A Getting Started PostgreSQL Tutorial

[Shivayan Mukherjee](https://www.sqlservercentral.com/author/shivayan007), 2020-09-28

Introduction to PostgreSQL

PostgreSQL is a free and general purpose open source object-relational database system that uses and extends the SQL language. Though originally designed to run on UNIX platforms, PostgreSQL is eligible to run on various platforms such as Linux, macOS, Solaris, and Windows. PostgreSQL databases provide enterprise-class database solutions and are used by a wide variety of enterprises across many industries, like financial services, information technology, government and media & communications.

PostgreSQL in addition to being free and open source is highly robust and extensible. For example, one can define his own data types, develop custom functions and conform to SQL standard without compromising on architectural designs. Almost all the features of SQL are supported though sometimes with a slightly different syntax. A few important features:

Has wide range of datatypes both out of the box and customized

Has full support for data integrity. All needed constraints like UNIQUE, NOT NULL, Primary Keys, Foreign Keys and Exclusions are supported

Uses powerful indexing techniques to deliver strong concurrency and performance

Data recovery mechanisms are also well supported in PostgreSQL thus making it reliable and secured to use

Has NoSQL-style JSON processing features, which makes it compatible to the modern day architecture and design patterns

Software Setup

We need to have PostgreSQL installed in our system.  This is [the link](https://www.postgresql.org/download/) which can be used to download the latest stable version. The package comes bundled with PostgreSQL server, pgAdmin4 (a GUI tool to manage the database) and a command line tool, psql.

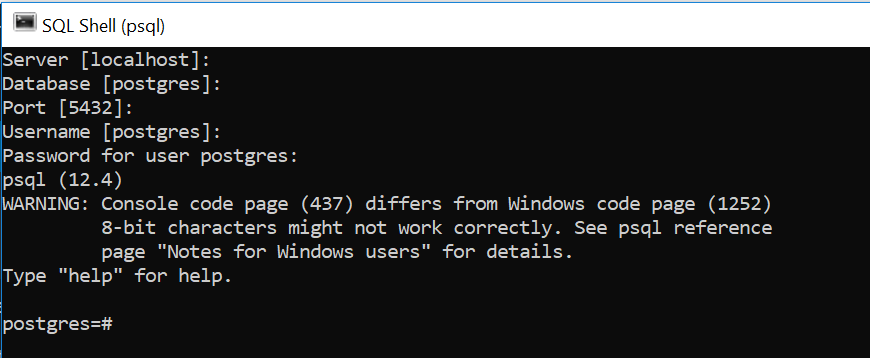
Typically there are two methods to execute PostgreSQL queries:

Using pgAdmin (a GUI Management tool)

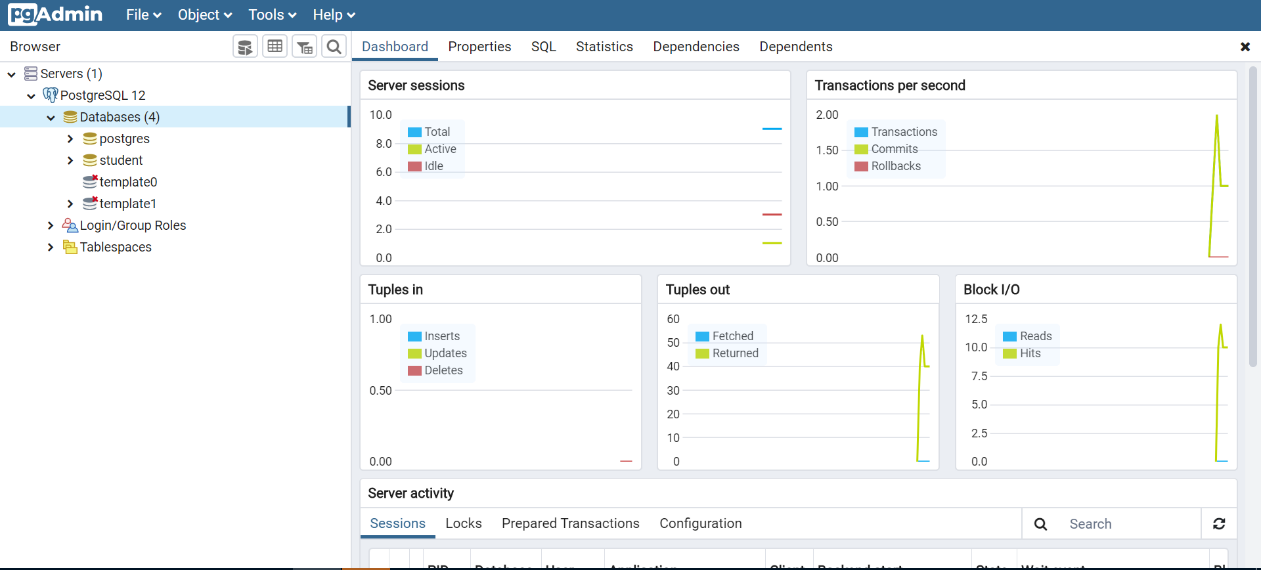
With the help of SQL Shell (psql).  This article covers both the techniques.

For Linux installation refer to this article, <https://www.sqlservercentral.com/articles/postgresql-jdbc-tutorial-on-linux>

On Windows, there is a little post installation work. To start a SQL Shell terminal, navigate to start->SQL Shell(psql) and enter the credentials set during installation.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f439d5910177.png)

To start pgAdmin4, navigate to start->pgAdmin4 and enter the 'postgres' user password set during installation.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f439fcad34b1.png)

PostgreSQL DDL Statements

There are a number of DDL statements you will need to work with PostgreSQL, such as creating, dropping, and working with a database.

Creating a Database with the SQL Shell

The following syntax shows how to create a database in PostgreSQL. Section also covers how to list and change databases post creation of a database using the psql shell. The 'CREATE DATABASE' statement is used to create a new database as shown below;

Syntax:

CREATE DATABASE database\_name;

Sample Code:

CREATE DATABASE studentDB;

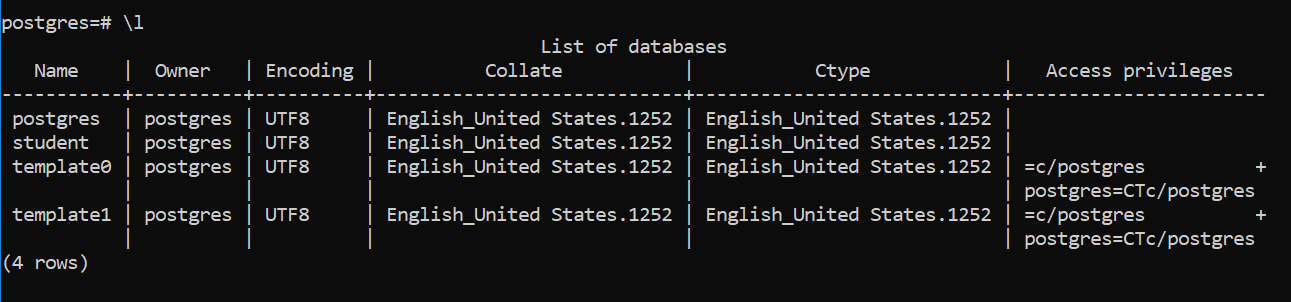
[https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a21853039-300x47.png](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a21853039.png)

Before jumping over to the next topic let us learn some psql commands to list and change databases;

To list databases, use \l.

postgres=# \l

You can see all the databases listed below.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f309d6ded5.png)

To change the database, use the \c command.

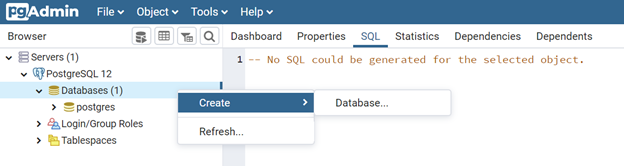
postgres=# \c database\_name;

postgres=# \c studentdb;

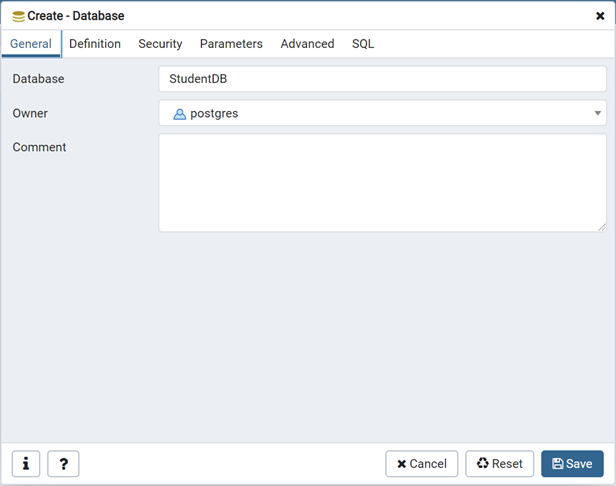
[https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a3edcdf5d.png](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a3edcdf5d.png)

Creating Database In PgAdmin

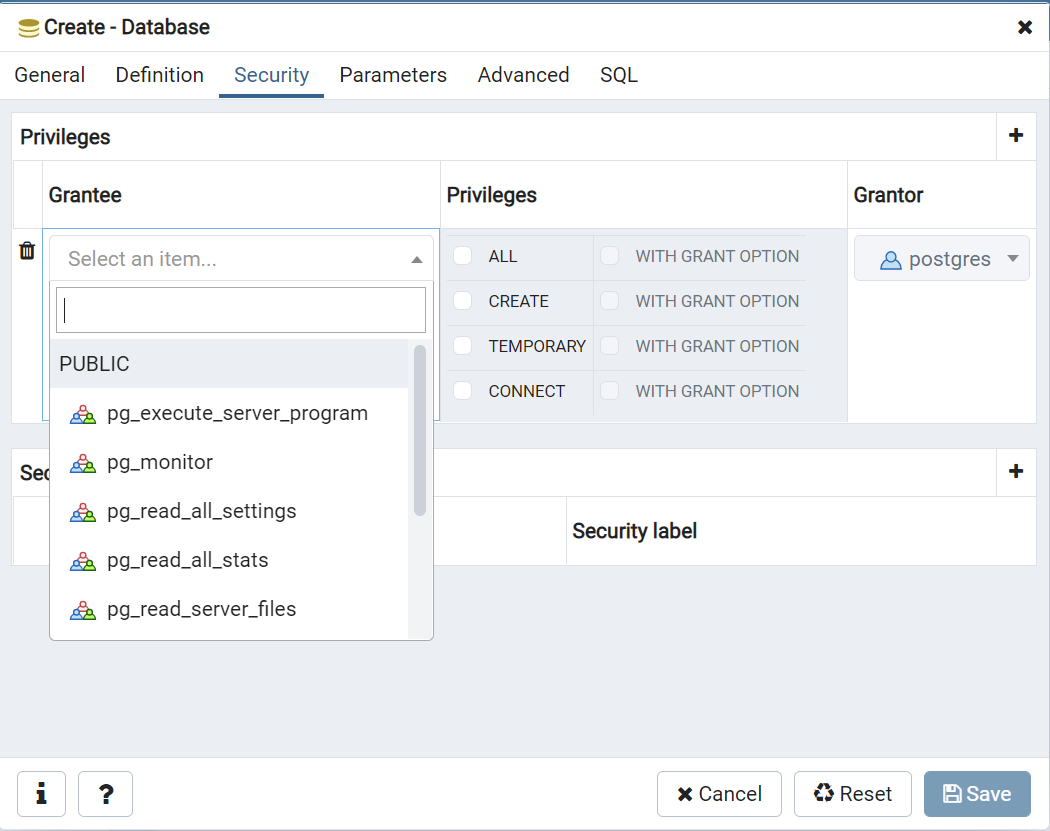
After starting PgAdmin, navigate to Databases->Create->Database and specify the details as shown below:

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a27989306.png)

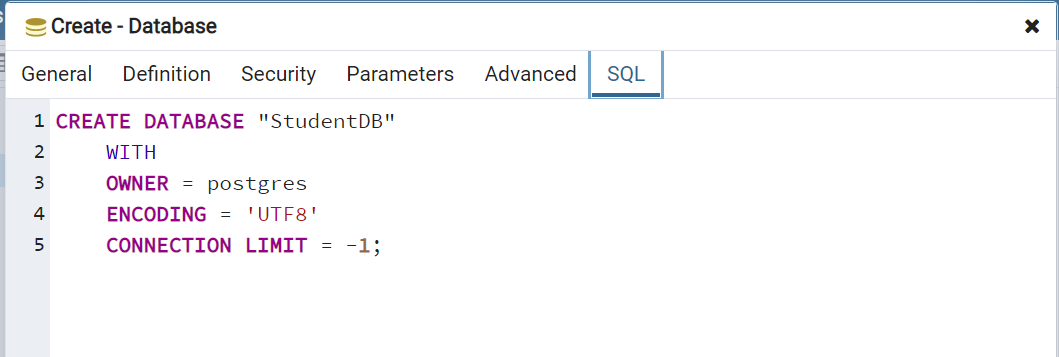
Specify the database name and comments if any. The comment is a note on the database object itself.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a32c5f750.png)

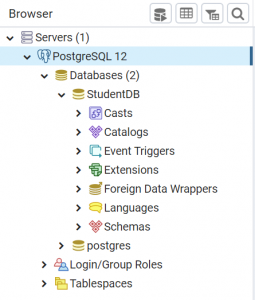
Grants/Permissions can be specified using the Security tab.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507a1396f5a.png)

Finally the complete SQL query can be viewed from the 'SQL' tab.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507aa3caba1.png)

Click Save to create the database. We can then see the database in our list of databases on the left side.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a3562cff0.png)

Dropping a Database with the SQL Shell

When a database is no longer required, it can be dropped or deleted using the 'DROP' command.

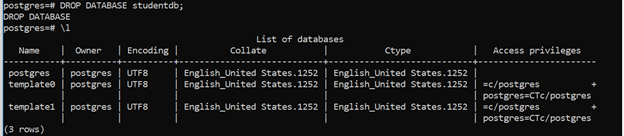
Syntax:

DROP DATABASE database\_name;

Sample Code:

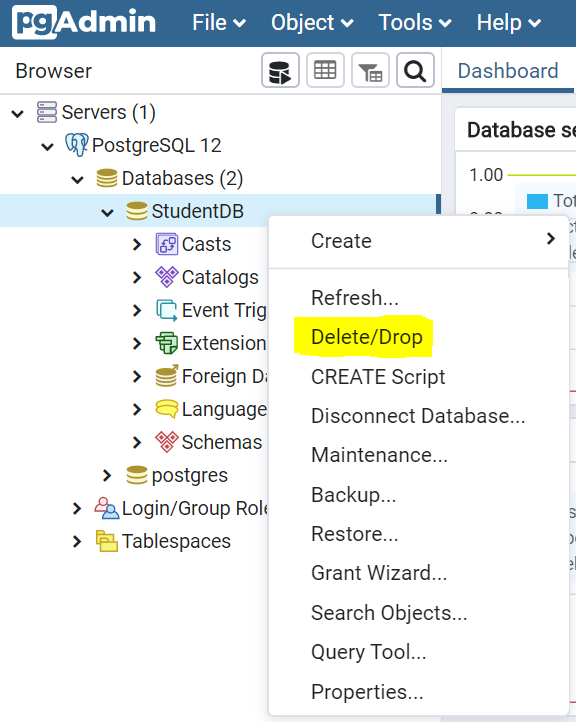
DROP DATABASE studentdb;

We can see the database has been removed.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a59f98b83.png)

Drop Database In PgAdmin

In Pgadmin, right click the database. Then navigate to Databases -> Target Database -> Delete/Drop.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a5e2787d9.png)

PostgreSQL Schema Commands

These are some of the common commands that you can use to work with schemas in PostgreSQL.

Creating a Schema

This command is used to create a new schema in an existing database.

Syntax:

CREATE SCHEMA schema\_name;

Sample Code:

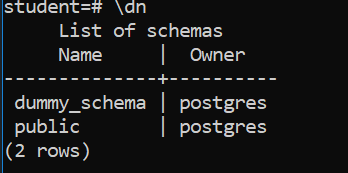
CREATE SCHEMA dummy\_schema;

[https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f448d6b34747.png](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f448d6b34747.png)

Before jumping over to the next topic, let us learn some psql commands to list, view, and change schemas.

List Schema

postgres=# \dn

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f36e2b1f07.png)List all schemas

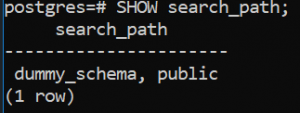
Change Schema

postgres=# SET search\_path TO dummy\_schema,public;

[https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f3822ad2e0.png](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f3822ad2e0.png)

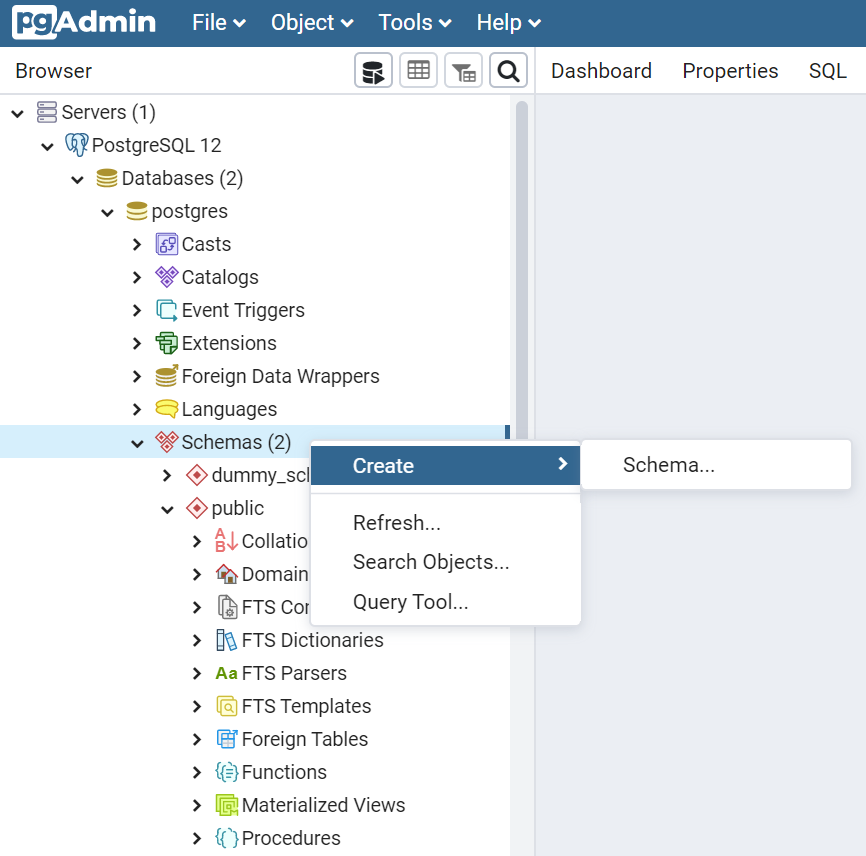
Show Current Schema

postgres=# SHOW search\_path;

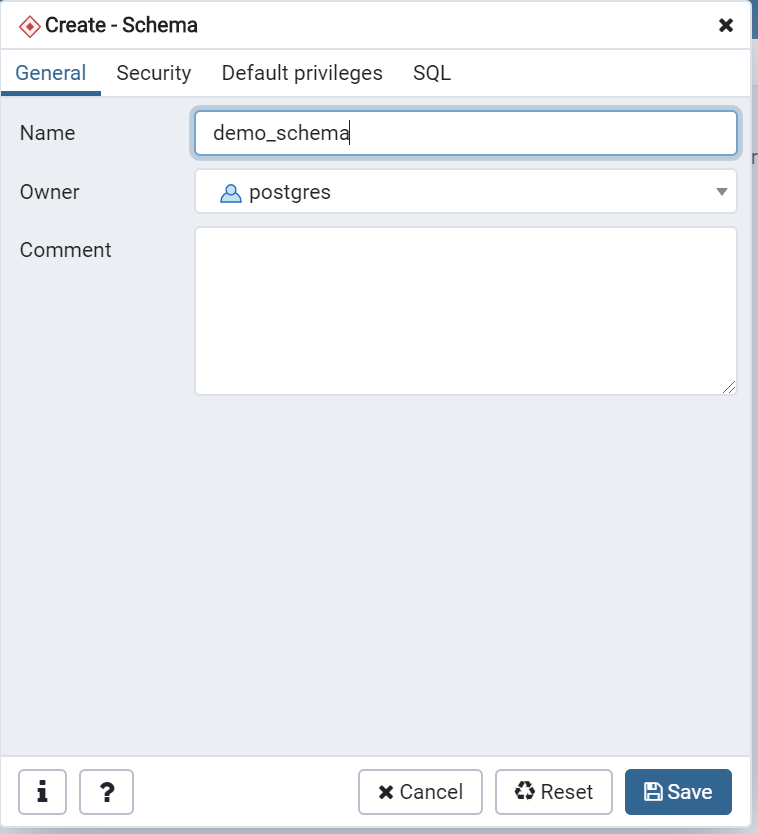
[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f4490297ccb6.png)Displays current Schema

Create Schema In PgAdmin

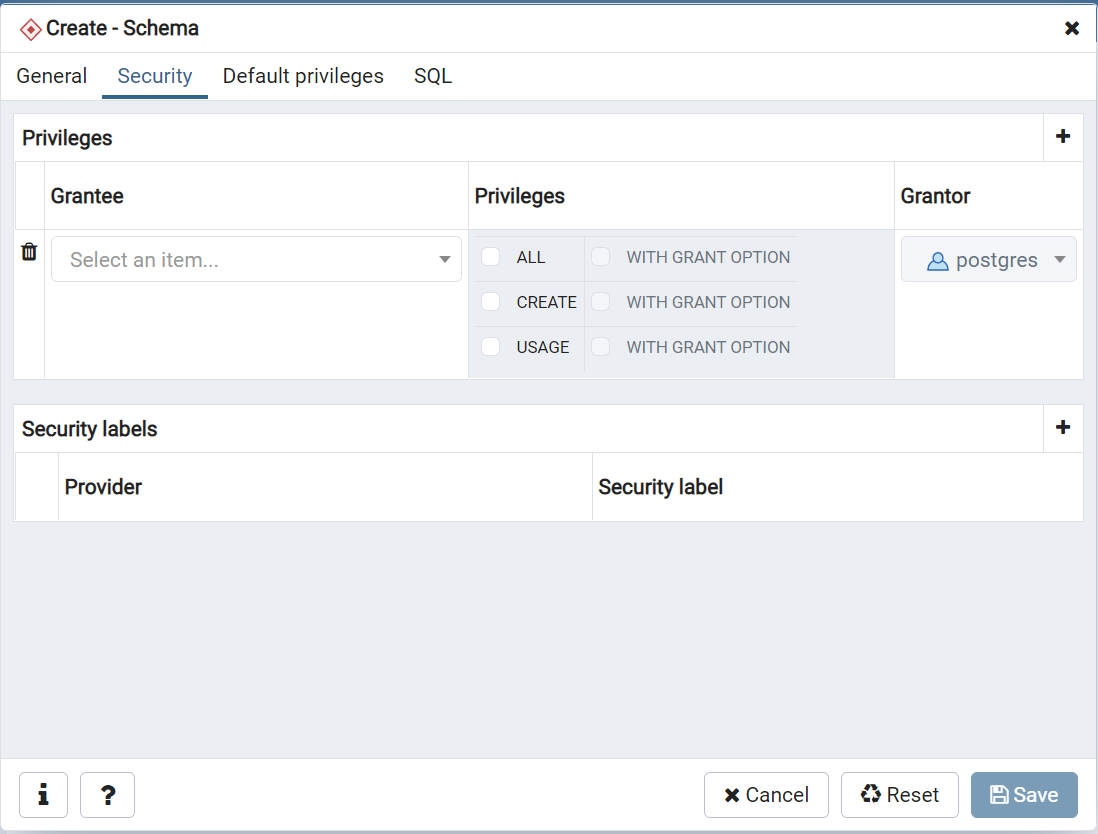
In PgAdmin, right click the Schemas entry. Then navigate to Databases-> Target Database -> Schemas -> Create -> Schema

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f44908c590f9.png)

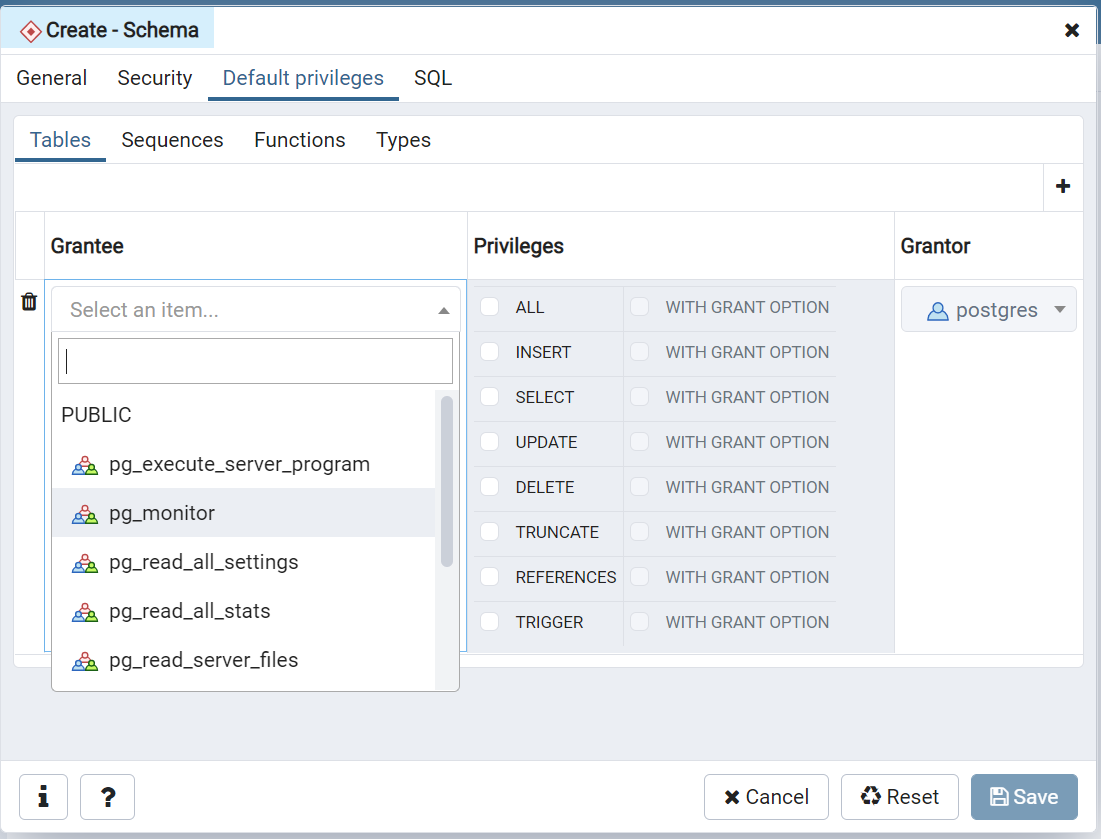
We can enter a name and comment.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f44931fe696d.png)

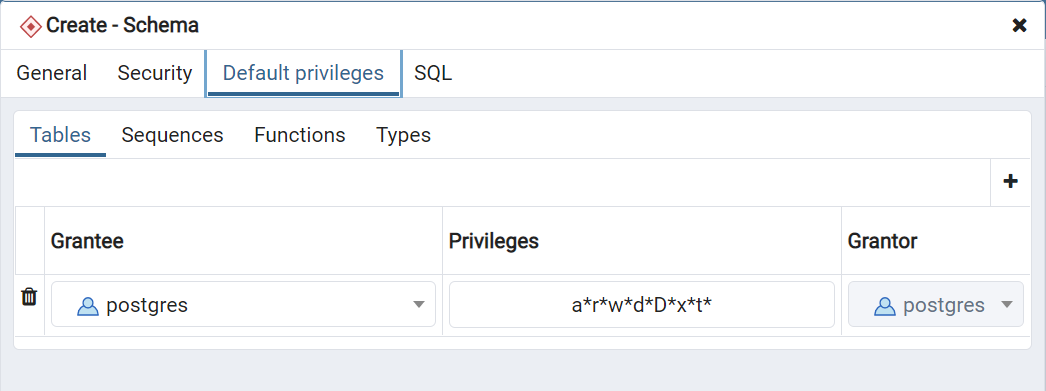
Grants/Permissions can be specified from the Security tab

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507b847439d.png)

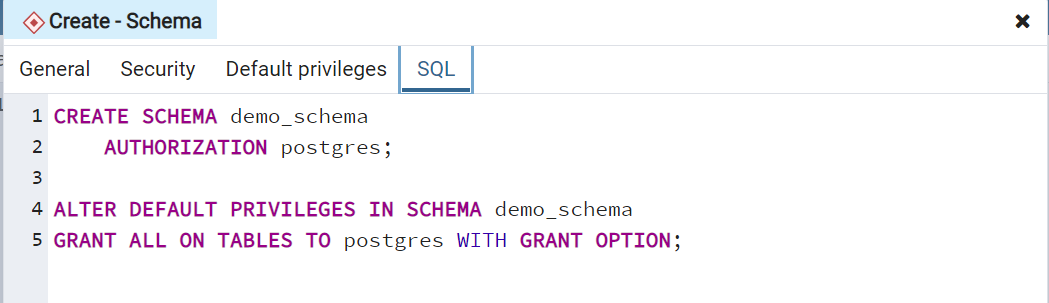
Default privileges which are applied globally to the entire database can be set from the 'Default Privileges' tab

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507ce2e10d6.png)

Here the user, postgres, is given all rights(insert,select,delete,update,truncate,references,trigger) with GRANT option

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507da5d8a6e.png)

Finally the complete SQL query can be viewed from the 'SQL' tab.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507e4483d46.png)

Click Save to create the schema.

Alter Schema in SQL Shell

The ALTER SCHEMA command is used to update an existing schema definition. Using this command it is possible to rename the schema or change its owner.

Example 1 - Syntax to alter schema name:

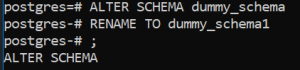
ALTER SCHEMA schema\_name

RENAME TO new\_name;

Sample code:

ALTER SCHEMA dummy\_schema

RENAME TO dummy\_schema1;

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f44955460f5e.png)

Example 2 - Syntax change alter the schema owner:

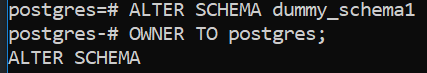
ALTER SCHEMA Schema1

OWNER TO new\_owner\_name;

Sample code:

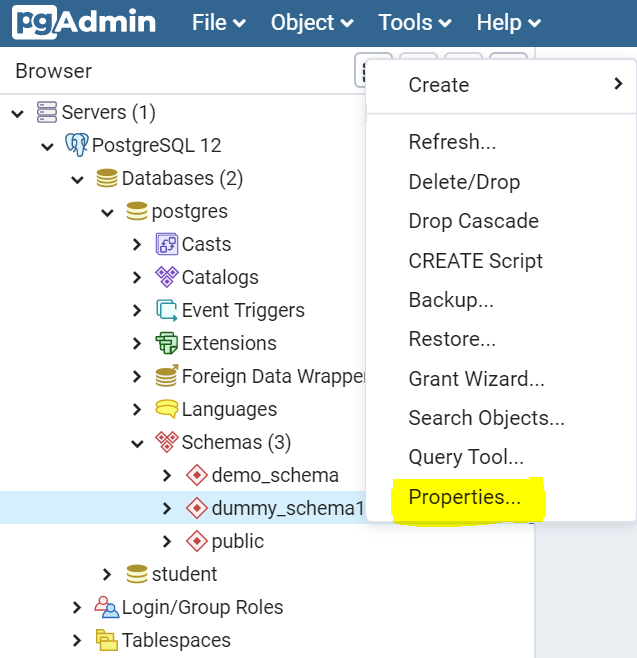
ALTER SCHEMA dummy\_schema1

OWNER TO postgres;

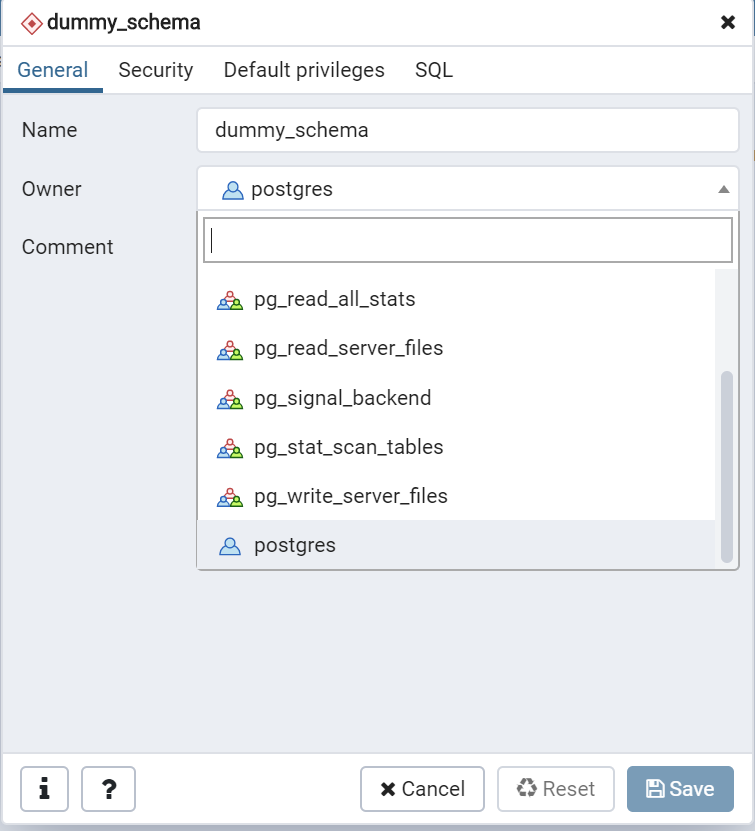
[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f4495bac6792.png)

Alter Schema In PgAdmin

In PgAdmin, right click the schema you wish to alter. Navigate to Databases -> Target Database -> Target Schema -> Properties

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f44972964191.png)

You can then pick a new owner for the schema. Click Save to make the change.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f3a9b95aea.png)

Drop Schema in SQL Shell

The 'DROP SCHEMA' statement is used to drop a schema when it is no longer needed. A schema can be either be dropped alone or it along with its dependent objects. When it is dropped along with its objects it is called a CASCADE DROP.

Syntax to drop a schema:

DROP SCHEMA schema\_name;

Sample code:

DROP SCHEMA dummy\_schema1;

[https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f4498f02338a-300x38.png](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f4498f02338a.png)

CASCADE DELETE - Dropping a schema in the above fashion may sometimes result in an error due to its dependent objects. In this scenario the CASCADE command is used.

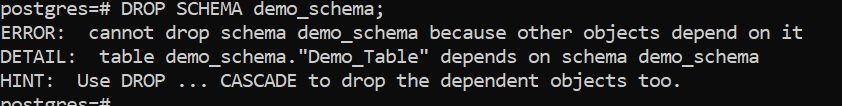
Syntax:

DROP SCHEMA schema\_name CASCADE;

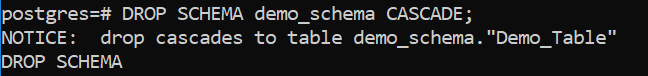
Sample code:

DROP SCHEMA demo\_schema CASCADE;

The image below shows the error about the presence of dependent objects.

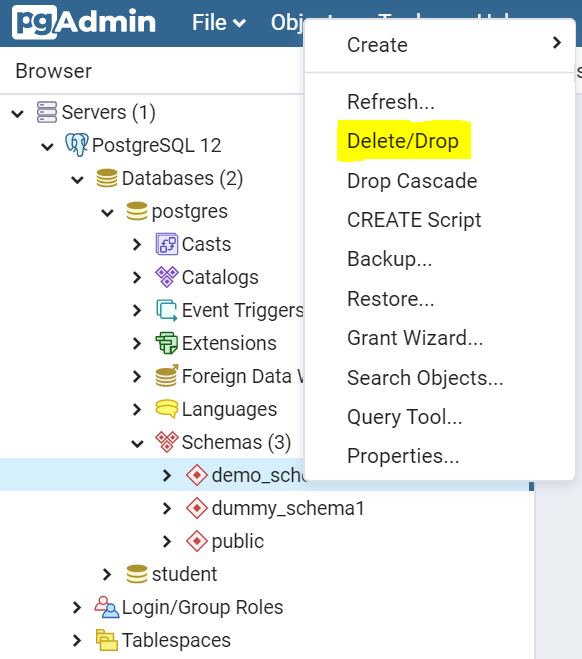
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f3d413a5b4.png)

If the CASCADE option is used, the schema is dropped.

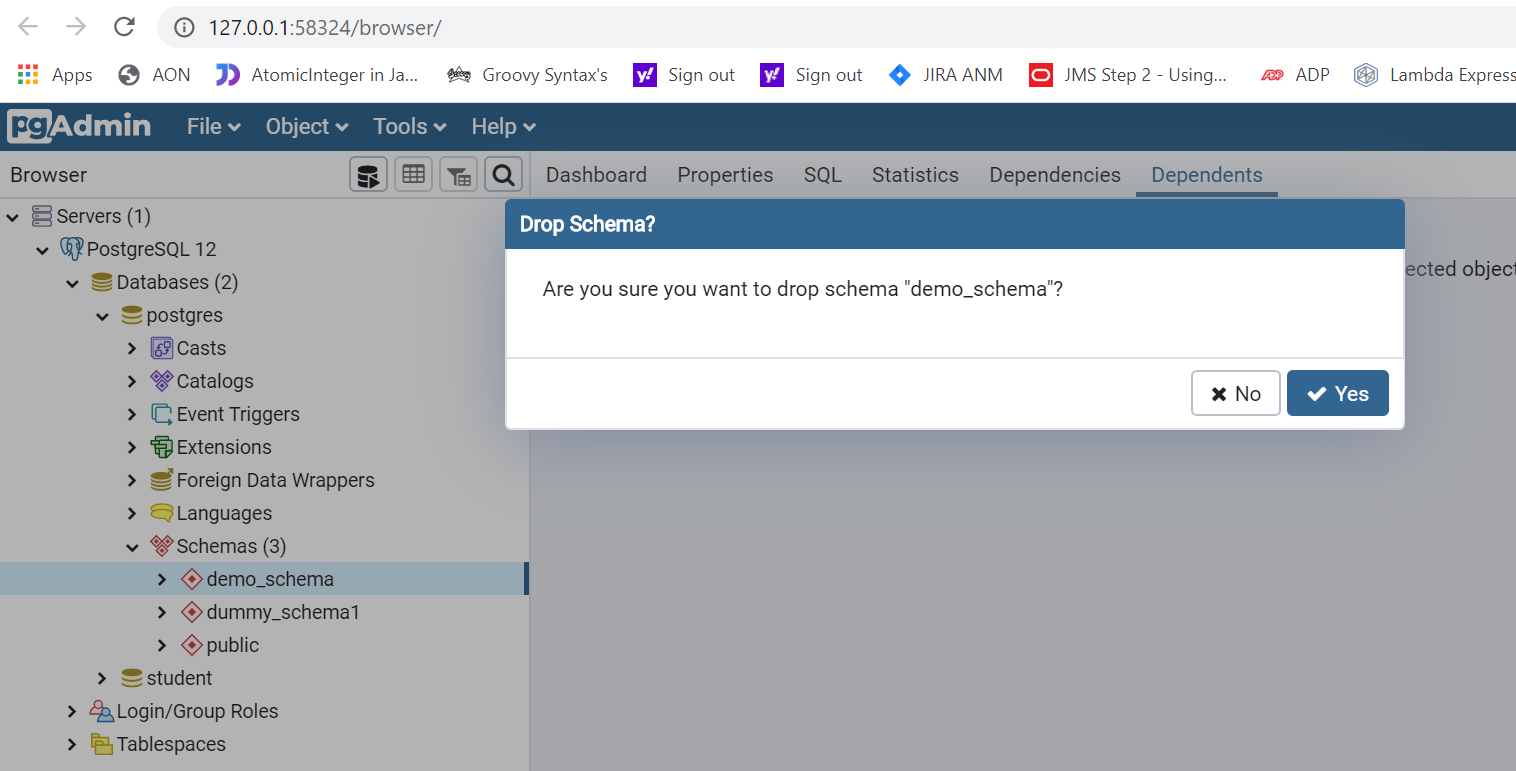
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f3db3e9152.png)

Drop Schema In PgAdmin

In PgAdmin, right click a schema and navigate to Databases -> Target Database -> Schemas -> Target Schema -> Delete/Drop.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f4499f78ba10.png)

Before you can drop a schema, you need to confirm the action.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f449ab57dd7a.png)

PostgreSQL Table Commands

These are the common commands for working with tables in PostgreSQL.

Creating a Table in SQL Shell

A table, as we know, is the core of any relational database. The following section illustrates how we can create a table in PostgreSQL.

Syntax:

CREATE TABLE table\_name

(

COLUMN\_NAME1 datatype,

COLUMN\_NAME2 datatype,

........

COLUMN\_NAMEN datatype,

);

Sample code:

CREATE TABLE student\_info

(

"STUDENT\_NAME" character varying(50) NOT NULL,

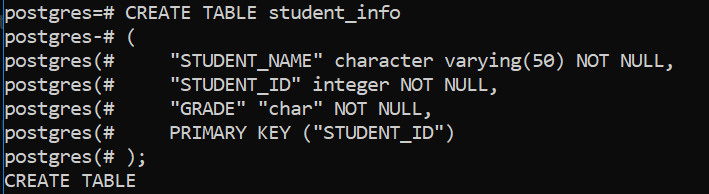
"STUDENT\_ID" integer NOT NULL,

"GRADE" "char" NOT NULL,

PRIMARY KEY ("STUDENT\_ID")

);

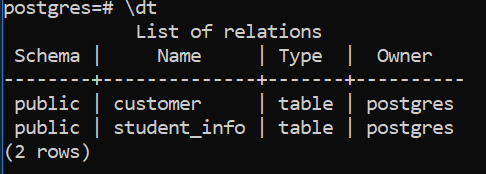
We get a confirmation that the table is created.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a81a36dda.png)

Let us now learn how to list tables in a schema. By default only the tables under the public schema are listed.

postgres=# \dt;

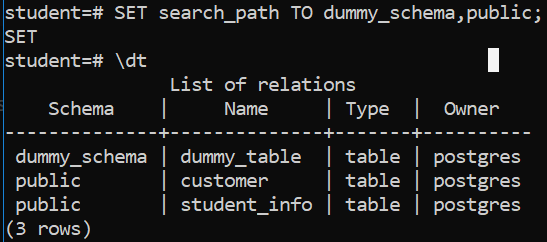
We can then see a list of the tables.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a8b2d802d.png)

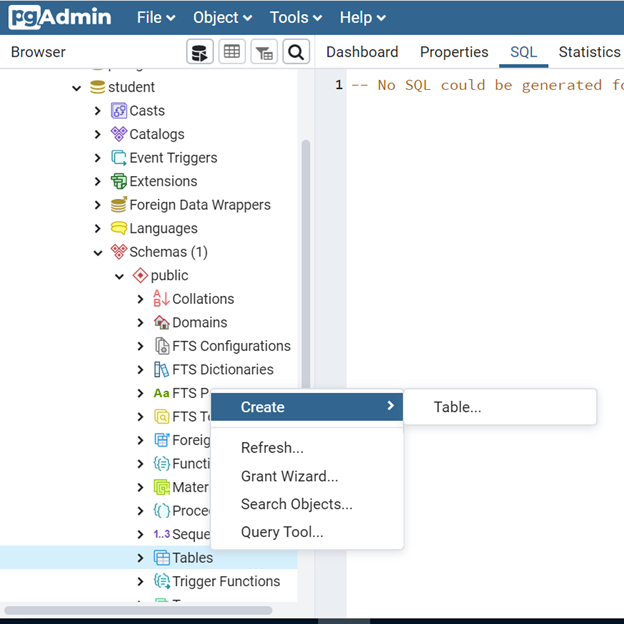
To list tables under any other schema, that particular schema needs to be set in the search path, as shown below. Using this command one or more schemas can be set to the search path. As we have set dummy\_schema and public, all their corresponding tables are being returned

SET search\_path TO dummy\_schema,public;

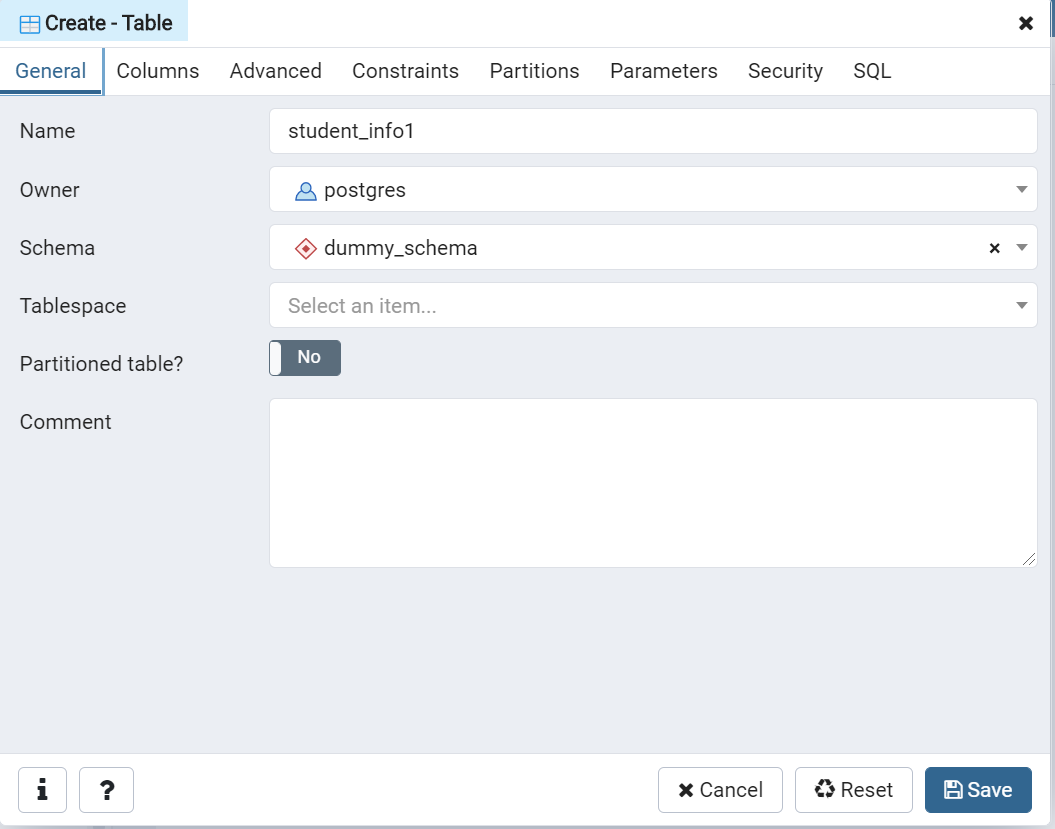
We see the tables with their schema listed.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f42b9a5e72.png)

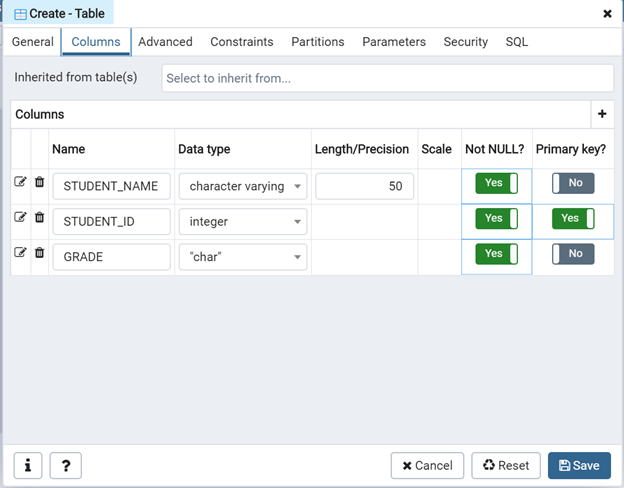
Create Table In PgAdmin

Right click the Tables item under a schema, and navigate to Databases-> Target Database-> Schemas-> Target Schema->Tables-  >Create->Table.[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43a9233344a.png)

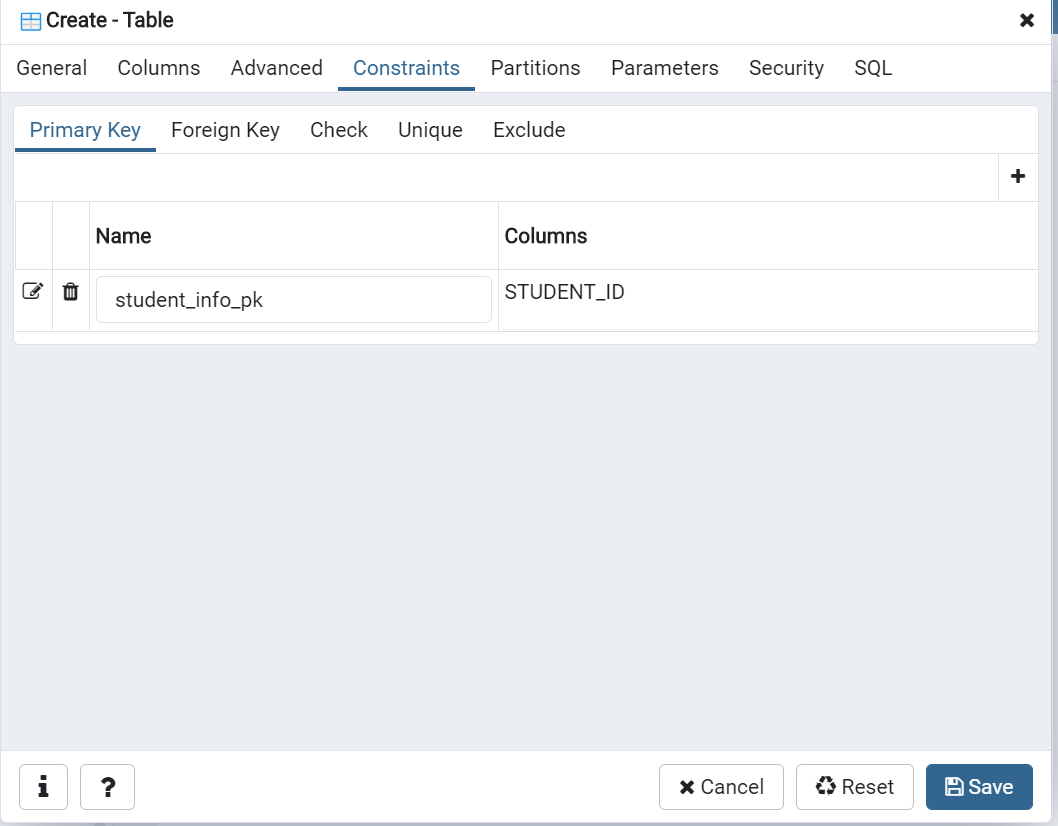
Creates a new table within 'public' schema of 'Student' database.We can fill in the name and owner for the table. Here, the owner refers to the user 'postgres'. The schema is 'public' by default. Any schema can be used as per the requirement.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f50775f16cb3.png)

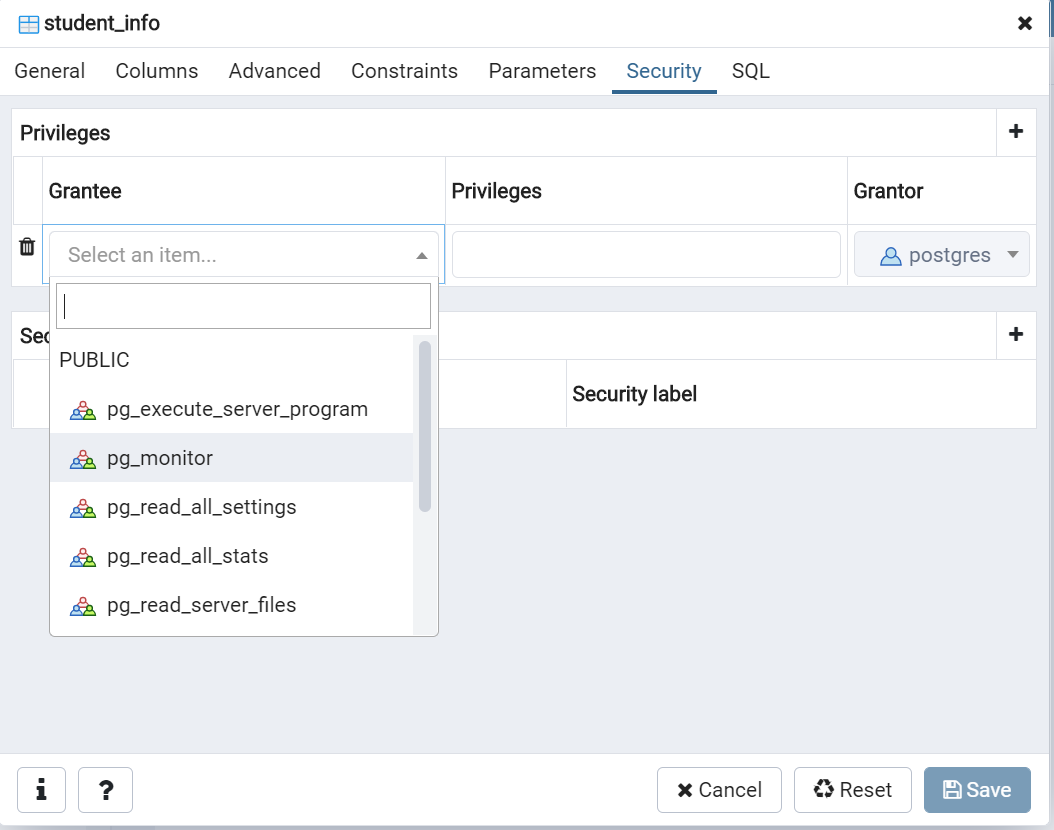
The second tab is where you specify the columns, their data types and precision/scale.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43aa35722fe.png)

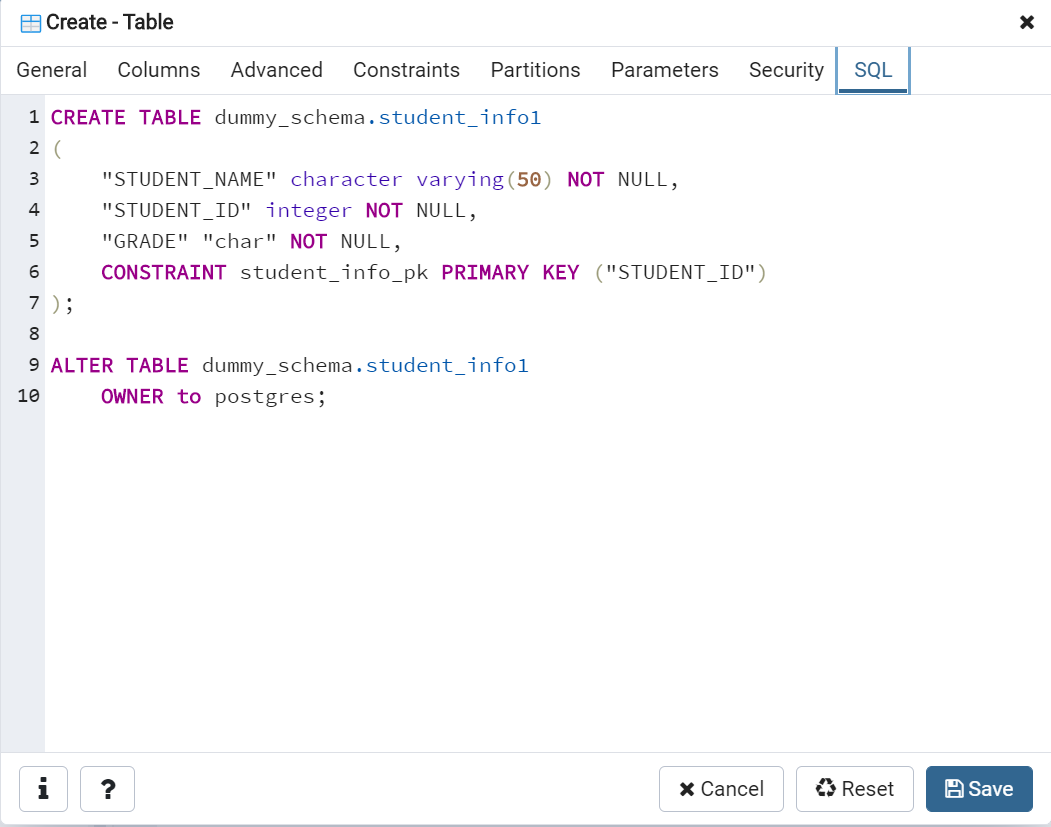
The Constraints tab is used to specify any constraint like Primary Key, Foreign Key, Unique, etc. Since we declared the column STUDENT\_ID as the primary key in the previous step, a primary key representing the same is automatically generated on saving. Other constraints if required can be added .

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5077b370f7b.png)

Grants and permissions can be specified using the Security tab.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f50763d8d00e.png)

Finally the complete SQL query can be viewed from the 'SQL' tab

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f507818d429f.png)

Click Save to create the table.

The  features available in the other tabs Advanced, Partitions are not very common and seldom used, except under very specific business requirements.

Alter a Table in SQL Shell

The 'ALTER  TABLE' statement is used to add, delete, modify columns in an existing table.

Syntax:

ALTER TABLE table\_name

ADD column\_name datatype;

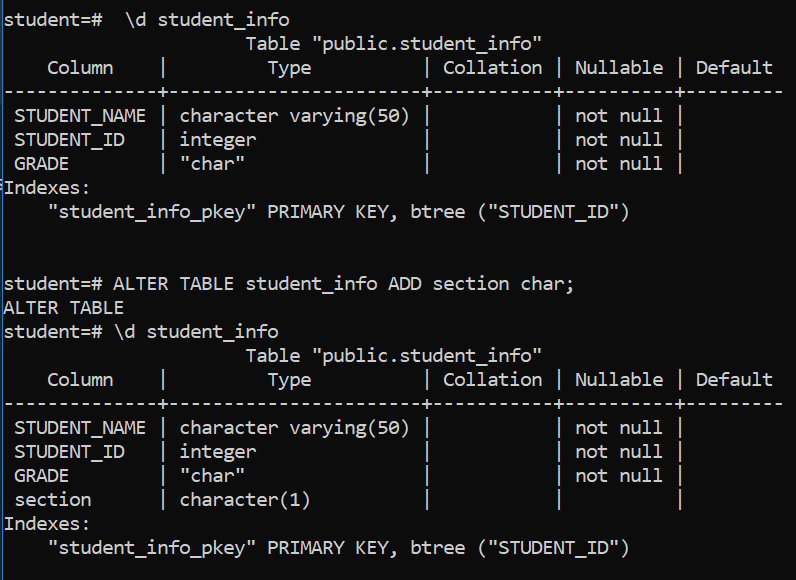
Sample code:

ALTER TABLE student\_info ADD section char;

We get a note that table has been altered.

[https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f4780abd9f.png](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f4780abd9f.png)

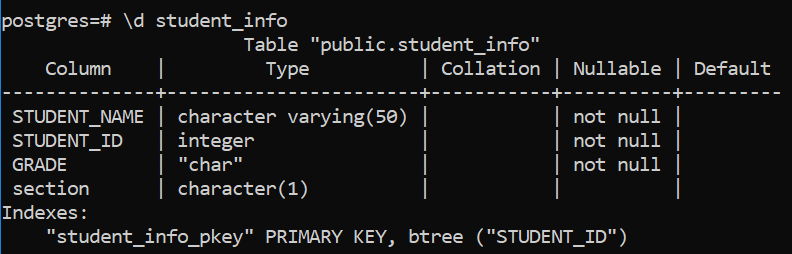
We can then display the table info. Notice the before and after table data in the image below.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f47f83ba8a.png)

Describing a Table in SQL Shell

In PostgreSQL there is no 'describe' keyword to find the definition of a table. There are two ways to do it. In the psql terminal, we can type the table name with a \d.

postres=# \d student\_info

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43ad96347a2.png)

 In pgAdmin, we can use a SQL query:

SELECT

column\_name,

data\_type

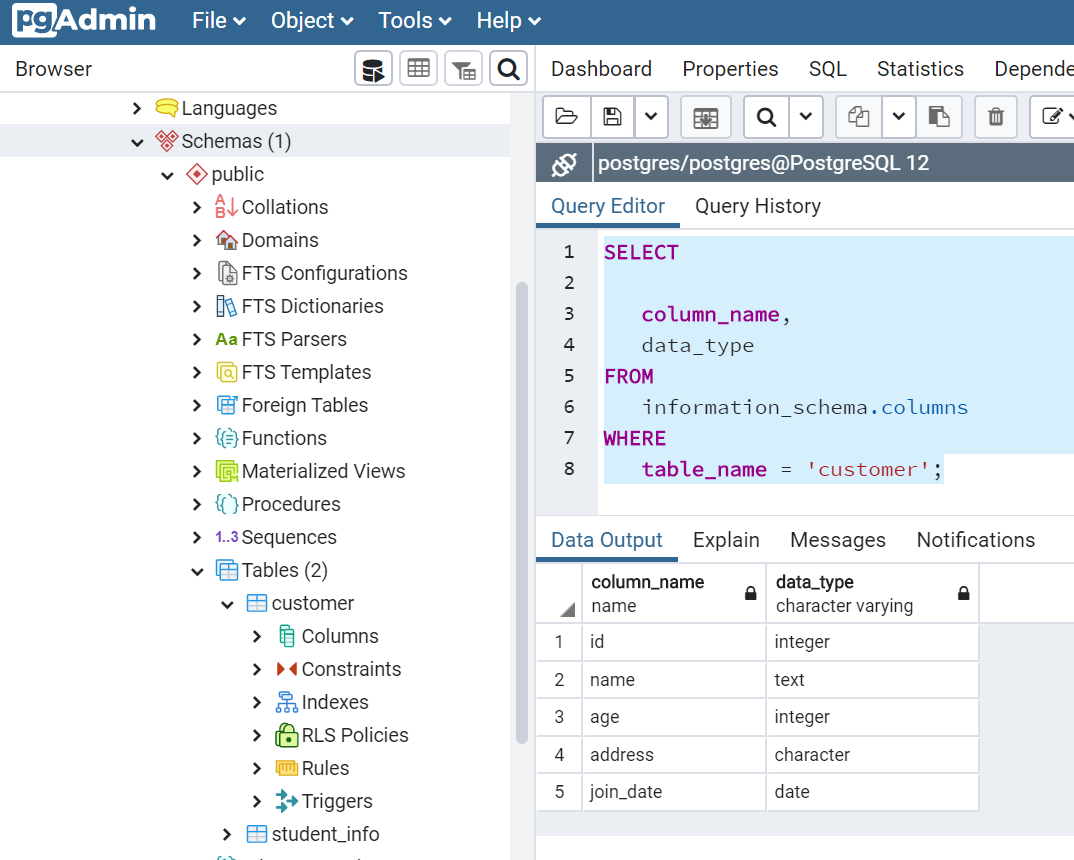
FROM

information\_schema.columns

WHERE

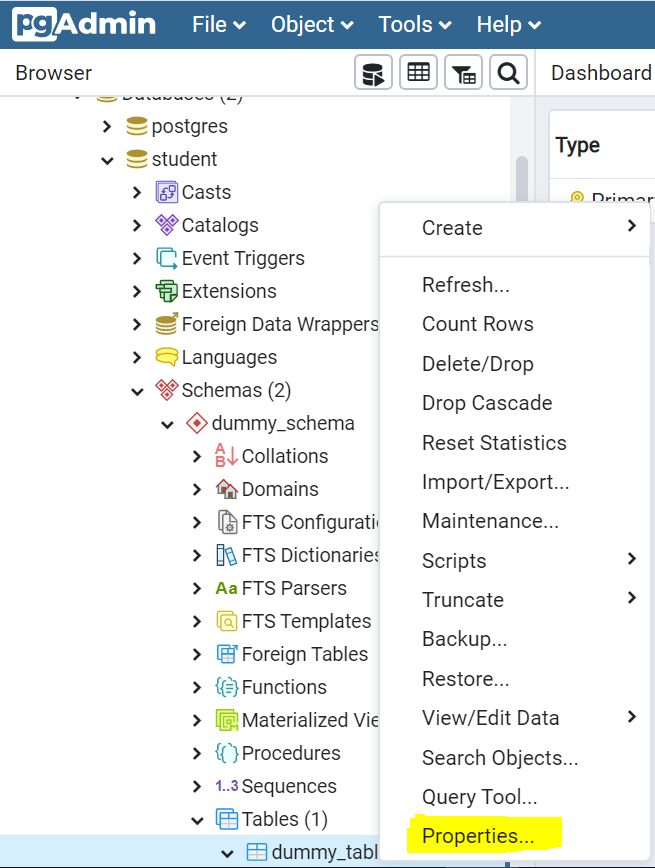
table\_name = 'customer';

Please note 'information\_schema' is a system defined schema where information pertaining to all tables is stored. You can see the results below.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43ae0970966.png)

Altering a Table In PgAdmin

Right click the table, and navigate to Databases-> Target Database-> Schemas-> Target Schema->Tables-  >Target Table->Properties.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f70a1740de.png)

Dropping a Table

The 'DROP TABLE' statement is used to drop a table when it is no longer needed.

Syntax:

DROP TABLE table\_name;

Sample code:

DROP TABLE student\_info;

This drops the table.

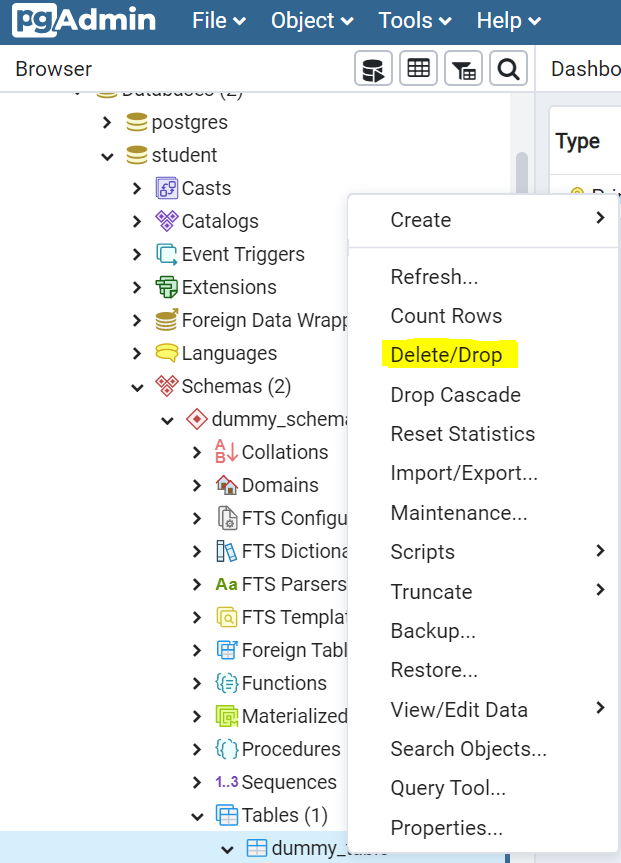
[https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f48e4dccb0.png](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f48e4dccb0.png)

We can confirm the deletion of the table.

[https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43af2ad6d1f.png](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43af2ad6d1f.png)

Drop Table In PgAdmin

Choose the table in PgAdmin, and navigate to Databases->Target Database->Schemas->Target Schema->Tables->Target Table->Delete/Drop.

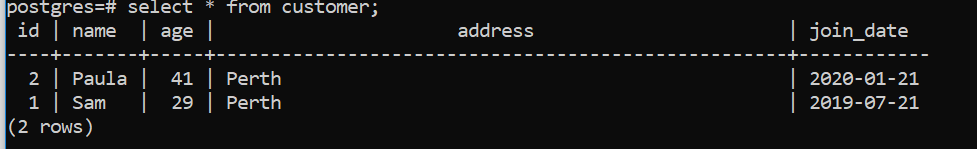
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f49efaeb30.png)

Truncate Vs Truncate Cascade Table

TRUNCATE statement is used to delete all rows/data in a table keeping the original table structure intact, is an option when table has no dependent objects.

TRUNCATE CASCADE is used to empty a table along with its dependent objects. For example a primary key-foreign key relationship exists between two tables. In this scenario if we try to execute TRUNCATE on the master table, system will throw an error because it's primary key is referenced as a foreign key in some other table.

TRUNCATE in PSQL shell:

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43b051660ea.png)Before Truncate

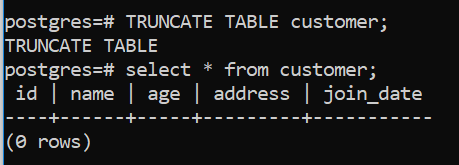
Syntax:

TRUNCATE TABLE table\_name;

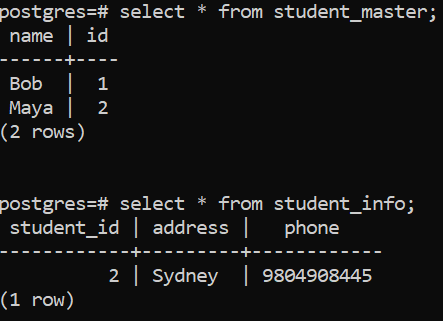
Sample code:

TRUNCATE TABLE customer;

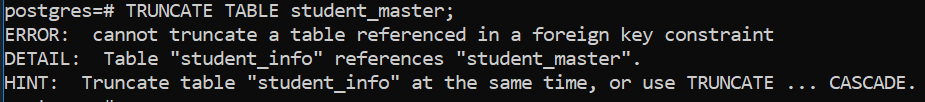
After the command, we see all data is deleted.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/08/img_5f43b0cede787.png)

TRUNCATE CASCADE in PSQL shell: Here is our table with rows i nit.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5093783e550.png)

On executing Truncate on the student\_master, system reports an error as explained above.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5093cb35088.png)

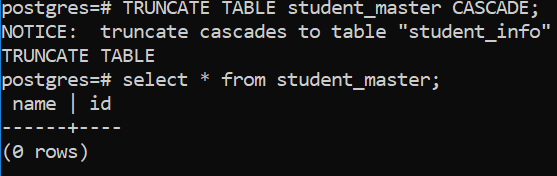
Syntax:

TRUNCATE TABLE table\_name CASCADE;

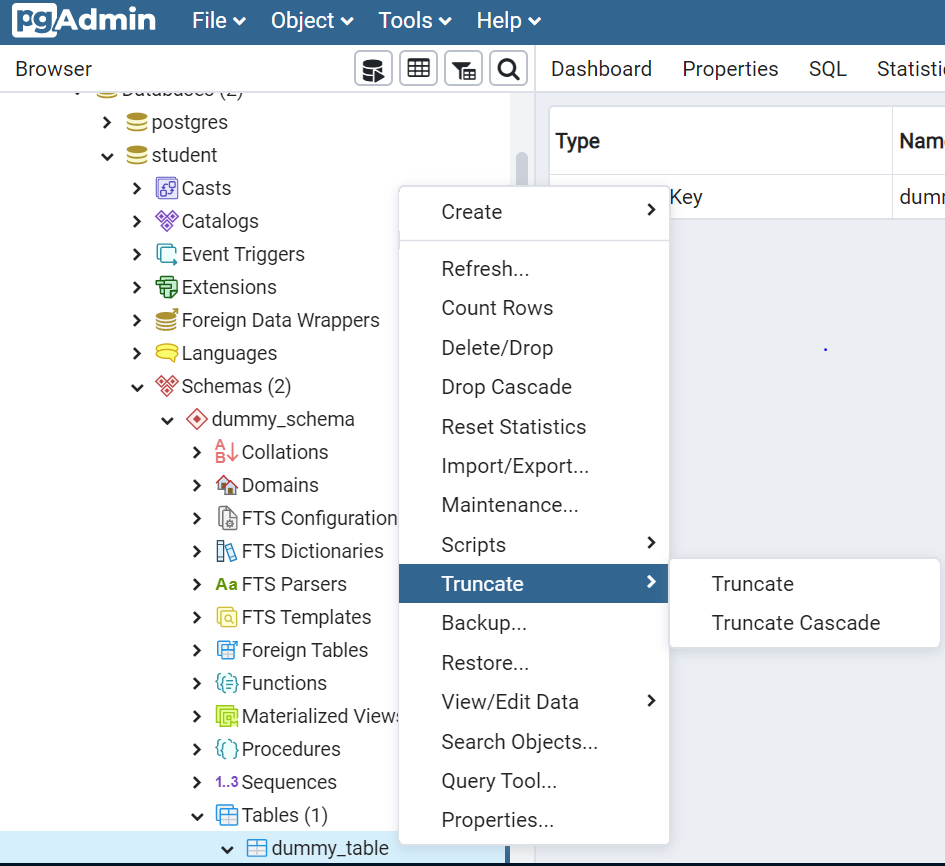
Sample code:

TRUNCATE TABLE student\_master CASCADE;

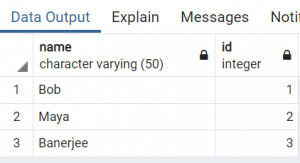
This is the table after truncating.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5094599f1f8.png)

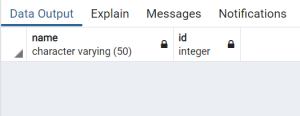
For TRUNCATE in PgAdmin: Pick the table in PgAdmin, and navigate to Databases->Target Database->Schemas->Target Schema->Tables->Target Table->Truncate.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f4f6fb1d1607.png)

We can see the table before truncating it.

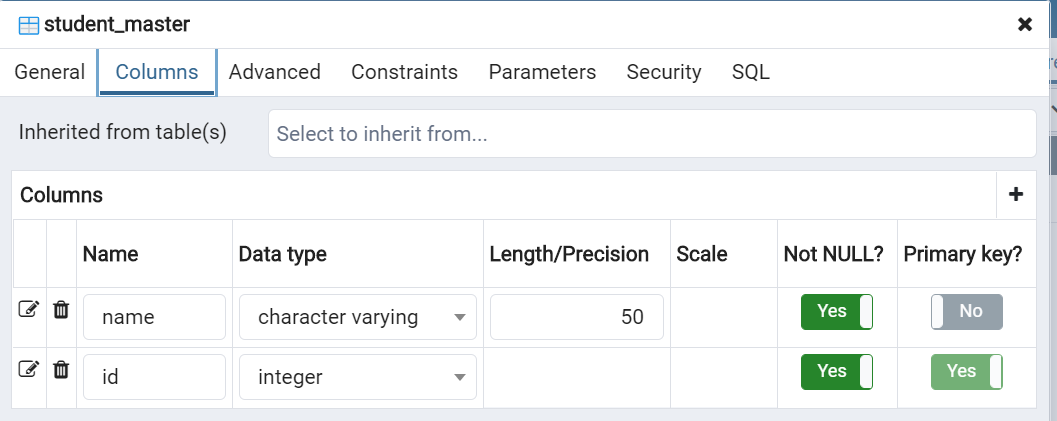
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5085a977135.png)

After truncate, there is no data.

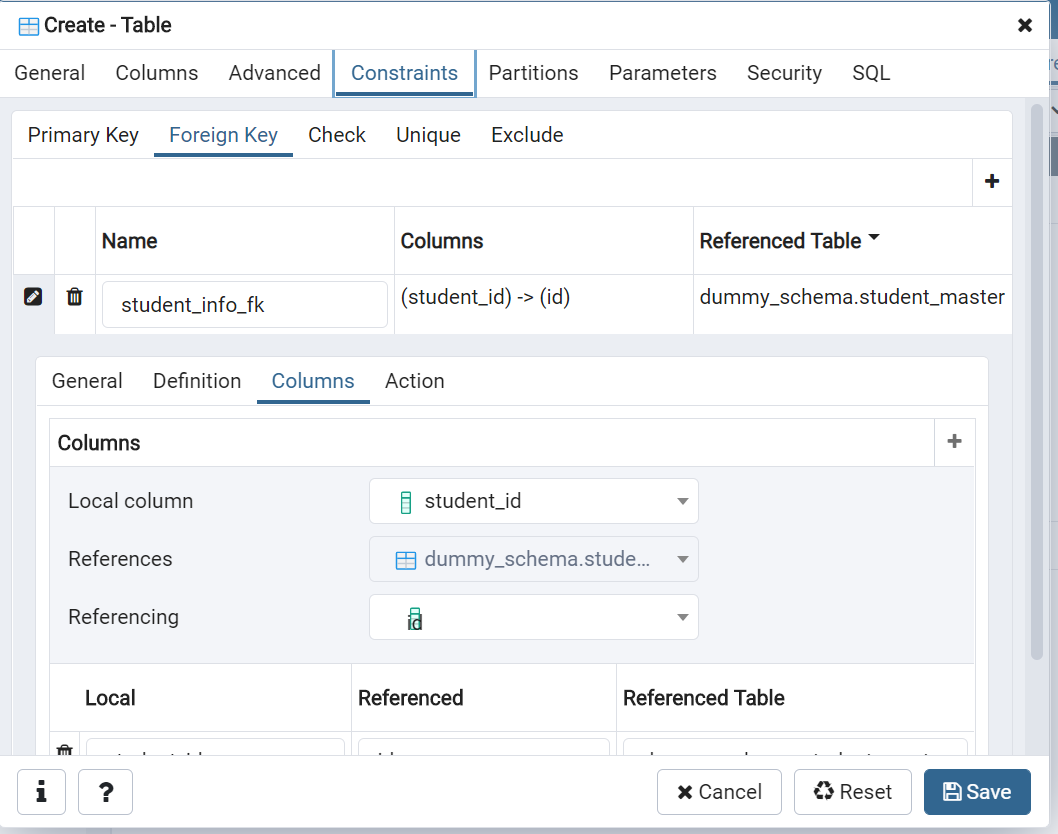
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5085f7ddc43.png)

TRUNCATE CASCADE in PgAdmin: Pick the table in PgAdmin, and navigate to Databases->Target Database->Schemas->Target Schema->Tables->Target Table->Truncate Cascade.

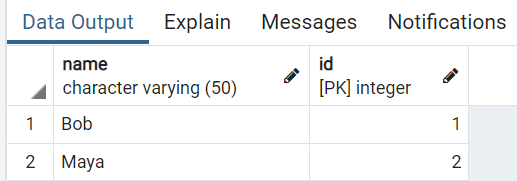
The parent table has 'id' as its primary key.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f508a27c01cf.png)

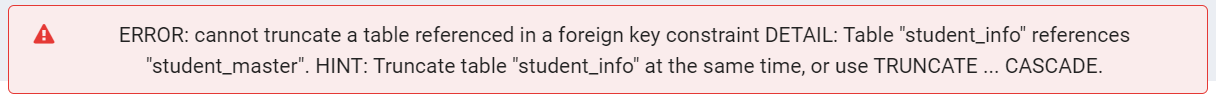
The child table whose column 'student\_id' is referenced as a foreign key with the parent table's primary key 'id'

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f50899baa650.png)

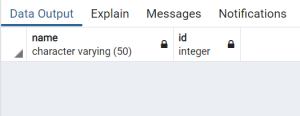
Data exists in both the tables before 'Truncate Cascade'

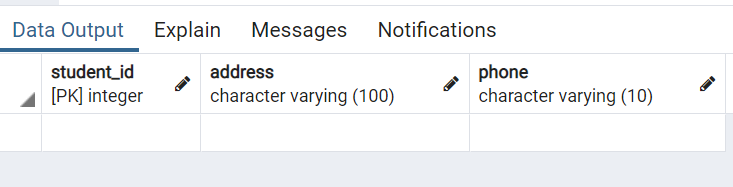
[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f508a7ce8618.png)student\_master[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f508be040f30.png) student\_info

On executing Truncate on the student\_master, the system reports an error as explained above.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f508c38e0c37.png)

Executing Truncate Cascade, both the parent and child tables are truncated.

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f5085f7ddc43.png)

[](https://www.sqlservercentral.com/wp-content/uploads/2020/09/img_5f508ca705e26.png)

Conclusion

This article gives an overview on the different aspects of DDL statements in PostgreSQL. We show how to construct a  relational database and add objects inside it with the help of schemas and tables. We hope this article will help you get started on your journey.