

Hands-on Lab: Database Design using ERDs



Estimated time needed: 45 minutes

In this lab, you will learn how to design a database by creating an entity relationship diagram (ERD) in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool. First, you will create an ERD of a database. Next, you will generate and execute an SQL script to create the database schema from its ERD. Finally, you will load the created database schema with data.

Software used in this lab

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



PostgreSQL

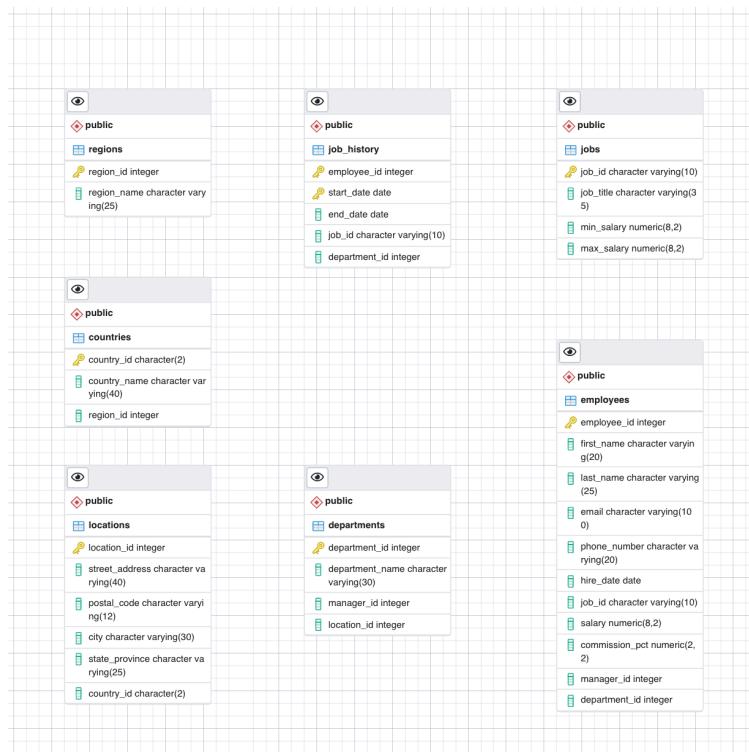
To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database used in this lab

The HR database used in this lab comes from the following source: [HR Sample Database](#) [Copyright 2021 - Oracle Corporation].

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

The following ERD shows the tables of the HR database:



Objectives

After completing this lab, you will be able to use pgAdmin with PostgreSQL to:

- Create an ERD of a database.
- Generate and execute an SQL script from an ERD to create a schema.
- Load the database schema with data.

This lab is divided into two exercises, *Example Exercise* and *Practice Exercise*.

Example Exercise

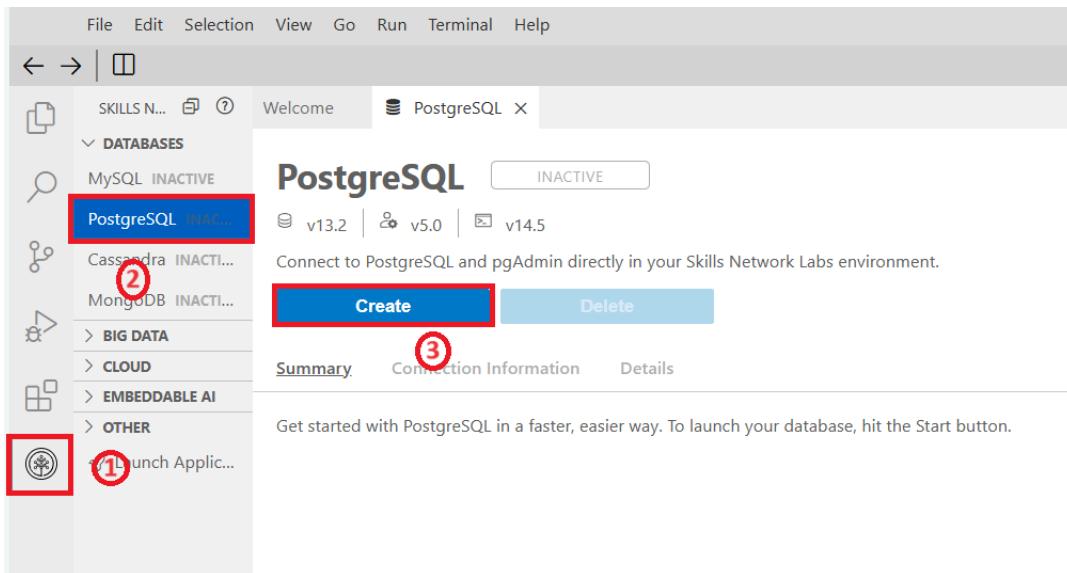
In this example exercise, you will first create a partial ERD of the HR database. Next, you will generate and execute an SQL script to create the partial schema of the HR database from its ERD. Finally, you will load the created database schema with data by using the Restore feature.

Task A: Create an Entity Relationship Diagram (ERD) of a database

In this task of the Example Exercise, you will create a partial ERD of the HR database.

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

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



4. Note down your PostgreSQL service session password because you may need to use it later in the lab.

5. Click the pgAdmin button in the same window where you started PostgresSQL.

6. You will see the pgAdmin GUI tool.

← → C ⌂ sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser     Dashboard Properties SQL Statistics Depen

>  Servers

Welcome

 pg**Admin**
Management Tools for Post
Feature rich | Maximises PostgreSQL
pgAdmin is an Open Source administration and management tool designed to answer the needs of developers, DBAs

Quick Links

 Add New

Getting Started

 PostgreSQL Documentation

Browser **1**

Servers (1) **2**

postgres **3**

Databases (1)

- postgres
 - Create
 - Cast...
 - Catalog...
 - Refresh...
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

Create - Database

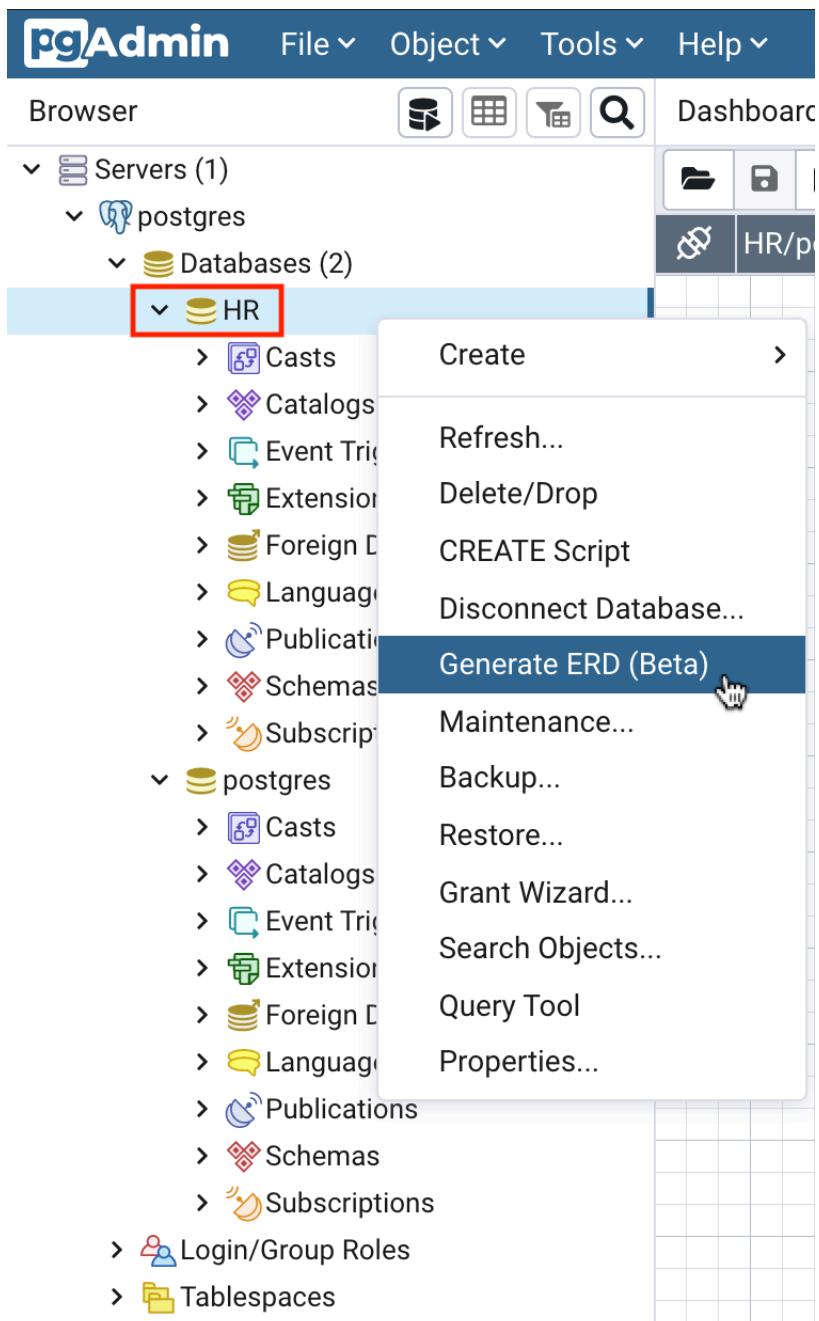
General Definition Security Parameters Advanced SQL

Database **HR**

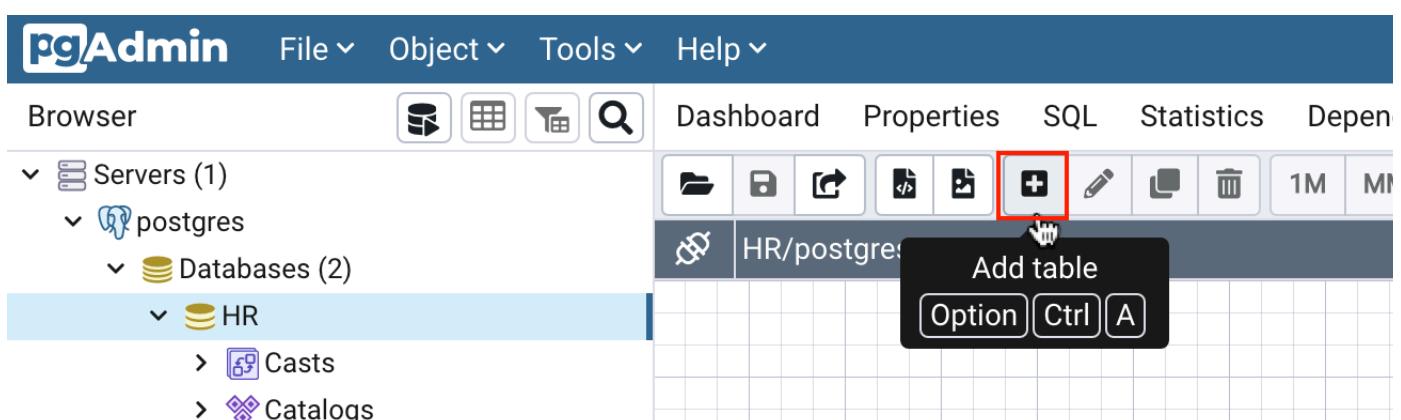
Owner postgres

Comment

i **?** **Cancel** **Reset** **S**



9. Click **Add table**. On the **General** tab, in the **Name** box, type **employees** as the name of the table. Don't click **OK**, proceed to the next step.



New table

General

Columns

Name

employees

Schema

public

Comment

-
10. Switch to the **Columns** tab and click **Add new row** to add the necessary column placeholders. Now enter the **employees** table definition information as shown in the image below to create its entity diagram. Then click **OK**.

New table

General

Columns

Columns

Name	Data type	Length/Precision	Scale

New table

General

Columns

Columns

	Name	Data type	Length/Pre
	employee_id	integer	
	first_name	character varying	
	last_name	character varying	
	email	character varying	
	phone_number	character varying	
	hire_date	date	
	job_id	character varying	
	salary	numeric	
	commission_pct	numeric	
	manager_id	integer	
	department_id	integer	

11. Similarly, create entity diagrams for the other three tables following steps 9 and 10:

▼ [Click here] Create an entity diagram for the jobs table

Click **Add table** icon. On the **General** tab, in the **Name** box, type **jobs** as the name of the table. Don't click **OK**. Switch to tab **Columns** and click **Add new row** to add the necessary column placeholders. Now enter the **jobs** table definition information as shown in the image below to create its entity diagram. Then click **OK**.

Table: jobs (public)

General Columns

Name jobs

Schema public

Comment

Cancel



Table: jobs (public)

General Columns

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary
		job_id	character varying ▾	10			
		job_title	character varying ▾	35			
		min_salary	numeric ▾	8	2		
		max_salary	numeric ▾	8	2		

Cancel



▼ [Click here] Create an entity diagram for the departments table

Click **Add table** icon. On the **General** tab, in the **Name** box, type **departments** as the name of the table. Don't click **OK**. Switch to tab **Columns** and click **Add new row** to add the necessary column placeholders. Now enter the **departments** table definition information as shown in the image below to create its entity diagram. Then click **OK**.

Table: departments (public)

General Columns

Name departments

Schema public

Comment

Cancel



Table: departments (public)

General Columns

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary
	department_id	integer			Yes	Yes
	department_name	character varying	30		Yes	No
	manager_id	integer			No	No
	location_id	integer			No	No

Cancel



▼ [Click here] Create an entity diagram for the locations table

Click **Add table** icon. On the **General** tab, in the **Name** box, type **locations** as the name of the table. Don't click **OK**. Switch to tab **Columns** and click **Add new row** to add the necessary column placeholders. Now enter the **locations** table definition information as shown in the image below to create its entity diagram. Then click **OK**.

Table: locations (public)

General Columns

Name locations

Schema public

Comment

Cancel



Table: locations (public)

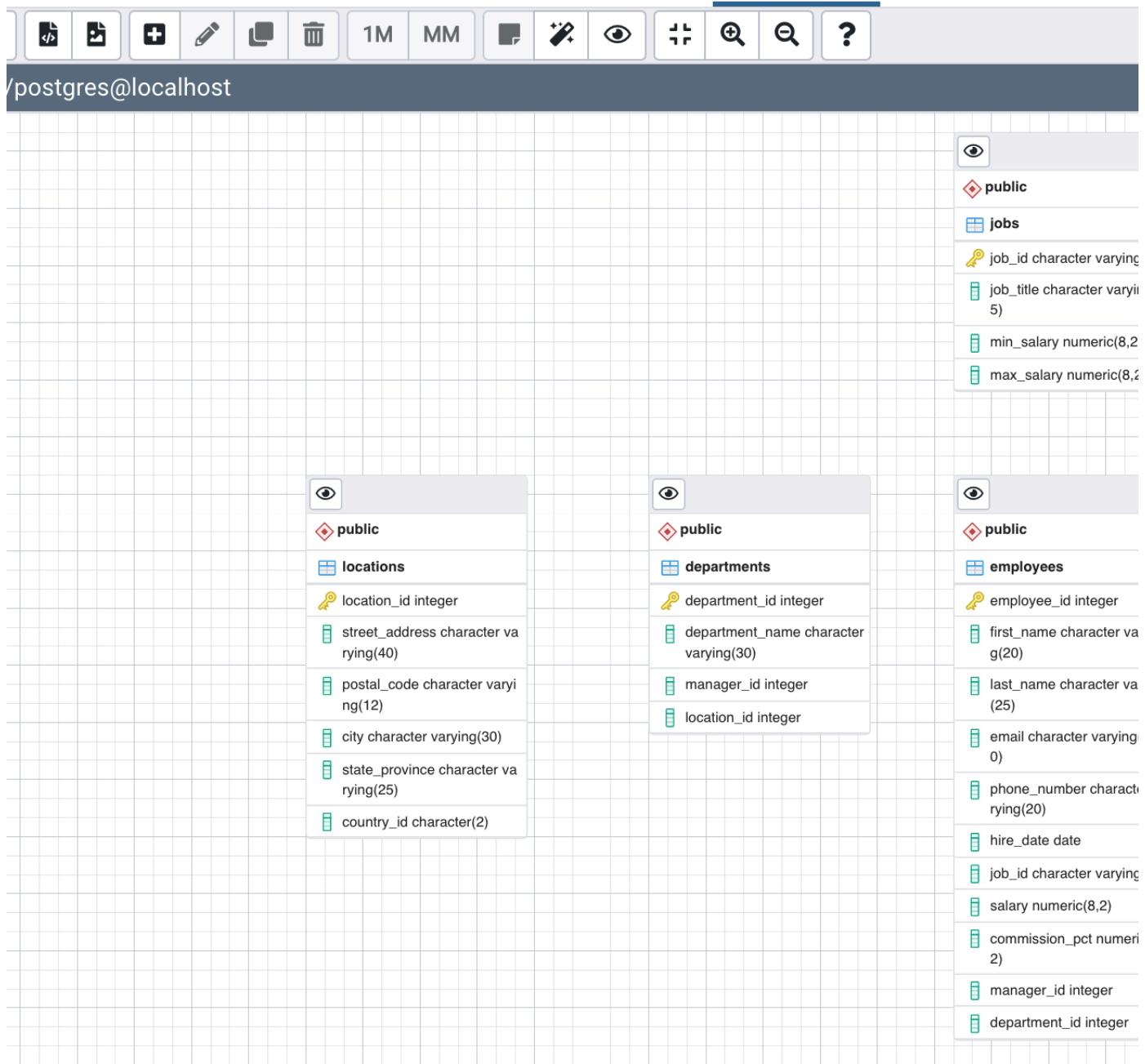
General Columns

Columns

	Name	Data type	Length/Precision	Scale	Not
	location_id	integer			
	street_address	character varying	40		
	postal_code	character varying	12		
	city	character varying	30		
	state_province	character varying	25		
	country_id	character	2		

12. After creating all four entity diagrams, the entities of the ERD are complete.

Properties SQL Statistics Dependencies Dependents Untitled*



13. Next, you will create relationships between the entities by adding foreign keys to the tables. Select the entity diagram **employees** and click **One-to-Many link**. Now enter the definition information for a foreign key on the **employees** table as shown in the image below to create the relationship. Then click **OK**.

HR/postgres@localhost

One-to-Many link
Option Ctrl O

1M MM

The screenshot shows a PostgreSQL database interface with a toolbar at the top containing various icons. A red box highlights the '1M' button in the toolbar. The main area displays two tables: 'employees' and 'departments'. The 'employees' table is highlighted with a red border. The 'departments' table is shown below it. To the right, there is a vertical list of table names and their descriptions. The interface has a light blue background with white grid lines.

employees

- employee_id integer
- first_name character varying(20)
- last_name character varying(25)
- email character varying(100)
- phone_number character varying(20)
- hire_date date
- job_id character varying(10)
- salary numeric(8,2)
- commission_pct numeric(2,2)
- manager_id integer
- department_id integer

departments

- department_id integer
- department_name character varying(30)
- manager_id integer
- location_id integer

public

public

jobs

job_id

job_type

min_salary

max_salary

locations

location_id

street_address

city

state_province

country

One to many relation



General

Local Table	(public) employees	▼
Local Column	department_id	✗ ▾
Referenced Table	(public) departments	✗ ▾
Referenced Column	department_id	✗ ▾

✗ Cancel

OK

12. Similarly, create the other relationships between the tables following the instructions in step 13:

▼ [Click here] Create a relationship between employees and jobs

Select the entity diagram **employees** and click **One-to-Many link**. Now enter the definition information for a foreign key on the **employees** table as shown in the image below to create the relationship. Then click **OK**.

One to many relation



General

Local Table	(public) employees	▼
Local Column	job_id	✗ ▾
Referenced Table	(public) jobs	✗ ▾
Referenced Column	job_id	✗ ▾

✗ Cancel

OK

▼ [Click here] Create a relationship between departments and locations

Select the entity diagram **departments** and click **One-to-Many link**. Now enter the definition information for a foreign key on the **departments** table as shown in the image below to create the relationship. Then click **OK**.

One to many relation



General

Local Table	(public) departments	▼
Local Column	location_id	✖ ▾
Referenced Table	(public) locations	✖ ▾
Referenced Column	location_id	✖ ▾

✖ Cancel

OK

▼ [Click here] Create a relationship between departments and employees

Select the entity diagram **departments** and click **One-to-Many link**. Now enter the definition information for a foreign key on the **departments** table as shown in the image below to create the relationship. Then click **OK**.

One to many relation



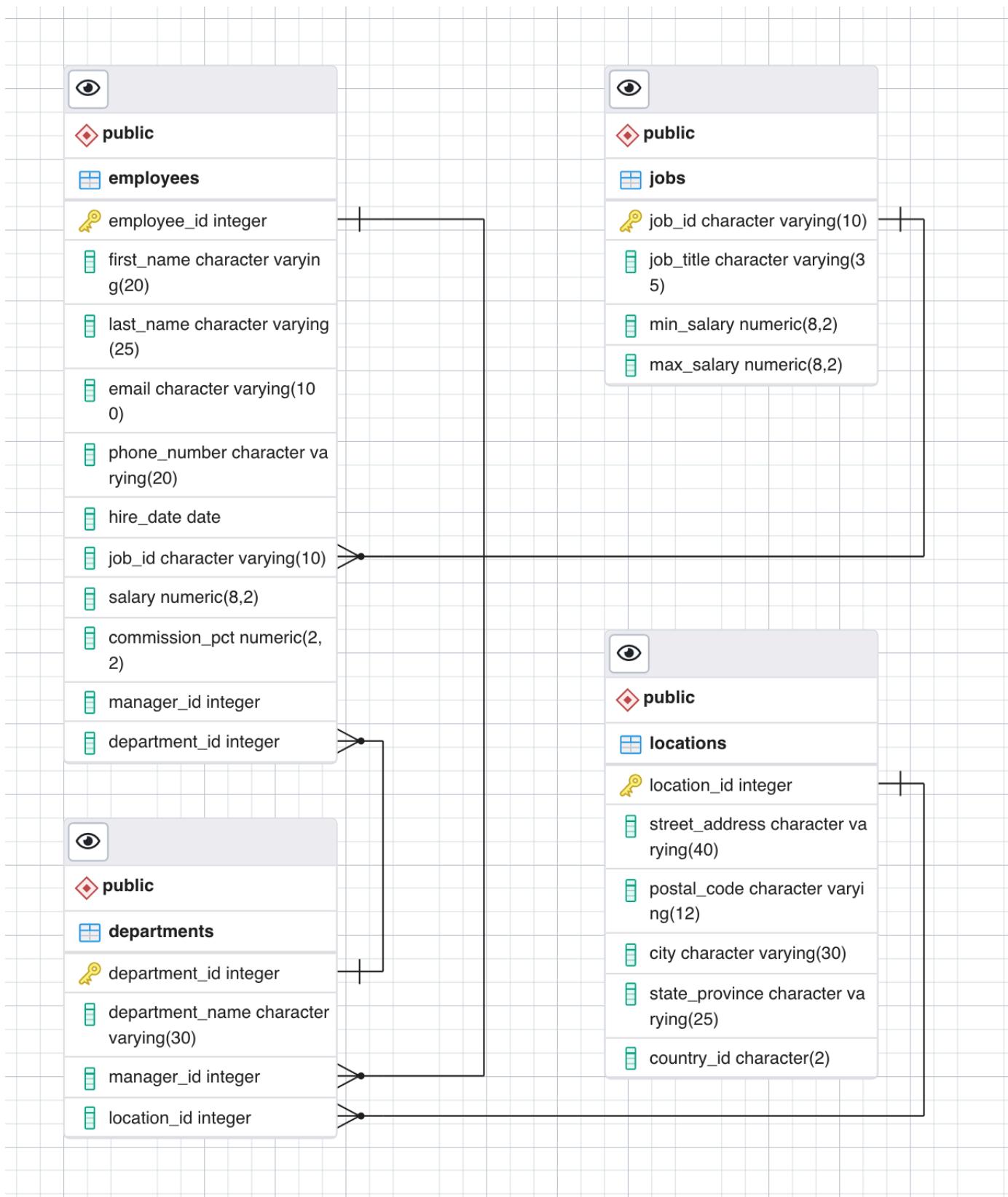
General

Local Table	(public) departments	▼
Local Column	manager_id	✖ ▾
Referenced Table	(public) employees	✖ ▾
Referenced Column	employee_id	✖ ▾

✖ Cancel

OK

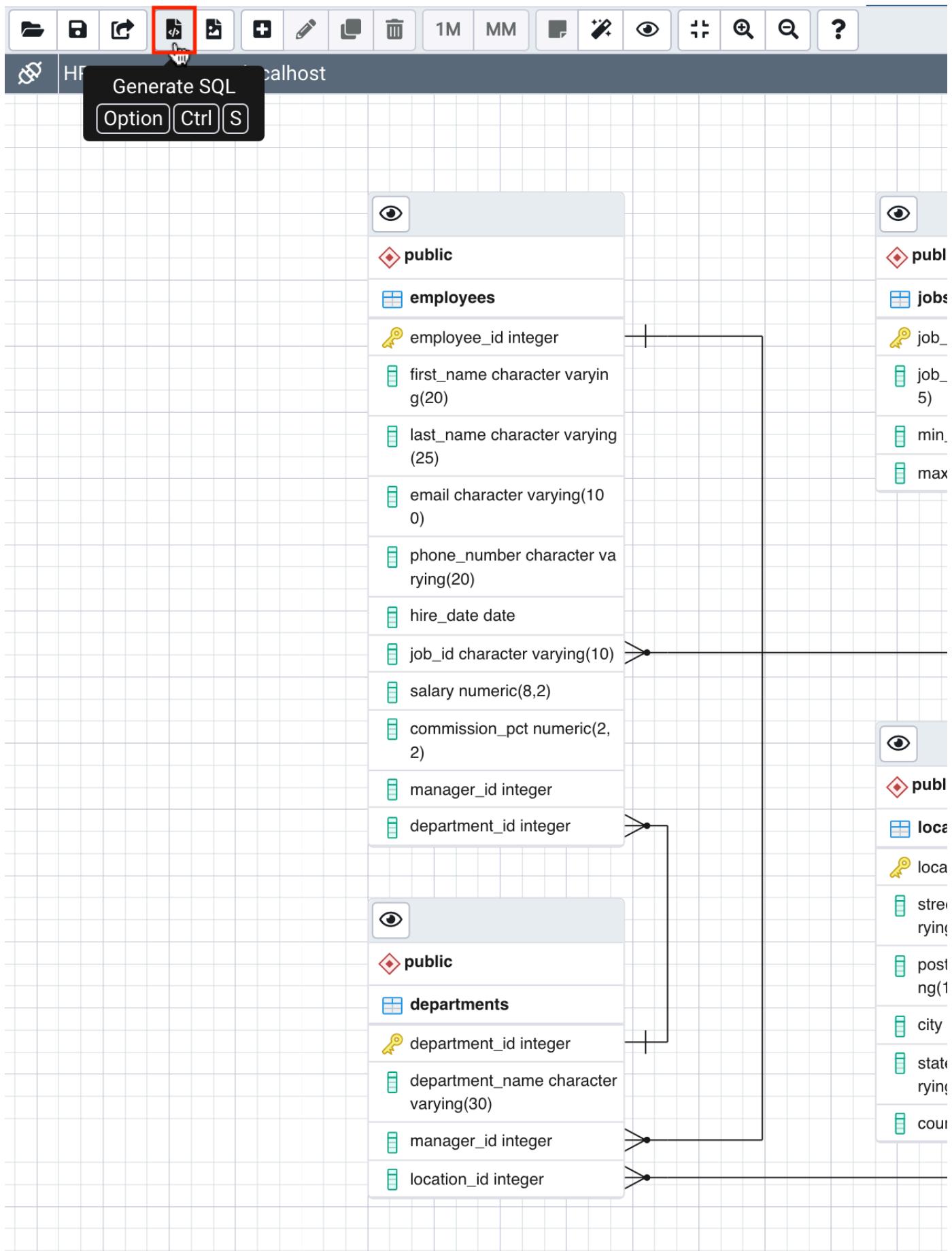
13. After creating all four relationships, you have completed the ERD for this exercise. Proceed to Task B.



Task B: Generate and execute SQL script from ERD to create the schema

In this task of the Example Exercise, you will generate and execute a SQL script from the ERD you created in Task A of the Example Exercise.

1. In the **Generate ERD (Beta)** window, click **Generate SQL**.



2. A new Query Editor window will open containing a SQL script generated from the ERD. Click Execute/Refresh to run the script. Proceed to Task C.

The screenshot shows the pgAdmin 4 interface with the Query Editor tab selected. The query window contains the following SQL code:

```
1 -- This script was generated by a beta version of the ERD tool in pgAdmin
2 -- Please log an issue at https://redmine.postgresql.org/projects/pgadmin
3 BEGIN;
4
5
6 CREATE TABLE public.departments
7 (
8     department_id integer NOT NULL,
9     department_name character varying(30) NOT NULL,
10    manager_id integer,
11    location_id integer,
12    PRIMARY KEY (department_id)
13 );
14
15 CREATE TABLE public.employees
16 (
17     employee_id integer NOT NULL,
18     first_name character varying(20),
19     last_name character varying(25) NOT NULL,
20     email character varying(100) NOT NULL,
21     phone_number character varying(20),
22     hire_date date NOT NULL,
23     job_id character varying(10) NOT NULL,
24     salary numeric(8, 2) NOT NULL,
25     commission_pct numeric(2, 2).
```

Data Output Explain Messages Notifications

COMMIT

Query returned successfully in 99 msec.

Task C: Load the database schema with data

In this task of the Example Exercise, you will load the database schema you created in Task B of the Example Exercise with data using the pgAdmin Restore feature.

1. Download the **HR_pgsql_dump_data_for_example-exercise.tar** PostgreSQL dump file (containing the partial HR database data) using the link below to your local computer.
 - [HR_pgsql_dump_data_for_example-exercise.tar](#)
2. Follow the instructions below to import/restore the data:
 - In the tree-view, expand **HR**. Right-click **HR** and click **Restore**.

pgAdmin File Object Tools Help

Browser Dashboard Properties

Servers (1) postres Databases (2) HR

HR Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Collations Domains FTS Configuration FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Procedures Sequences Tables Trigger Functions Types Views Subscriptions

Create Refresh... Delete/Drop CREATE Script Disconnect Database... Generate ERD (Beta) Maintenance... Backup... Restore... Grant Wizard... Search Objects... Query Tool Properties...

On the General tab, click Select file by the Filename box.

Restore (Database: HR)

General

Restore options

Format

Custom or tar

Filename

Number of jobs

Role name

Select an item...



Cancel



- Initially make sure the folder details empty and select the var option from the list as shown in the screenshot below. Select var folder.

Select file

Name	Date Modified	Size
run	Thu Sep 5 20:00:40 2024	
sbin	Mon Jul 29 05:02:19 2024	
srv	Mon Jul 22 14:34:18 2024	
sys	Thu Sep 5 20:08:48 2024	
tmp	Thu Sep 5 20:09:17 2024	
usr	Mon Jul 29 05:02:18 2024	
var	Mon Jul 29 05:02:20 2024	(1)
venv	Mon Jul 29 04:58:51 2024	

21 items

File Format All Files ▾

X Cancel ✓ Select

- Select lib folder.

Select file

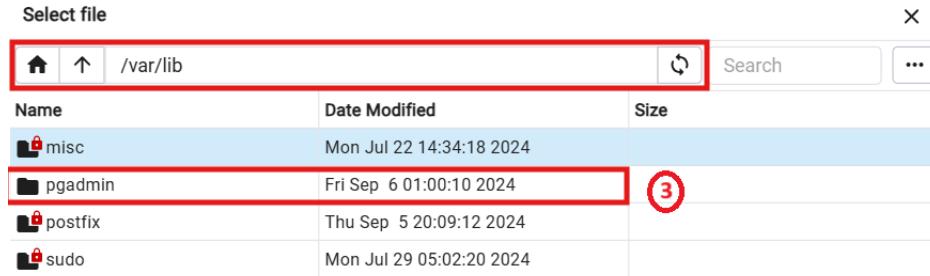
Name	Date Modified	Size
cache	Mon Jul 22 14:34:18 2024	
db	Mon Jul 29 05:02:20 2024	
empty	Mon Jul 22 14:34:18 2024	
lib	Mon Jul 29 05:02:26 2024	(2)
local	Mon Jul 22 14:34:18 2024	
lock	Mon Jul 22 14:34:18 2024	
log	Mon Jul 22 14:34:18 2024	
mail	Mon Jul 22 14:34:18 2024	

12 items

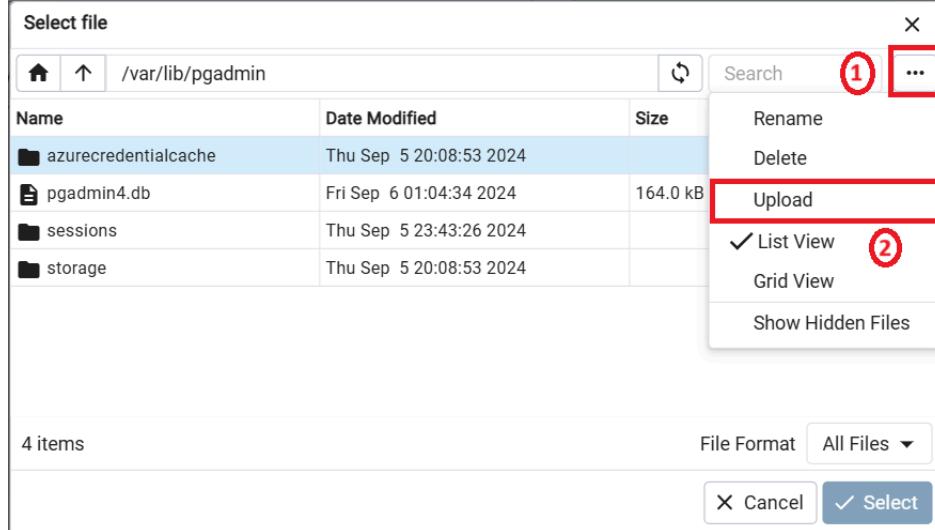
File Format All Files ▾

X Cancel ✓ Select

- Select pgadmin folder. Here you could notice the folders are locked except the pgadmin folder.



- Click Upload File. Now select upload as mentioned here.



- Double-click on the drop files area and load the **HR_pgsql_dump_data_for_example-exercise.tar** you downloaded earlier on your local computer.

Note: Ensure that you upload the files to this path: /var/lib/pgadmin/

Select file



/var/lib/pgadmin/



Double click on this space

Drop files here to upload. The file size limit (per file) is 50 mb.

Show hidden files and folders?

Format

- When the upload is complete, close the drop files area by clicking X.

Select file



/var/lib/pgadmin/



21 KB



HR_pgsql_dump...
exercise.tar

100%

Drop files here to upload. The file size limit (per file) is 50 mb.

Show hidden files and folders?

Format

- Ensure Format is set to All Files, select the uploaded HR_pgsql_dump_data_for_example-exercise.tar file from the list, and then click Select.

Select file

/var/lib/pgadmin/HR_pgsql_dump_data_for_example-ex...

Name	Size	Modified
HR_pgsql_dump_data_for_example-exercise.tar	20.5 kB	Thu Apr 1 13:46:47 2024
pgadmin4.db	156.0 kB	Thu Apr 1 13:45:18 2024
sessions	4.0 kB	Thu Apr 1 09:25:00 2024
storage	4.0 kB	Thu Apr 1 09:24:00 2024

Show hidden files and folders?

Format

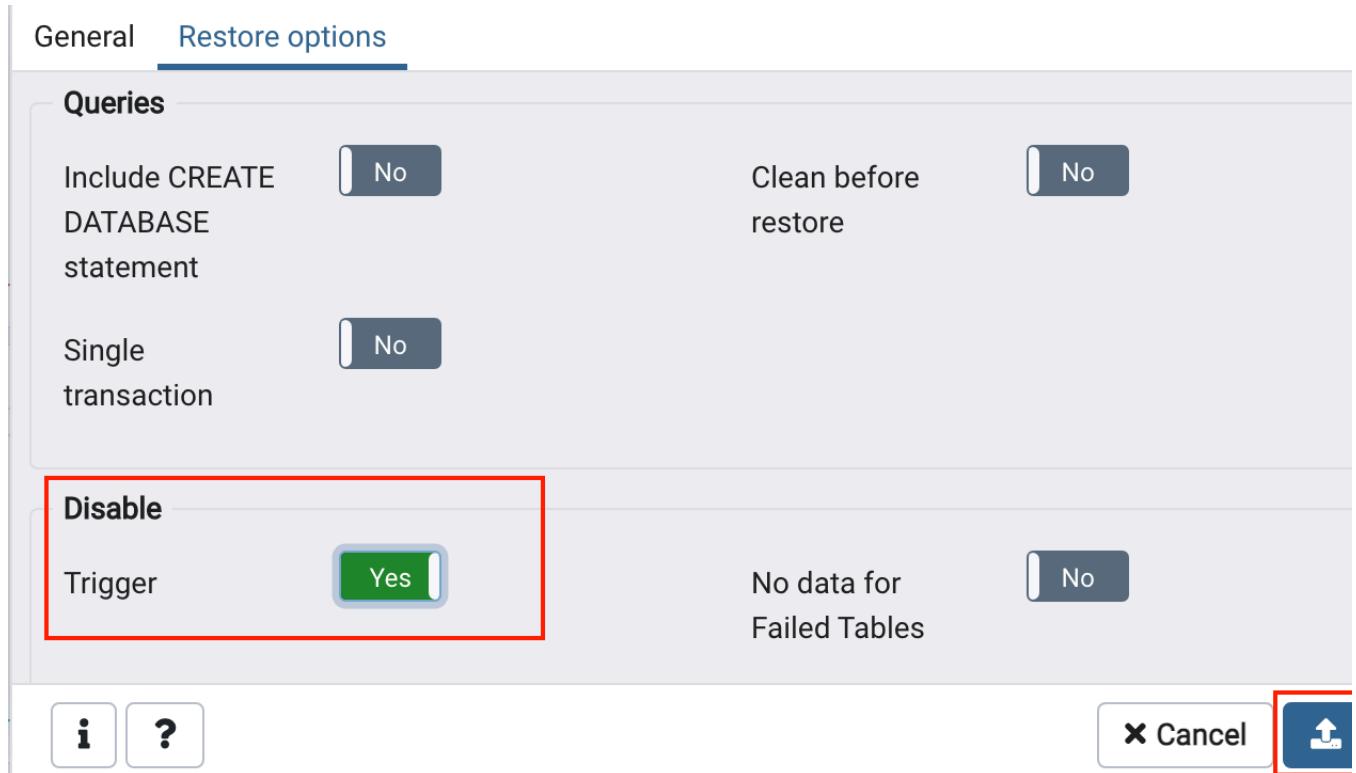
- Now switch to the **Restore options** tab.

Restore (Database: HR)

General **Restore options**

Format	Custom or tar
Filename	/var/lib/pgadmin/HR_pgsql_dump_data_for_example-exercise.tar
Number of jobs	
Role name	Select an item...

- Under **Disable**, set the **Trigger** option to **Yes**. Then click **Restore**.



Practice Exercise

In this practice exercise, first you will finish creating a partially complete ERD for the HR database. Next, you will generate and execute an SQL script to build the complete schema of the HR database from its ERD. Finally, you will load the complete database schema with data by using the Restore feature.

- Download the [HR_pgsql_ERD_for_practice-exercise.pgerd](#) ERD file (containing a partial HR database ERD based on the one that you created in Task A of the Example Exercise) below to your local computer.
- In pgAdmin, create a new database named **HR_Complete**.
- Open the ERD Tool and use **Load from file** to load the [HR_pgsql_ERD_for_practice-exercise.pgerd](#) file.

The screenshot shows the pgAdmin interface with the 'HR_Complete' database selected in the 'Browser' pane. A context menu is open over the database node, with the 'Load from file' option highlighted and a keyboard shortcut 'Ctrl + O' displayed. The menu also includes other options like 'New', 'Edit', 'Properties', 'SQL', 'Statistics', and 'Dependencies'.

Tip: Follow Example Exercise Task C for how to load any file in pgAdmin.

4. You will see the previous four entity diagrams along with relationships that you created in the Example Exercise. You will also see three new entity diagrams for the **job_history**, **regions**, and **countries** tables and one new relationship within the entity diagram of the **employees** table between *manager_id* as local column and *employee_id* as referenced column.

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser

- Servers (1)
 - postgres
 - Databases (2)
 - HR_Complete

Dashboard Properties SQL Statistics Depen

HR_Complete/postgres@postgres

public

employees

employee_id integer

first_name character varying(20)

last_name character(25)

email character(0)

phone_number character varying(20)

hire_date date

job_id character

salary numeric

commission_percent(2)

manager_id integer

department_id integer

regions

region_id integer

region_name character varying(25)

departments

department_id integer

department_name character varying(30)

manager_id integer

location_id integer

countries

country_id character(2)

country_name character varying(40)

region_id integer

locations

location_id integer

street_address character varying(40)

postal_code character varying(12)

city character

state_province character varying(25)

country_id character

5. Create the remaining relationships between the tables:

- [Click here] Create a relationship between countries and regions
- [Click here] Create a relationship between job_history and departments
- [Click here] Create a relationship between job_history and employees
- [Click here] Create a relationship between job_history and jobs
- [Click here] Create a relationship between locations and countries

Tip: Follow Example Exercise Task A for how to create relationships between the entities by adding foreign keys to the tables.

6. After creating the remaining relationships, the complete ERD of the HR database will look like the following image:



7. Generate and execute an SQL script from the ERD to create the schema of the **HR_Complete** database.

Tip: Follow Example Exercise Task B.

8. Download the **HR_pgsql_dump_data.tar** PostgreSQL dump file (containing the complete HR database data) below to your local computer. Use the dump file to restore/import the data to the **HR Complete** database.

- [HR_pgsql_dump_data.tar](#)

Tip: Follow Example Exercise Task C.

Conclusion

Congratulations! You have completed this lab, and you have learned how to create an ERD of a database, generate and execute an SQL script from an ERD to create a schema, and load the database schema with data.

Author(s)

- [Sandip Saha Joy](#)

© IBM Corporation. All rights reserved.