LOADING THE SAMPLE DATASETS - WINDOWS

Most exercises in this course use a sample database, **sqlda**, which contains fabricated data for a fictional electric vehicle company called **ZoomZoom**. Let's set it up by performing the following steps:

1. Open the command line and go to the location of your PostgreSQL **bin** folder using the **cd** command. Execute the command by pressing the **return** key.

```
cd "C:\Program Files\PostgreSQL\13\bin"
```

   Note that the path may be different according to wherever it was installed on your system. It will also depend on the version you have downloaded. This book was originally written and tested using version 10.9, but there are newer versions available now.

2. Now, let's create a database titled **sqlda**. Type or paste the following command. Then, press the **return** key to execute it.

```
createdb -U postgres sqlda
```

   You will be prompted to enter the password that you set for the **postgres** superuser during installation.

```
C:\Users\abhis\Documents\Packt Work\Applied SQL>createdb -U postgres sqlda
Password:
```

Figure 0.1: PostgreSQL shell password request

3. Next, type or paste the following command. Press the **return** key to execute it:

```
psql -U postgres
```

Enter the password you set when downloading and installing PostgreSQL. Then press the **return** key. You should be able to login to the PostgreSQL console:

```
C:\Users\abhis>psql -U postgres
Password for user postgres:
psql (12.3)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# _
```

Figure 0.2: PostgreSQL console login

4. Type **\l** (that's a backslash and a lowercase L) and then press the **return** key to check if the database has been created. The **sqlda** database should appear along with a list of the default databases.

```
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |      Collate       |       Ctype        |   Access privileges
-----------+----------+----------+--------------------+--------------------+-----------------------
postgres   | postgres | UTF8     | English_India.1252 | English_India.1252 |
sqlda      | postgres | UTF8     | English_India.1252 | English_India.1252 |
template0  | postgres | UTF8     | English_India.1252 | English_India.1252 | =c/postgres          +
           |          |          |                    |                    | postgres=CTc/postgres
template1  | postgres | UTF8     | English_India.1252 | English_India.1252 | =c/postgres          +
           |          |          |                    |                    | postgres=CTc/postgres
```

Figure 0.3: PostgreSQL list of databases

5. Type and the **\q** command to exit the PostgreSQL console.
6. Download the **data.dump** file from the **Datasets** folder in the GitHub repository of this course by clicking this link: https://github.com/TrainingByPackt/SQL-for-Data-Analytics/tree/master/Datasets. Modify the highlighted path in the following command based on where the file is located on your system. Type or paste the command into the command line and press the **return** key to execute it. You will be prompted for your password.

```
psql -U postgres -d sqlda -f C:\<path>\data.dump
```

You should get an output similar to the one that follows:

```
C:\Users\abhis\Documents\Packt Work\Applied SQL>psql -U postgres -d sqlda -f data.dump
Password for user postgres:
SET
SET
SET
SET
SET
 set_config
------------

(1 row)


SET
SET
SET
SET
CREATE EXTENSION
COMMENT
CREATE EXTENSION
COMMENT
CREATE TEXT SEARCH DICTIONARY
SET
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE MATERIALIZED VIEW
CREATE TABLE
CREATE MATERIALIZED VIEW
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
COPY 44533
COPY 0
COPY 50000
COPY 32
COPY 50000
COPY 20
COPY 418158
COPY 12
COPY 15412
COPY 37711
COPY 300
COPY 20
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
REVOKE
GRANT
REFRESH MATERIALIZED VIEW
REFRESH MATERIALIZED VIEW

C:\Users\abhis\Documents\Packt Work\Applied SQL>
```

Figure 0.4: PostgreSQL database import

7. Let's check if the database has been loaded correctly. Login to the PostgreSQL console again by typing in or pasting the following command. Press the **`return`** key to execute it.

```
psql -U postgres
```

8. In the shell, type and run the following command to connect to the **`sqlda`** database:

```
\c sqlda
```

Then type and run the following:

```
\dt
```

This command should list all the tables in the database, as follows:

```
C:\Users\abhis\Documents\Packt Work\Applied SQL>psql -U postgres
Password for user postgres:
psql (12.3)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c sqlda
You are now connected to database "sqlda" as user "postgres".
sqlda=# \dt
                  List of relations
 Schema |             Name              | Type  |  Owner
--------+-------------------------------+-------+----------
 public | closest_dealerships           | table | postgres
 public | countries                     | table | postgres
 public | customer_sales                | table | postgres
 public | customer_survey               | table | postgres
 public | customers                     | table | postgres
 public | dealerships                   | table | postgres
 public | emails                        | table | postgres
 public | products                      | table | postgres
 public | public_transportation_by_zip | table | postgres
 public | sales                         | table | postgres
 public | salespeople                   | table | postgres
 public | top_cities_data               | table | postgres
(12 rows)


sqlda=#
```

Figure 0.5: Validating that the database has been imported

9. Finally, type and run the **`\q`** command to exit the PostgreSQL console, then close the command line window.

LOADING THE SAMPLE DATASETS – LINUX

Most exercises in this course use a sample database, **`sqlda`**, which contains fabricated data for a fictional electric vehicle company called **ZoomZoom**. Let's set it up by performing the following steps:

1. Switch to the **`postgres`** user by typing in the following command in the terminal. Press the **`return`** key to execute it.

```
sudo su postgres
```

   You should see your shell change as follows:

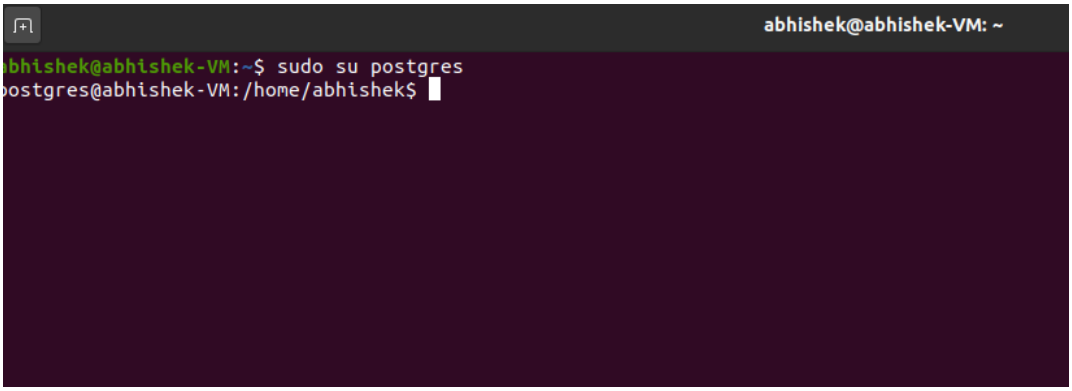

Figure 0.6: Loading the sample datasets on Linux

2. Type in or paste the following command to create a new database called **`sqlda`**. Press the **`return`** key to execute it.

```
createdb sqlda
```

You can then type the **psql** command to enter the PostgreSQL shell, followed by **\l** (that's a backslash followed by lowercase L) to check if the database was successfully created.

```
postgres@abhishek-VM:/home/abhishek$ createdb sqlda
postgres@abhishek-VM:/home/abhishek$ psql
psql (12.2 (Ubuntu 12.2-4))
Type "help" for help.

postgres=# \l
                              List of databases
   Name    |  Owner   | Encoding | Collate | Ctype |   Access privileges
-----------+----------+----------+---------+-------+-----------------------
 postgres  | postgres | UTF8     | en_IN   | en_IN |
 sqlda     | postgres | UTF8     | en_IN   | en_IN |
 template0 | postgres | UTF8     | en_IN   | en_IN | =c/postgres          +
           |          |          |         |       | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_IN   | en_IN | =c/postgres          +
           |          |          |         |       | postgres=CTc/postgres
(4 rows)

postgres=#
```

Figure 0.7: Accessing the PostgreSQL shell on Linux

Enter **\q** and then press the **return** key to quit the PostgreSQL shell.

3. Download the **data.dump** file from the **Datasets** folder in the GitHub repository of this course by clicking this link: https://github.com/TrainingByPackt/SQL-for-Data-Analytics/tree/master/Datasets. Navigate to the folder where you have downloaded the file using the **cd** command. Then, type the following command:

```
psql -d sqlda data.dump
```

4. Then wait for the dataset to be imported:

```
postgres@abhishek-VM:/home/abhishek/Downloads$ psql -U postgres -d sqlda < data.dump
SET
SET
SET
SET
SET
 set_config
------------

(1 row)

SET
SET
SET
SET
CREATE EXTENSION
COMMENT
CREATE EXTENSION
COMMENT
CREATE TEXT SEARCH DICTIONARY
SET
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE MATERIALIZED VIEW
CREATE TABLE
CREATE MATERIALIZED VIEW
```

Figure 0.8: Importing the dataset on Linux

5. To test if the dataset was imported correctly, type **`psql`** and then press the **`return`** key to enter the PostgreSQL shell. Then, run **`\c sqlda`** followed by **`\dt`** to see the list of tables within the database.

```
postgres=# \c sqlda
You are now connected to database "sqlda" as user "postgres".
sqlda=# \dt
                  List of relations
 Schema |             Name              | Type  |  Owner
--------+-------------------------------+-------+----------
 public | closest_dealerships           | table | postgres
 public | countries                     | table | postgres
 public | customer_sales                | table | postgres
 public | customer_survey               | table | postgres
 public | customers                     | table | postgres
 public | dealerships                   | table | postgres
 public | emails                        | table | postgres
 public | products                      | table | postgres
 public | public_transportation_by_zip  | table | postgres
 public | sales                         | table | postgres
 public | salespeople                   | table | postgres
 public | top_cities_data               | table | postgres
(12 rows)
```

Figure 0.9: Validating the import on Linux

LOADING THE SAMPLE DATASETS – MACOS

Most exercises in this course use a sample database, sqlda, which contains fabricated data for a fictional electric vehicle company called ZoomZoom. Let's set it up by performing the following steps:

1. Enter the PostgreSQL shell by typing the following command in Terminal. Press the return key to execute it:

```
psql postgres
```

2. Now, create a new database called **sqlda** by typing the following command and pressing **return**. (Don't forget the semicolon at the end.)

```
create database sqlda;
```

3. You should see the following output. Type **\l** (that's a backslash followed by lowercase L) in Terminal and press the **return** key to check if the database was successfully created (you should see the **sqlda** database listed there).

```
postgres=# create database sqlda;
CREATE DATABASE
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding | Collate | Ctype |   Access privileges
-----------+----------+----------+---------+-------+-----------------------
 maahedev  | maahedev | UTF8     | C       | C     |
 postgres  | maahedev | UTF8     | C       | C     |
 sqlda     | maahedev | UTF8     | C       | C     |
 template0 | maahedev | UTF8     | C       | C     | =c/maahedev          +
           |          |          |         |       | maahedev=CTc/maahedev
 template1 | maahedev | UTF8     | C       | C     | =c/maahedev          +
           |          |          |         |       | maahedev=CTc/maahedev
(5 rows)

postgres=#
```

Figure 0.10: Checking if a new database is successfully created

4. Type or paste **\q** in the PostgreSQL shell and press the **return** key to exit.
5. Download the **data.dump** file from the **Datasets** folder in the GitHub repository of this course by clicking this link: https://github.com/TrainingByPackt/SQL-for-Data-Analytics/tree/master/Datasets. Navigate to the folder where you have downloaded the file using the **cd** command. Then, type the following command:

```
psql sqlda < ~/Downloads/data.dump
```

> **NOTE**
>
> The preceding command assumes that the file is saved in the **`Downloads`** directory. Make sure you change the highlighted path based on the location of the **`data.dump`** file on your system.

Then, wait for the dataset to be imported:

```
~
➜ psql  sqlda < ~/Downloads/data.dump
SET
SET
SET
SET
SET
 set_config
------------

(1 row)

SET
SET
SET
SET
CREATE EXTENSION
COMMENT
CREATE EXTENSION
COMMENT
CREATE TEXT SEARCH DICTIONARY
SET
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE MATERIALIZED VIEW
CREATE TABLE
CREATE MATERIALIZED VIEW
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

Figure 0.11: Importing the dataset

6. To test if the dataset was imported correctly, type **`psql`** and then press the **`return`** key to enter the PostgreSQL shell again. Then, run **`\c sqlda`** followed by **`\dt`** to see the list of tables within the database.

```
postgres=# \c sqlda
You are now connected to database "sqlda" as user "maahedev".
sqlda=# \dt
                  List of relations
 Schema |             Name             | Type  |  Owner
--------+------------------------------+-------+----------
 public | closest_dealerships          | table | maahedev
 public | countries                    | table | maahedev
 public | customer_sales               | table | maahedev
 public | customer_survey              | table | maahedev
 public | customers                    | table | maahedev
 public | dealerships                  | table | maahedev
 public | emails                       | table | maahedev
 public | products                     | table | maahedev
 public | public_transportation_by_zip | table | maahedev
 public | sales                        | table | maahedev
 public | salespeople                  | table | maahedev
 public | top_cities_data              | table | maahedev
(12 rows)

sqlda=#
```

Figure 0.12: List of tables within the sqlda database