" does not exist
LINE 1: select * from public.accounts;
 ^

SQL state: 42P01
Character: 15

If the server returns an error, the error message will be displayed on the *Messages* tab, and the syntax that caused the error will be 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:

The screenshot shows the Query Tool interface with the connection 'postgres/postgres@PostgreSQL 16' selected. The 'Query' tab is active, displaying two SQL queries: `select * from pg_class;` and `select * from pg_attribute;`. The 'Messages' tab is selected in the bottom panel, showing the message: 'Successfully run. Total query runtime: 67 msec. 556 rows affected.'

Notifications Panel

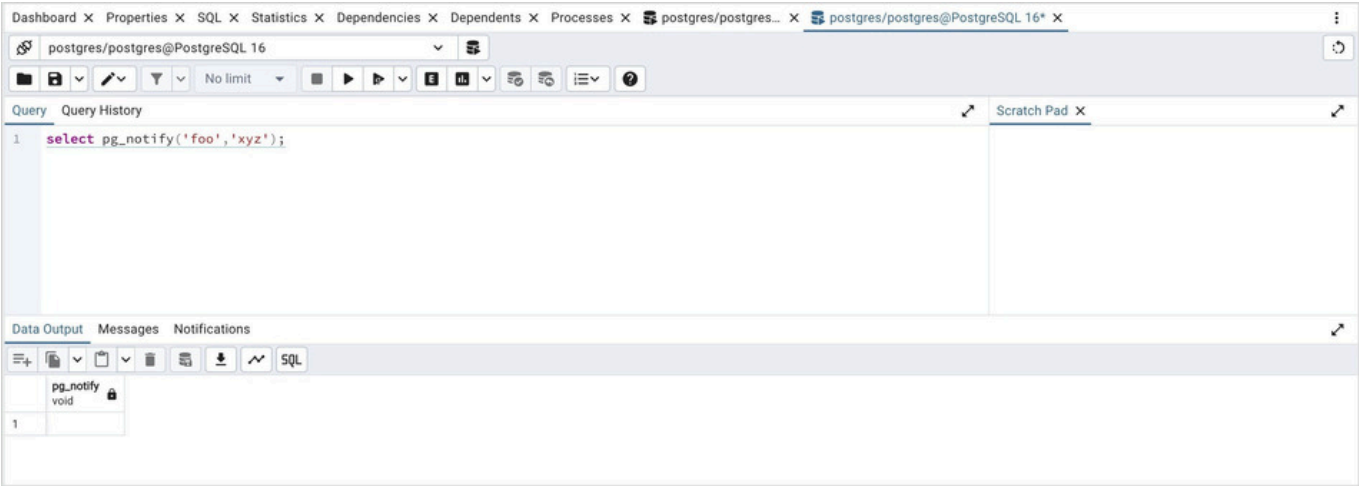
Use the *Notifications* tab to view the notifications using PostgreSQL *Listen/ Notify* feature. For more details see [PostgreSQL documentation](#).

Example:

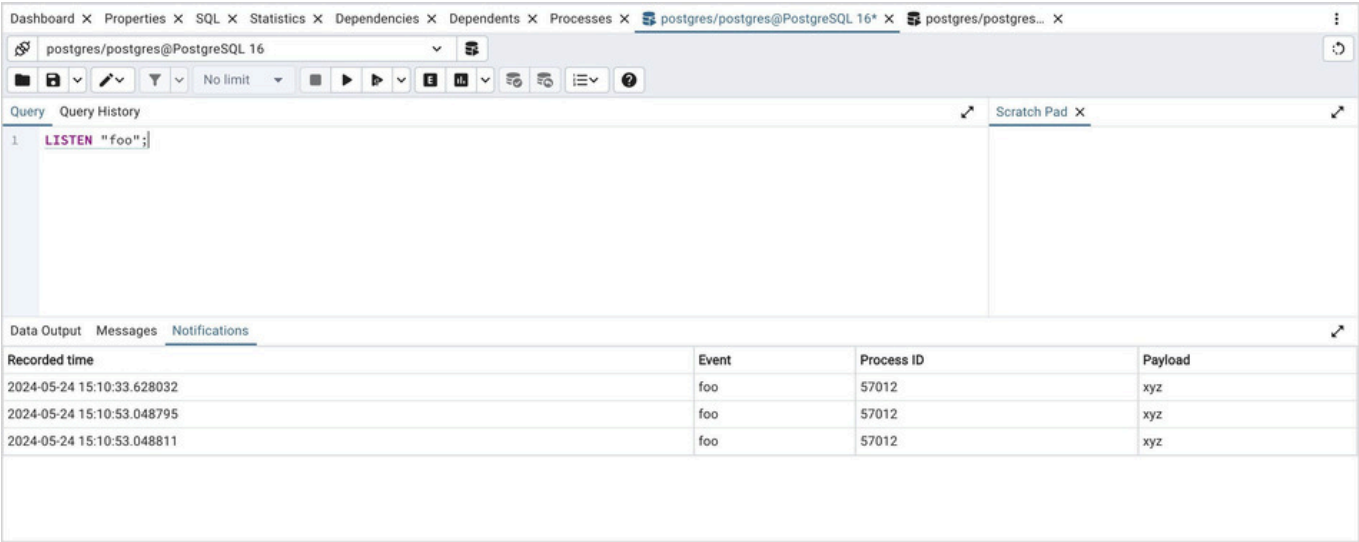
1. Execute *LISTEN "foo"* in first *Query Tool* session

The screenshot shows the Query Tool interface with the connection 'postgres/postgres@PostgreSQL 16' selected. The 'Query' tab is active, displaying the query `LISTEN "foo"`. The 'Notifications' tab is selected in the bottom panel, showing a table with columns: 'Recorded time', 'Event', 'Process ID', and 'Payload'. The table is currently empty.

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.

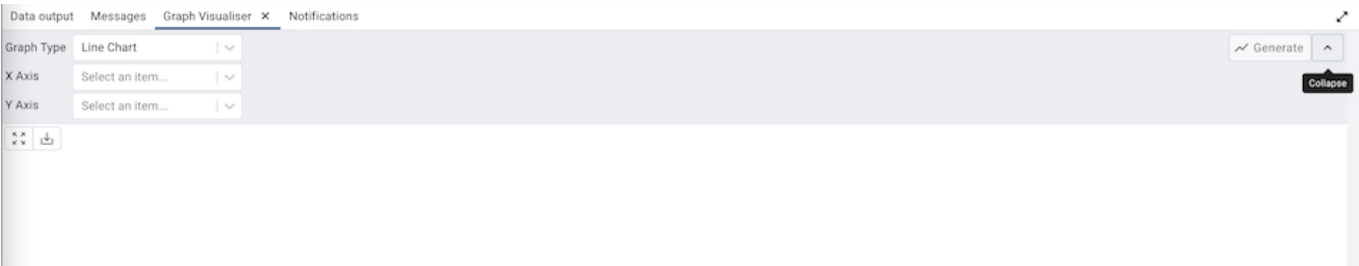


3. You can observe the *Notification* tab in the first *Query Tool* session where it shows the Recorded time, Event, Process ID, and the Payload of the particular channel.



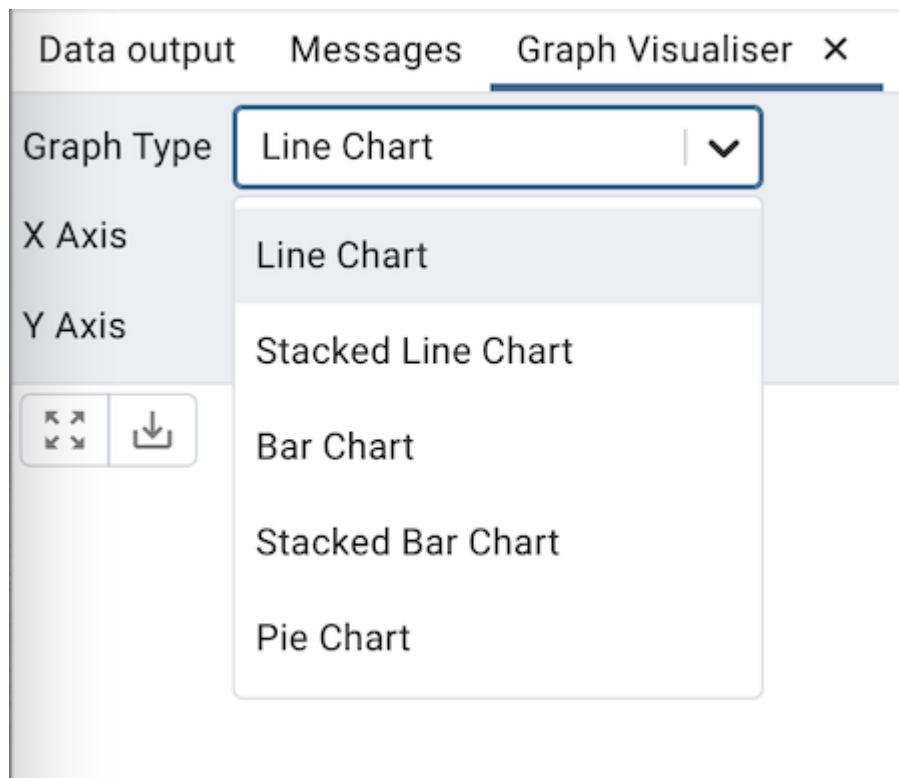
Graph Visualiser Panel

Click the Graph Visualiser button in the toolbar to generate the *Graphs* of the query results. The graph visualiser supports Line Charts, Stacked Line Charts, Bar Charts, Stacked Bar Charts, and Pie Charts.



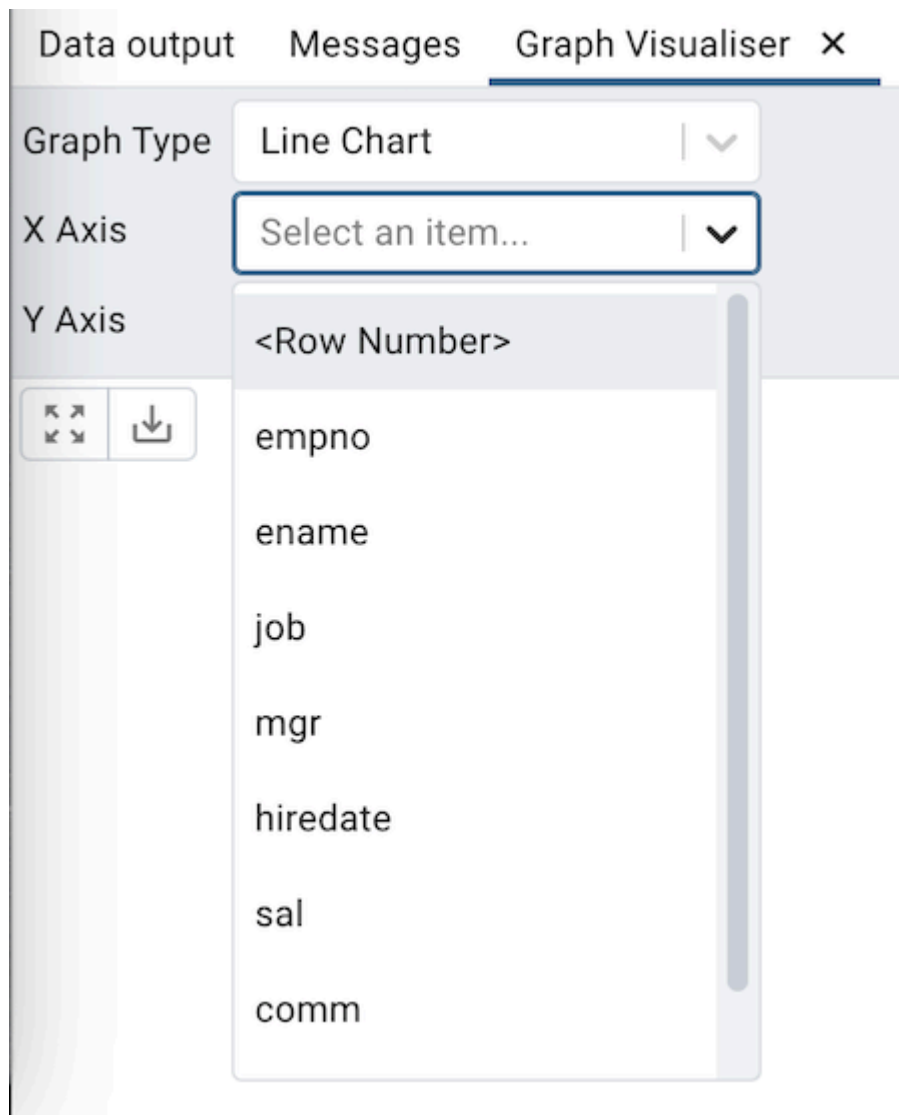
- Graph Type

Choose the type of the graph that you would like to generate.



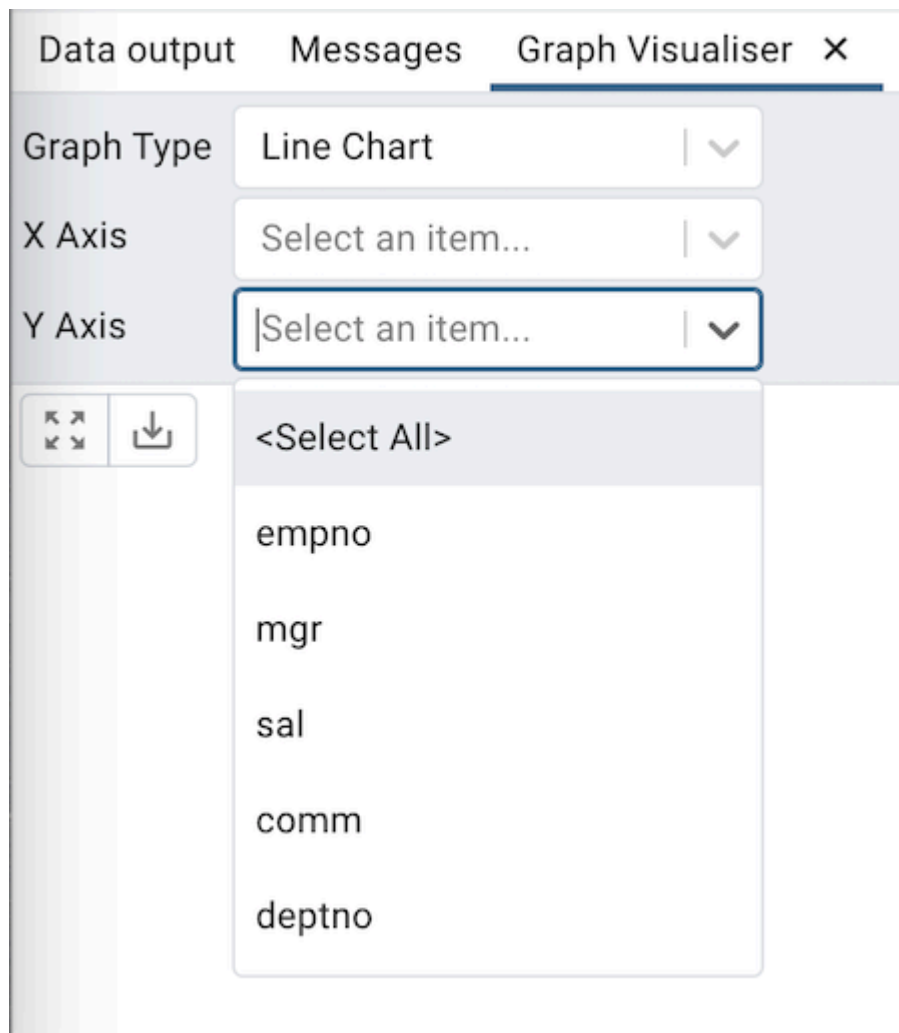
- X Axis

Choose the column whose value you wish to display on X-axis from the *X Axis* dropdown. Select the *<Row Number>* option to use the number of rows as labels on the X-axis.



- Y Axis

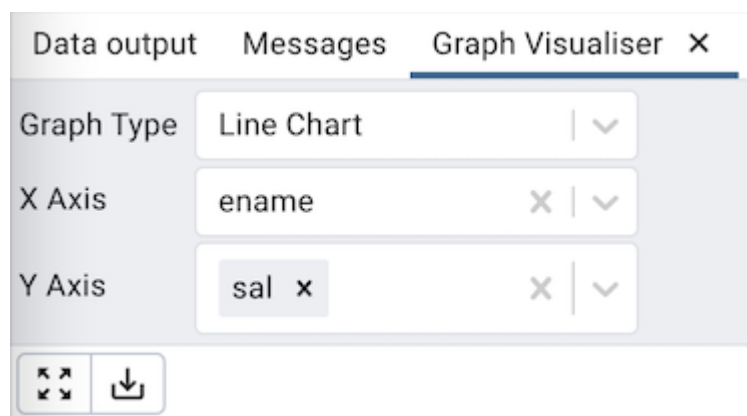
Choose the columns whose value you wish to display on Y-axis from the *Y Axis* drop-down. Users can choose multiple columns. Choose the *<Select All>* option from the drop-down menu to select all the columns.



- Download and Zoom button

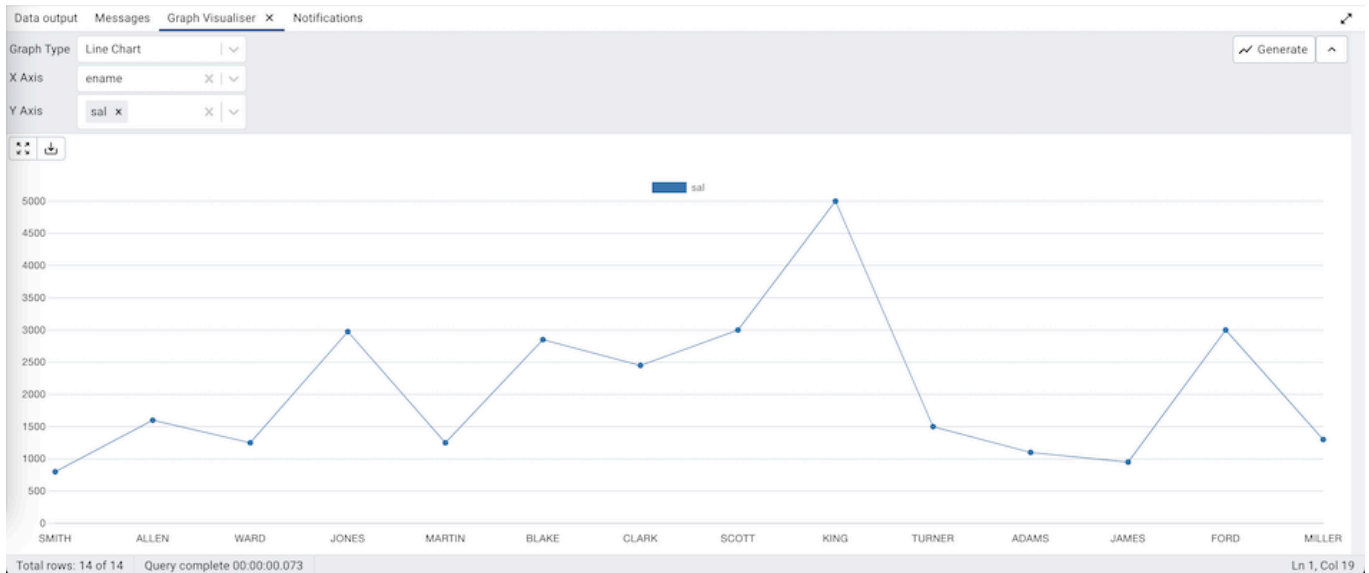
Zooming is performed by clicking and selecting an area over the chart with the mouse. The *Zoom to original* button will bring you back to the original zoom level.

Click the *Download* button on the button bar to download the chart.



Line Chart

The *Line Chart* can be generated by selecting the 'Line Chart' from the Graph Type drop-down, selecting the X-axis and the Y-axis, and clicking on the 'Generate' button. Below is an example of a chart of employee names and their salaries.



Set *Use different data point styles?* option to true in the [Preferences Dialog](#), to show data points in a different style on each graph lines.

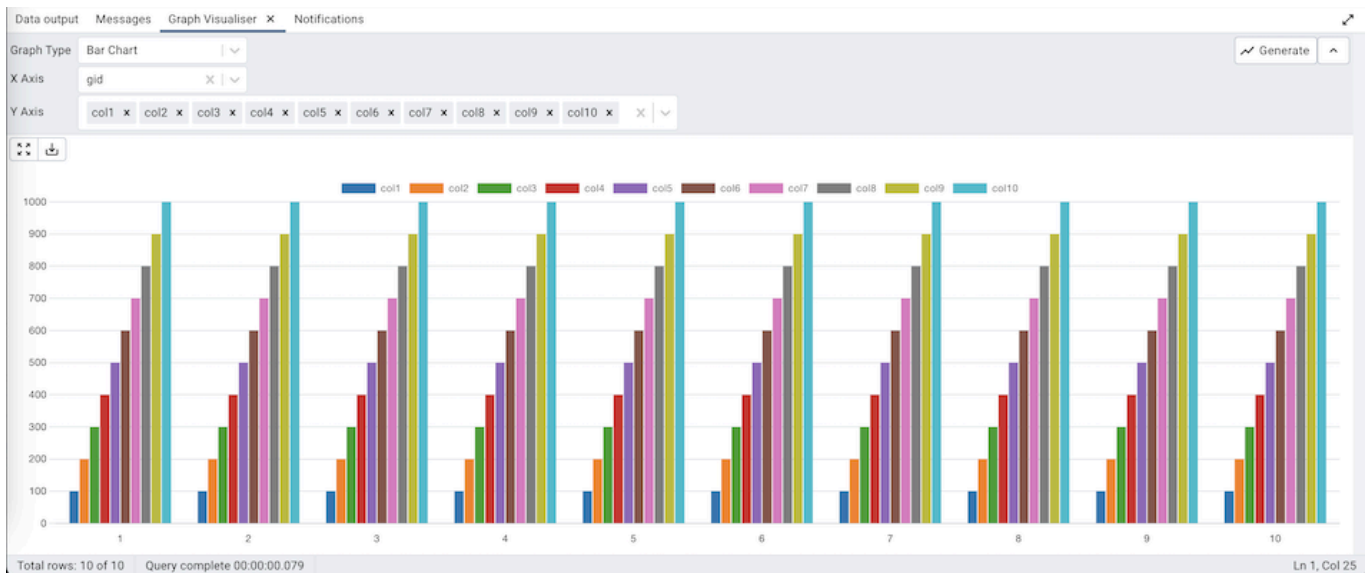
Stacked Line Chart

The *Stacked Line Chart* can be generated by selecting the 'Stacked Line Chart' from the Graph Type drop-down, selecting the X-axis and the Y-axis, and clicking on the 'Generate' button.



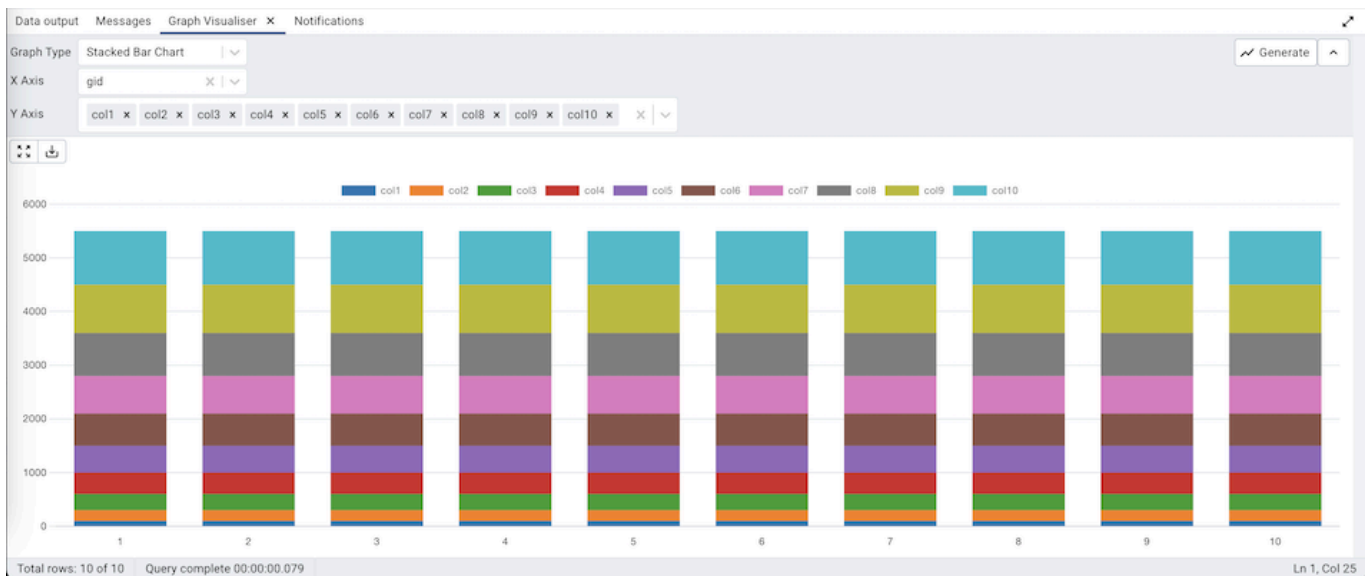
Bar Chart

The *Bar Chart* can be generated by selecting the 'Bar Chart' from the Graph Type drop-down, selecting the X-axis and the Y-axis, and clicking on the 'Generate' button.



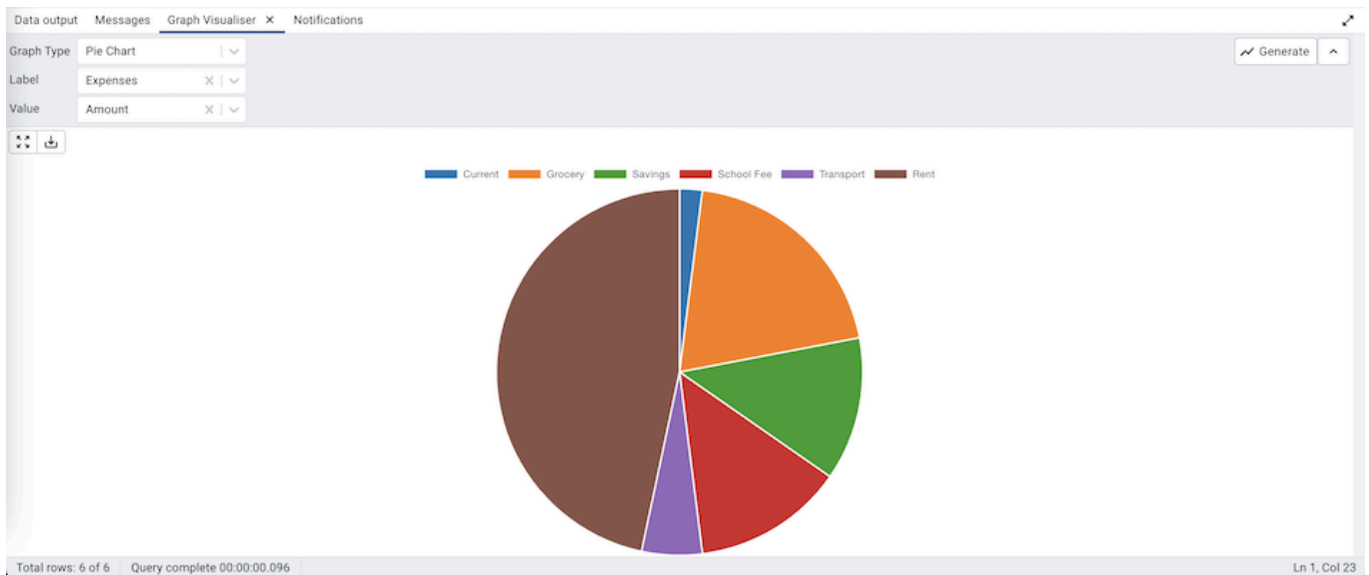
Stacked Bar Chart

The *Stacked Bar Chart* can be generated by selecting the 'Stacked Bar Chart' from the Graph Type drop-down, selecting the X-axis and the Y-axis, and clicking on the 'Generate' button.



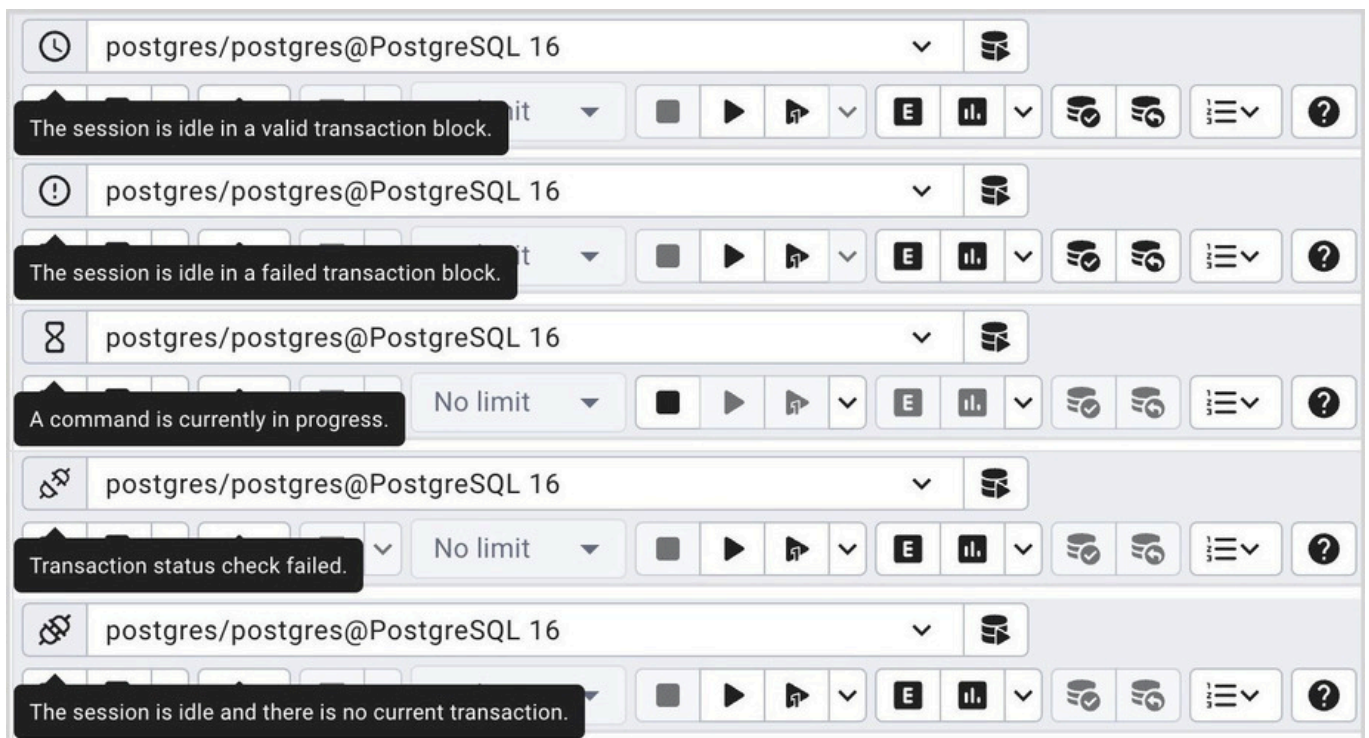
Pie Chart

The *Pie Chart* can be generated by selecting the 'Pie Chart' from the Graph Type drop-down, selecting the Label and Value, and clicking on the 'Generate' button.



Connection Status

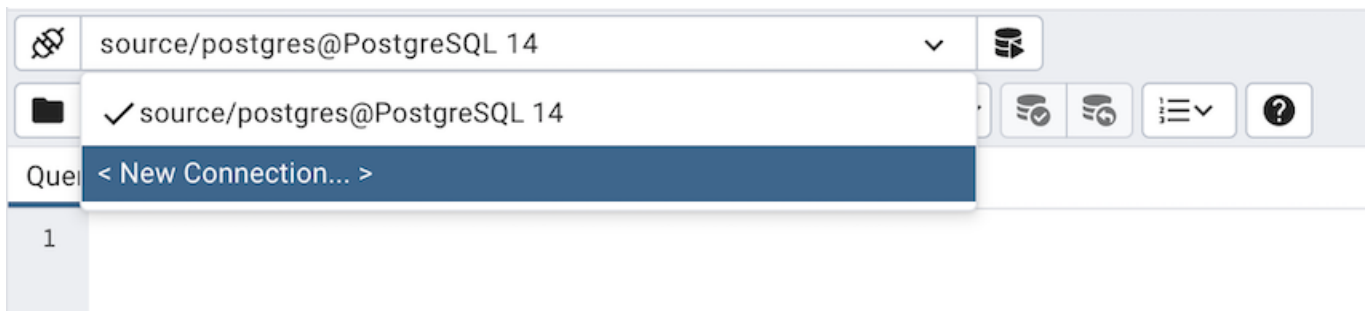
Use the *Connection status* feature to view the current connection and transaction status by clicking on the status icon in the Query Tool:



Change connection

User can connect to another server or database from existing open session of query tool.

- Click on the connection link next to connection status.
- Now click on the *<New Connection>* option from the dropdown.



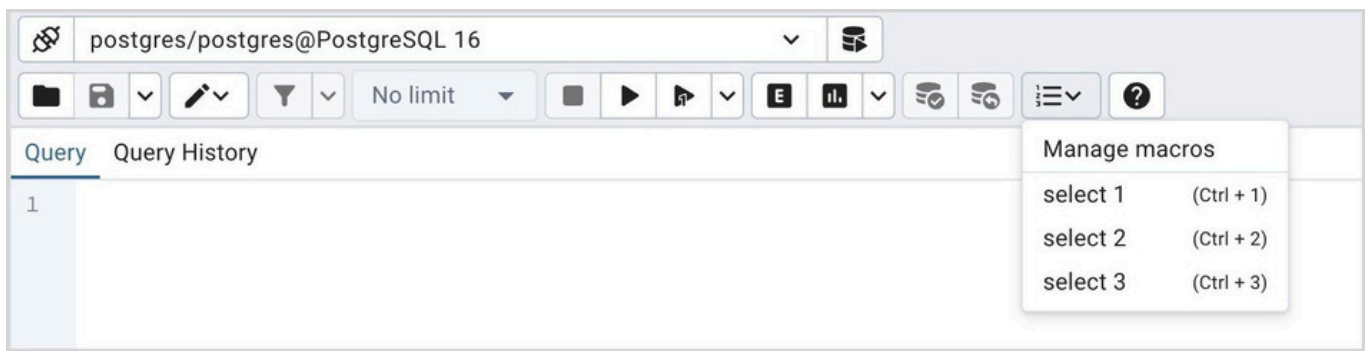
- Now select server, database, user, and role to connect and click on the 'Save' button.

 A screenshot of the 'Add New Connection' dialog box. The dialog has a title bar with 'Add New Connection' and a close button (X). Inside, there are four fields: 'Server' with a dropdown showing 'PostgreSQL 14' and a blue elephant icon; 'Database' with a dropdown showing 'postgres'; 'User' with a dropdown showing 'postgres'; and 'Role' with a dropdown showing 'Select an item...'. At the bottom of the dialog, there are three buttons: 'Close' (with an X icon), 'Reset' (with a circular arrow icon), and 'Save' (with a floppy disk icon). There are also information (i) and help (?) icons on the left side of the bottom bar.

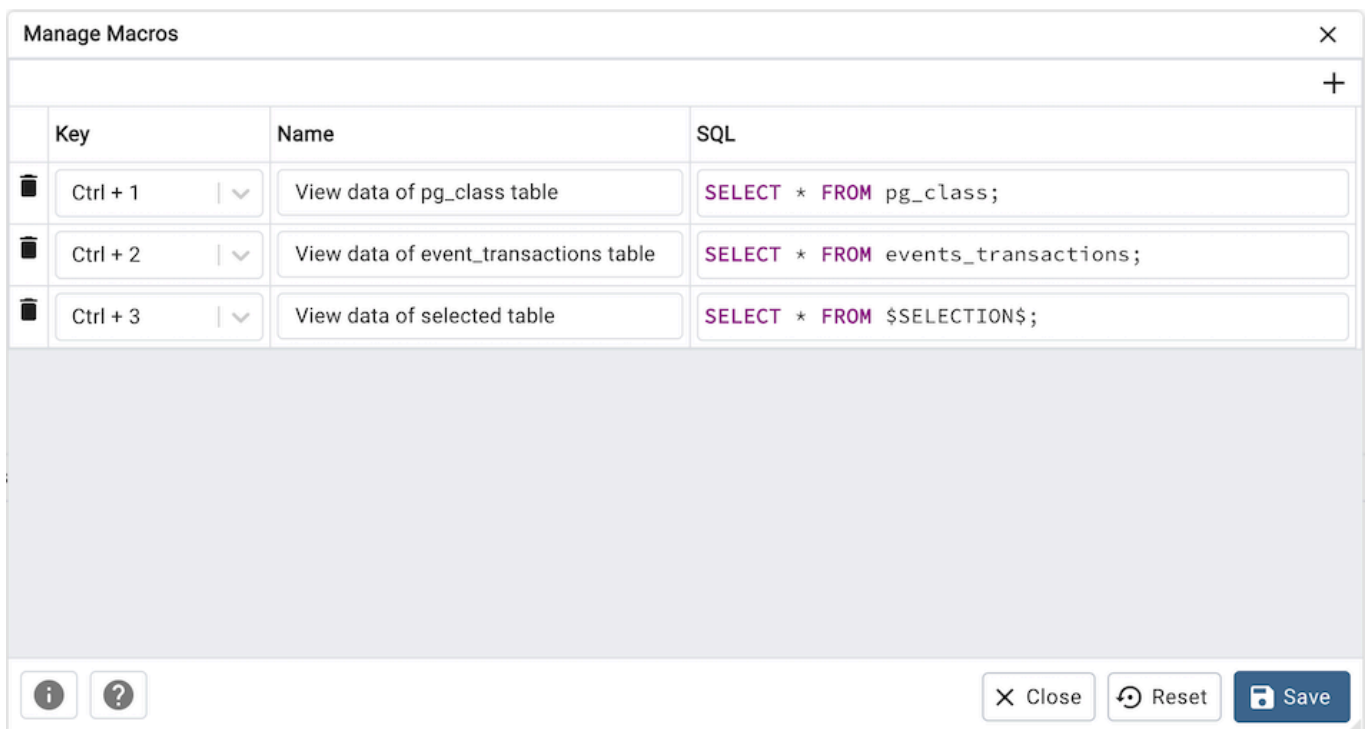
- A newly created connection will now get listed in the options.
- To connect, select the newly created connection from the dropdown list.

Macros

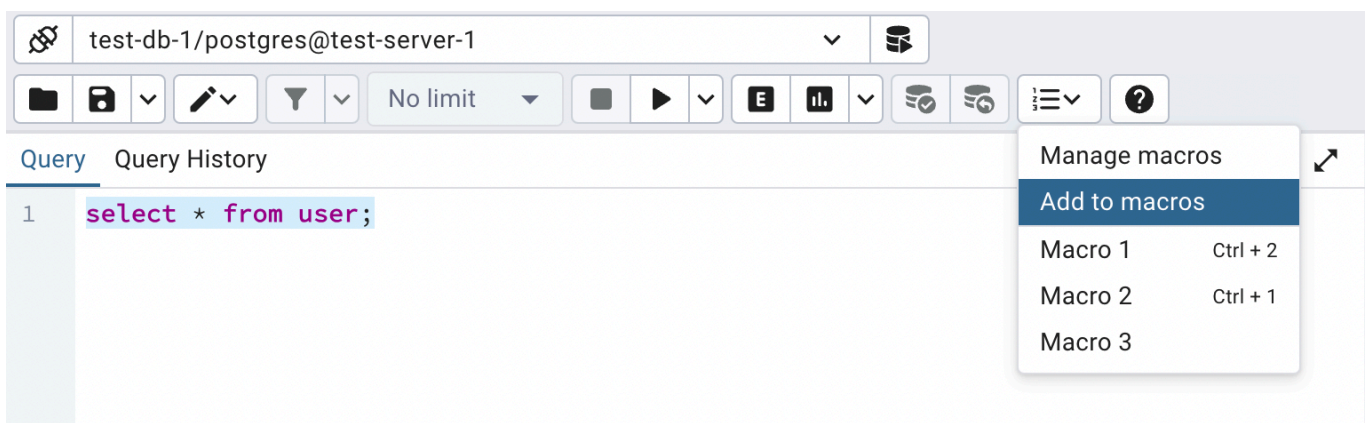
Query Tool Macros enable you to execute pre-defined SQL queries with a single key press. Pre-defined queries can contain the placeholder \$SELECTION\$. Upon macro execution, the placeholder will be replaced with the currently selected text in the Query Editor pane of the Query Tool.



To create a macro, select the *Manage Macros* option from the *Macros* menu on the *Query Tool*. Select the key you wish to use, enter the name of the macro, and the query, optionally including the selection placeholder, and then click the *Save* button to store the macro.



To add a query to macros, write and select your query, then go to the *Macros* menu in the Query Tool and click *Add to macros*. Your query will be automatically saved to macros.



To delete a macro, select the macro on the *Manage Macros* dialogue, and then click the *Delete* button. The server will prompt you for confirmation to delete the macro.

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

[illegible]