

Laboratório Prático: Introdução à Linha de Comando do PostgreSQL



Tempo estimado necessário: 20 minutos

Neste laboratório, você usará a interface de linha de comando (CLI) do PostgreSQL para criar um banco de dados e restaurar a estrutura e o conteúdo de suas tabelas. Em seguida, você aprenderá como explorar e consultar tabelas. Por fim, você aprenderá como exportar/fazer backup de tabelas de um banco de dados.

Software utilizado neste laboratório

Neste laboratório, você usará um [Banco de Dados PostgreSQL](#). O PostgreSQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) projetado para armazenar, manipular e recuperar dados de forma eficiente.



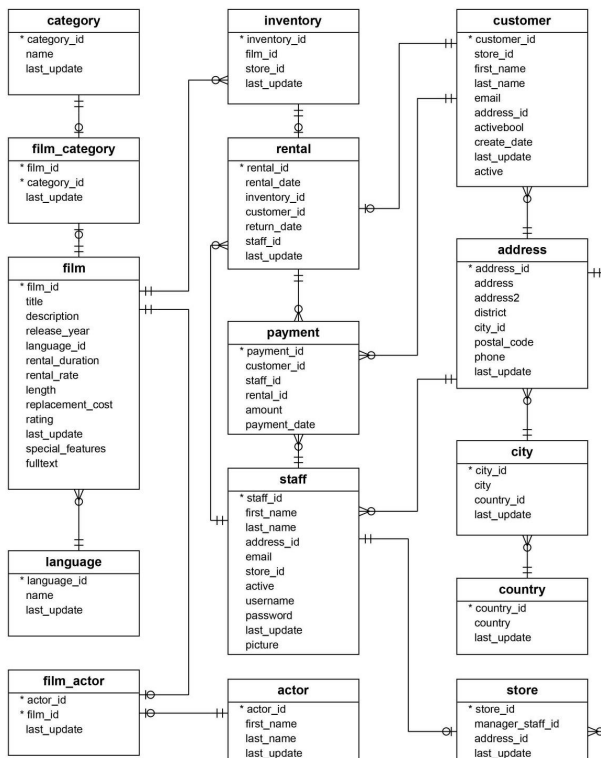
Para completar este laboratório, você utilizará o serviço de banco de dados relacional PostgreSQL disponível como parte do IBM Skills Network Labs (SN Labs) Cloud IDE. O SN Labs é um ambiente de laboratório virtual utilizado neste curso.

Banco de dados utilizado neste laboratório

O banco de dados Sakila utilizado neste laboratório vem da seguinte fonte: <https://dev.mysql.com/doc/sakila/en/> sob a [Nova licença BSD](#) [Copyright 2021 - Oracle Corporation].

Você usará uma versão modificada do banco de dados para o laboratório. Para seguir as instruções do laboratório com sucesso, use o banco de dados fornecido pelo laboratório em vez do banco de dados da fonte.

O seguinte diagrama de relação de entidade (ERD) mostra a estrutura do esquema do banco de dados Sakila:



Objetivos

Após concluir este laboratório, você será capaz de usar a linha de comando do PostgreSQL para:

- Criar um banco de dados
- Restaurar a estrutura e os dados de uma tabela
- Explorar e consultar tabelas
- Exportar/fazer backup de tabelas de um banco de dados

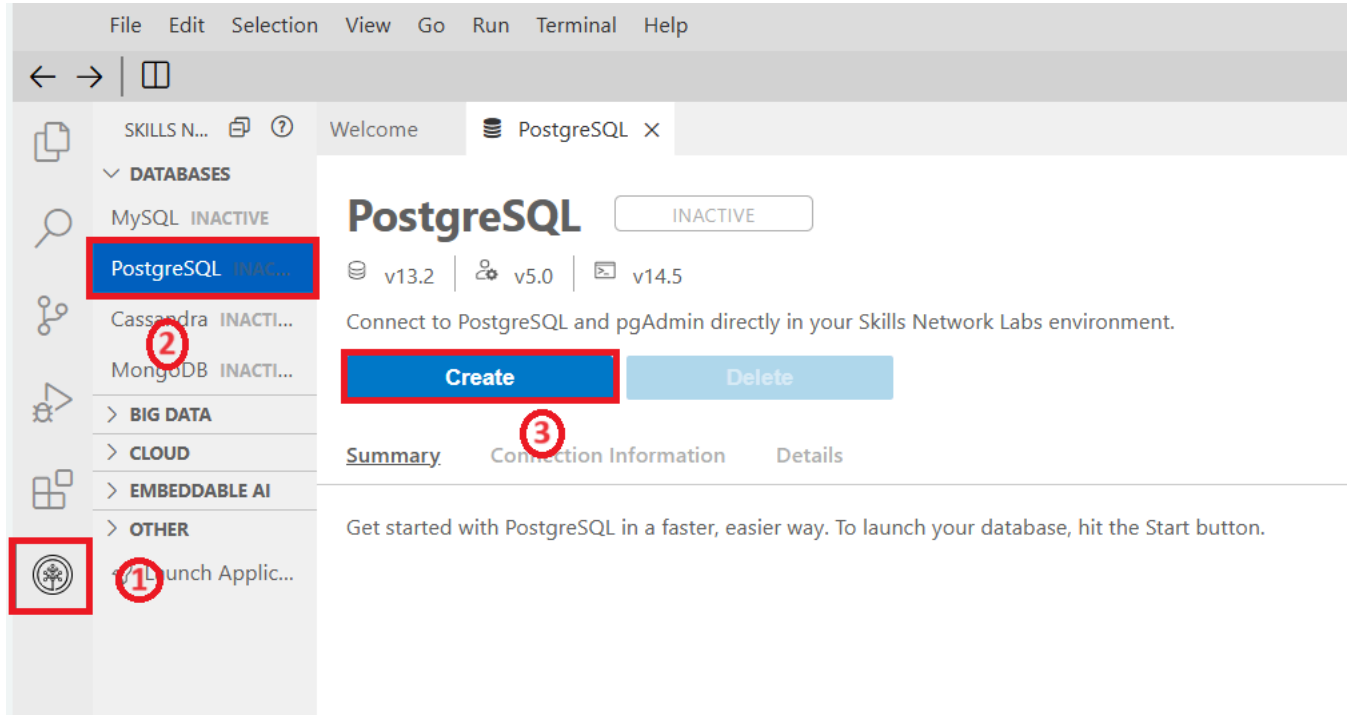
Estrutura do laboratório

Neste exercício, você passará por várias subtarefas onde usará a interface de linha de comando (CLI) do PostgreSQL para criar um banco de dados e restaurar a estrutura e o conteúdo das tabelas. Em seguida, você aprenderá como explorar e consultar tabelas. Por fim, você aprenderá como exportar/fazer backup de tabelas de um banco de dados.

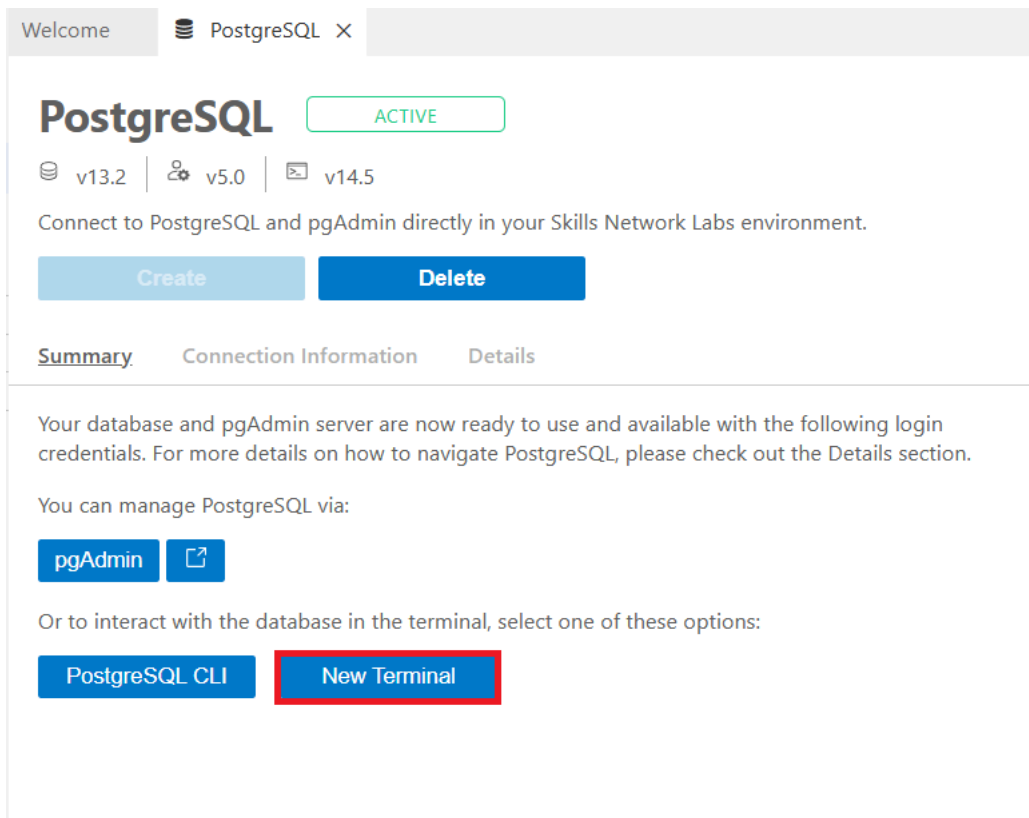
Tarefa A: Criar um banco de dados

Para começar este laboratório, inicie o PostgreSQL usando o Cloud IDE. Você pode fazer isso seguindo estas etapas:

1. Clique no botão de extensão do Skills Network no lado esquerdo da janela.
2. Abra o menu **DATABASES** e clique em **PostgreSQL**.
3. Clique em **Criar**. O PostgreSQL pode levar alguns momentos para iniciar.



4. Abra um novo terminal de comando clicando em **Novo Terminal**.



5. Copie o comando abaixo clicando no pequeno botão de copiar à direita do bloco de código e, em seguida, cole-o no terminal usando **Ctrl + V** (Mac: **⌘ + V**) para buscar o arquivo [sakila_pgsq1_dump.sql](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsq1_dump.sql) para o Cloud IDE.

```
wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsq1_dump.sql
```

```

theia@theiadocker-sowmyaag: /home/project X

theia@theiadocker-sowmyaag:/home/project$ wget https://cf-courses-data.s3.us.cloud-object-storage/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsql_dump.sql
--2024-08-28 21:29:02-- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-
network/datasets/sakila/sakila_pgsql_dump.sql
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.clou
appdomain.cloud)... 169.63.118.104, 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.
age.appdomain.cloud)|169.63.118.104|:443... connected.
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%[=====>]    2.64M  --.-KB/s

2024-08-28 21:29:03 (148 MB/s) - 'sakila_pgsql_dump.sql' saved [2764132/2764132]

theia@theiadocker-sowmyaag:/home/project$

```

6. Agora, abra a Interface de Linha de Comando do PostgreSQL (CLI) clicando em **PostgreSQL CLI**.

```

theia@theiadocker-sowmyaag: /home/project theia@theiadocker-sowmyaag: /home/project X

theia@theiadocker-sowmyaag:/home/project$ export PGPASSWORD=HRohjPW6Ns1XC9htrfXu4KEi; psql --host
p 5432 -U postgres
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 13.2)
Type "help" for help.

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

```

7. Crie um novo banco de dados chamado **sakila** usando o seguinte comando no terminal:

```
create database sakila;
```

```

theia@theiadocker-sowmyaag: /home/project theia@theiadocker-sowmyaag: /home/project X

theia@theiadocker-sowmyaag:/home/project$ export PGPASSWORD=HRohjPW6Ns1XC9htrfXu4KEi; psql --host
p 5432 -U postgres
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 13.2)
Type "help" for help.

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

```

Nota: Você está usando o comando **create database** para criar um novo banco de dados dentro da CLI do PostgreSQL. Para criar um novo banco de dados chamado sakila fora da interface de linha de comando, você pode usar o seguinte comando diretamente em uma janela de terminal: `createdb --username=postgres --host=postgres --password sakila` após sair da sessão do prompt de comando psql com o comando `\q`.

Tarefa B: Restaurar a estrutura e os dados de uma tabela

1. Para se conectar ao novo banco de dados vaziao sakila, use o seguinte comando no terminal e insira sua senha de sessão do serviço PostgreSQL:

```
\connect sakila;
```

```
postgres=# \connect sakila;
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 13.2)
You are now connected to database "sakila" as user "postgres".
sakila=#
```

2. Restaure o arquivo de despejo do PostgreSQL sakila (contendo as definições e dados das tabelas do banco de dados sakila) para o novo banco de dados vazio sakila usando o seguinte comando no terminal:

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

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

Nota: Você está usando o comando `\include` para restaurar o arquivo de despejo do banco de dados dentro da CLI do PostgreSQL. Para restaurar o arquivo de despejo do banco de dados fora da Interface de Linha de Comando, você pode usar o comando `pg_restore --username=postgres --host=postgres --password --dbname=sakila < sakila_pgsql_dump.tar` após sair da sessão do prompt CLI com o comando `\q`. Dumps em formato não texto `.tar` são restaurados usando o comando `pg_restore`. Portanto, antes de usar o comando `pg_restore`, primeiro busque a versão `.tar` deste arquivo de despejo usando o comando `wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0110EN-SkillsNetwork/datasets/sakila/sakila_pgsql_dump.tar`

3. Repita a Etapa 1 para reconectar ao banco de dados sakila após restaurar o arquivo de despejo.

Tarefa C: Explorar e consultar tabelas

1. Para listar todos os nomes das tabelas do banco de dados sakila, use o seguinte comando no terminal:

```
\dt
```

```
sakila=# \connect sakila;
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 13.2)
You are now connected to database "sakila" as user "postgres".
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
```

2. Explore a estrutura da tabela **store** usando o seguinte comando no terminal:

```
\d store;
```

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

Column	Type	Table "public.store"	Collation	Nullable	Default
store_id	integer			not null	nextval('store_store_id_
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) ON UPDATE CASCADE

"store_manager_staff_id_fkey" FOREIGN KEY (manager_staff_id) REFERENCES staff(staff_id) ON UPDATE RESTRICT

Triggers:

last_updated BEFORE UPDATE ON store FOR EACH ROW EXECUTE FUNCTION last_updated()

3. Recupere todos os registros da tabela **store** usando o seguinte comando no terminal:

```
SELECT * FROM store;
```

```
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. Saia da sessão do prompt de comando do PostgreSQL usando o seguinte comando no terminal.

```
\q
```

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

Tarefa D: Exportar/fazer backup de tabelas de um banco de dados

1. Finalmente, para exportar/fazer backup da tabela **store** do banco de dados, use o seguinte comando no terminal e insira sua senha de sessão do serviço PostgreSQL:

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

Nota: Para exportar/fazer backup apenas da tabela **store** do banco de dados em formato não texto **.tar**, você pode usar o comando `pg_dump --username=postgres --host=postgres --password --dbname=sakila --table=store --format=tar > sakila_store_pgsql_dump.tar`

2. Para visualizar o arquivo de despejo dentro do terminal, use o seguinte comando:

```
cat sakila_store_pgsql_dump.sql
```

```
theia@theiadocker-sandipsahajo:/home/project$ pg_dump --username=pos
Password:
theia@theiadocker-sandipsahajo:/home/project$ cat sakila_store_pgsql
--
-- 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),
    manager_staff_id smallint NOT NULL,
    address_id smallint NOT NULL,
    last_update timestamp without time zone DEFAULT now() NOT NULL
);
```

Conclusão

Parabéns! Você completou este laboratório e agora aprendeu como criar um banco de dados, restaurar a estrutura e os dados de uma tabela, explorar e consultar tabelas, e exportar/fazer backup de tabelas de um banco de dados.

Autor: [Sandip Saha Joy](#)

Outros Contribuintes: [David Pasternak](#)



Skills Network

© IBM Corporation. Todos os direitos reservados.