

# Hands-on Lab : Getting started with PostgreSQL command line

**Estimated time needed:** 20 minutes

In this lab, you will use the PostgreSQL command line interface (CLI) to create a database and to restore the structure and contents of the tables it contains. Then you will learn how to explore and query tables. Finally, you will learn how to dump/backup tables from a database.

## Software Used in this Lab

In this lab, you will use a [PostgreSQL Database](#). PostgreSQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve the data.



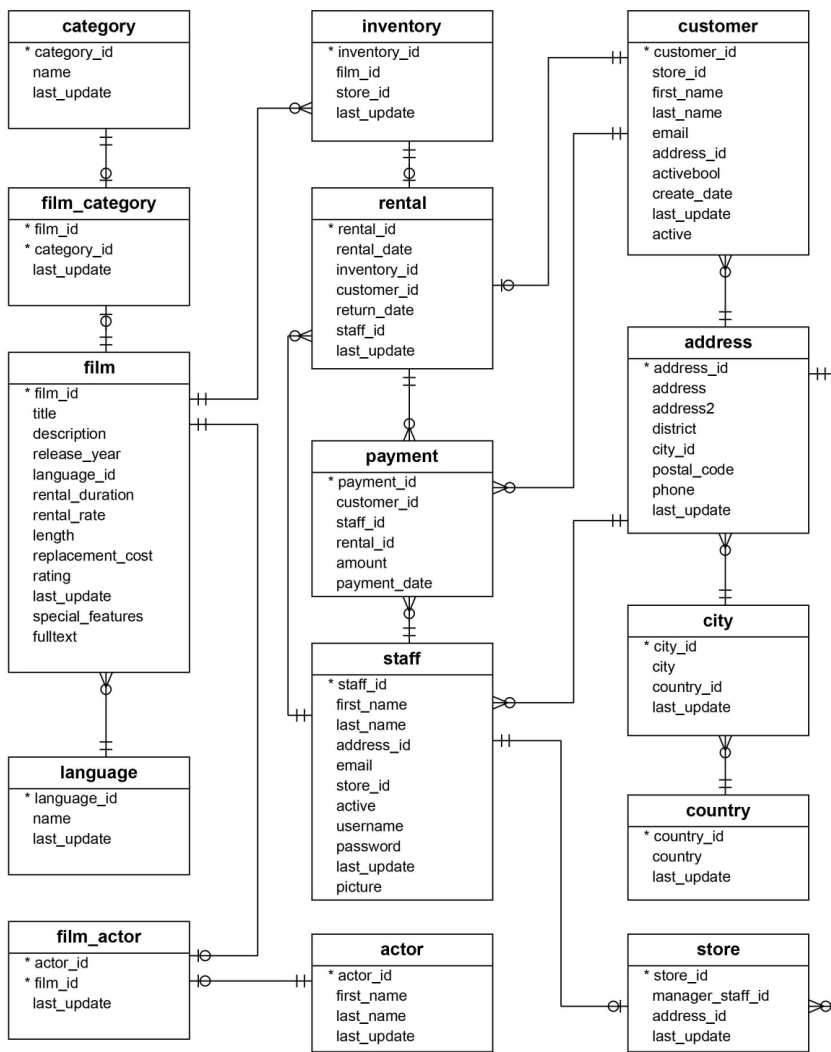
To complete this lab you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

::page{title="Database Used in this Lab"}

The Sakila database used in this lab comes from the following source: <https://dev.mysql.com/doc/sakila/en/> under [New BSD license](#) [Copyright 2021 - Oracle Corporation].

You will use a modified version of the database for the lab, so to follow the lab instructions successfully please use the database provided with the lab, rather than the database from the original source.

The following Entity Relation Diagram (ERD) diagram shows the structure of the schema of the Sakila database:



# bjectives

After completing this lab, you will be able to use the PostgreSQL command line to:

- Create a database.
- Restore the structure and data of a table.
- Explore and query tables.
- Dump/backup tables from a database.

# Lab Structure



In this exercise, you will go through several subtasks where you will use the PostgreSQL command line interface (CLI) to create a database and to restore the structure and contents of tables. Then you will learn how to explore and query tables. Finally, you will learn how to dump/backup tables from a database.


## Task A: Create a database

To get started with this lab, launch PostgreSQL using the Cloud IDE. You can do this by following these steps:

1. Click on the Skills Network extension button on the left side of the window.
2. Open the **DATABASES** drop down menu and click on **PostgreSQL**
3. Click on the **Start** button. PostgreSQL may take a few moments to start.

File Edit Selection View Go Run Terminal Help

SKILLS NETWORK TOOLB...  

MySQL PostgreSQL 

▼ DATABASES

MySQL INACTIVE

PostgreSQL INACTIVE **2**


Cassandra INACTIVE


MongoDB INACTIVE


> BIG DATA

> CLOUD

> OTHER

 **1**





# PostgreSQL

INACTIVE

v13.2 | v5.0 | v13.2

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

**Start** **3**

Summary Connection Information Details

Get started with PostgreSQL in a faster, easier way. To launch your database, hi

4. Open up a new command terminal by clicking on the **New Terminal** button.



PostgreSQL x



# PostgreSQL

ACTIVE

 v13.2 |  v5.0 |  v13.2

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

Stop


Summary

Connection Information

Details

Your database and pgAdmin server are now ready to use and available with the following on how to navigate PostgreSQL, please check out the Details section.

**Username:**

**Password:**

You can manage PostgreSQL via:

pgAdmin



Or to interact with the database in the terminal, select one of these options:



PostgreSQL CLI

New Terminal

5. Copy the command below by clicking on the little copy button on the bottom right of the codeblock and then paste it into the terminal using **Ctrl + V** (Mac: **⌘ + V**) to fetch the [sakila\\_pgsql\\_dump.sql](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsql_dump.sql) file to the Cloud IDE.

```
1. 1
1. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsql_dump.sql
```

Copied!

```
theia@theiadocker-sandipsahajo:/home/project$ wget https://cf-courses-data.s3.
--2021-03-22 04:19:25-- https://cf-courses-data.s3.us.cloud-object-storage.ap
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-cours
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-c
HTTP request sent, awaiting response... 200 OK
Length: 2764132 (2.6M) [application/x-sql]
Saving to: 'sakila_pgsql_dump.sql'

sakila_pgsql_dump.sql          100%[=====

2021-03-22 04:19:28 (972 KB/s) - 'sakila_pgsql_dump.sql' saved [2764132/276413

theia@theiadocker-sandipsahajo:/home/project$
```

6. Now open up the PostgreSQL Command Line Interface (CLI) by clicking on the **PostgreSQL CLI** button.



PostgreSQL x



# PostgreSQL

ACTIVE

 v13.2 |  v5.0 |  v13.2

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

Stop

Summary

Connection Information

Details

Your database and pgAdmin server are now ready to use and available with the following on how to navigate PostgreSQL, please check out the Details section.

Username:

Password:

You can manage PostgreSQL via:

pgAdmin



Or to interact with the database in the terminal, select one of these options:



PostgreSQL CLI

New Terminal

7. Create a new database **sakila** using the command below in the terminal and proceed to Task B:

```
1. 1
1. create database sakila;
```

Copied!

```
theia@theiadocker-sandipsahajo:/home/project$ psql --username=postgres --host=
Password:
psql (13.2 (Ubuntu 13.2-1.pgdg18.04+1))
Type "help" for help.

postgres=# create database sakila;
CREATE DATABASE
postgres=#
```

**Note:** You are using **create database** command to create a new database within the PostgreSQL CLI. To create a new database named sakila outside the command line interface, you can use the following command directly in a terminal window: `createdb --username=postgres --host=localhost --password sakila` after quitting the psql command prompt session with command `\q`.

## Task B: Restore the structure and data of a table

1. To connect to the newly created empty sakila database, use the command below in the terminal and enter your PostgreSQL service session password:

```
1. 1
1. \connect sakila;
```

Copied!

```
postgres=# \connect sakila;
Password:
You are now connected to database "sakila" as user "postgres".
```



2. Restore the sakila PostgreSQL dump file (containing the sakila database table definitions and data) to the newly created empty sakila database using the command below in the terminal:

```
1. 1
1. \include sakila_pgsql_dump.sql;
```

Copied!

```
sakila=# \include sakila_pgsql_dump.sql;
```

**Note:** You are using the **\include** command to restore the database dump file within the PostgreSQL CLI. To restore the database dump file outside of the Command Line Interface, you can use the command `pg_restore --username=postgres --host=localhost --password --dbname=sakila < sakila_pgsql_dump.tar` after quitting the CLI prompt session with command `\q`. Non-text format **.tar** dumps are restored using the **pg\_restore** command. So, before the using mentioned **pg\_restore** command, first fetch the .tar version of this dump file using the command `wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsql_dump.tar`

3. Repeat Step 1 to reconnect to the sakila database after restoring the dump file. Proceed to Task C.

## Task C: Explore and query tables

1. To list all the tables names from the sakila database, use the command below in the terminal:

```
1. 1
1. \dt
```

Copied!

```
sakila=# \dt;
```

List of relations

Schema	Name	Type	Owner
public	actor	table	postgres
public	address	table	postgres
public	category	table	postgres
public	city	table	postgres
public	country	table	postgres
public	customer	table	postgres
public	film	table	postgres
public	film_actor	table	postgres
public	film_category	table	postgres
public	inventory	table	postgres
public	language	table	postgres
public	payment	table	postgres
public	rental	table	postgres
public	staff	table	postgres
public	store	table	postgres

(15 rows)

```
sakila=#
```

2. Explore the structure of the **store** table using the command below in the terminal:

```
1. 1
1. \d store;
```

Copied!

```
sakila=# \d store;
```

Table "public.store"				
Column	Type	Collation	Nullable	
store_id	integer		not null	nextval('store_id_seq'::regclass)
manager_staff_id	smallint		not null	
address_id	smallint		not null	
last_update	timestamp without time zone		not null	now()

Indexes:

"store\_pkey" PRIMARY KEY, btree (store\_id)

"idx\_unq\_manager\_staff\_id" UNIQUE, btree (manager\_staff\_id)

Foreign-key constraints:

"store\_address\_id\_fkey" FOREIGN KEY (address\_id) REFERENCES address(address\_id)

"store\_manager\_staff\_id\_fkey" FOREIGN KEY (manager\_staff\_id) REFERENCES staff(staff\_id)

Triggers:

last\_updated BEFORE UPDATE ON store FOR EACH ROW EXECUTE FUNCTION last\_updated()

```
sakila=#
```

3. Retrieve all the records from the **store** table using the command below in the terminal:

```
1. 1
```

```
1. SELECT * FROM store;
```

Copied!

```
sakila=# SELECT * FROM store;
```

store_id	manager_staff_id	address_id	last_update
1	1	1	2006-02-15 09:57:12
2	2	2	2006-02-15 09:57:12

(2 rows)

4. Quit the PostgreSQL command prompt session using the command below in the terminal and proceed to Task D:

```
1. 1
1. \q
```

Copied!

```
sakila=# \q
theia@theiadocker-sandipsahajo:/home/project$
```

::page{title="Task D: Dump/backup tables from a database"}

1. Finally, to dump/backup the **store** table from the database, use the command below in the terminal and enter your PostgreSQL service session password:

```
1. 1
1. pg_dump --username=postgres --host=localhost --password --dbname=sakila --table=store --format=plain > sakila_store_pgsql_dump.sql
```

Copied!

**Note:** To only dump/backup the table **store** from the database in non-text format **.tar**, you can use command `pg_dump --username=postgres --host=localhost --password --dbname=sakila --table=store --format=tar > sakila_store_pgsql_dump.tar`

2. To view the dump file within the terminal, use the command below in the terminal:

```
1. 1
1. cat sakila_store_pgsql_dump.sql
```

Copied!

```
theia@theiadocker-sandipsahajo:/home/project$ pg_dump --username=postgres --h
Password:
theia@theiadocker-sandipsahajo:/home/project$ cat sakila_store_pgsql_dump.sql
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.2
-- Dumped by pg_dump version 13.2 (Ubuntu 13.2-1.pgdg18.04+1)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: store; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.store (
    store_id integer DEFAULT nextval('public.store_store_id_seq'::regclass) NO
```

```
store_id integer DEFAULT nextval('public.store_store_id_seq'::regclass) NOT NULL,  
manager_staff_id smallint NOT NULL,  
address_id smallint NOT NULL,  
last_update timestamp without time zone DEFAULT now() NOT NULL  
);
```

## Conclusion

Congratulations! You have completed this lab, and you are ready for the next topic.

## Author

- [Sandip Saha Joy](#)

## Other Contributors

- [David Pasternak](#)

## Changelog

Date	Version	Changed by	Change Description
2021-03-15	1.0	Sandip Saha Joy	Created initial version
2021-10-18	1.1	David Pasternak	Updated lab instructions

© IBM Corporation 2020. All rights reserved.