

# 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.



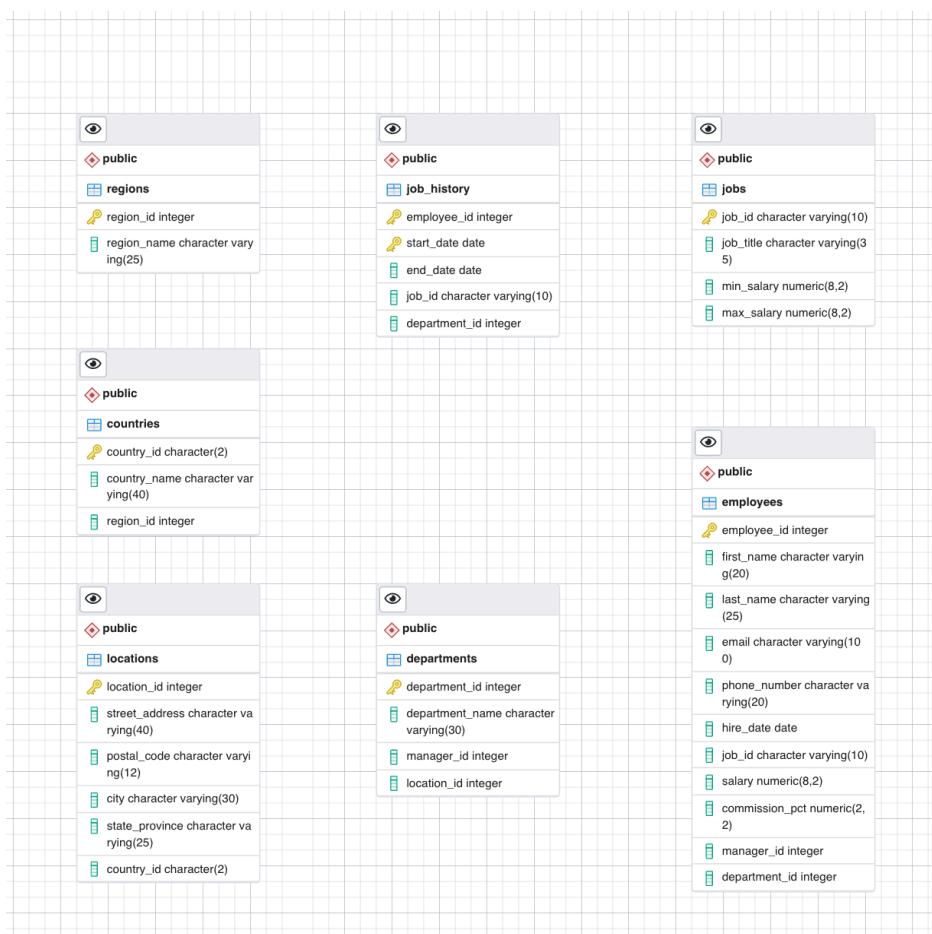
To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

## 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

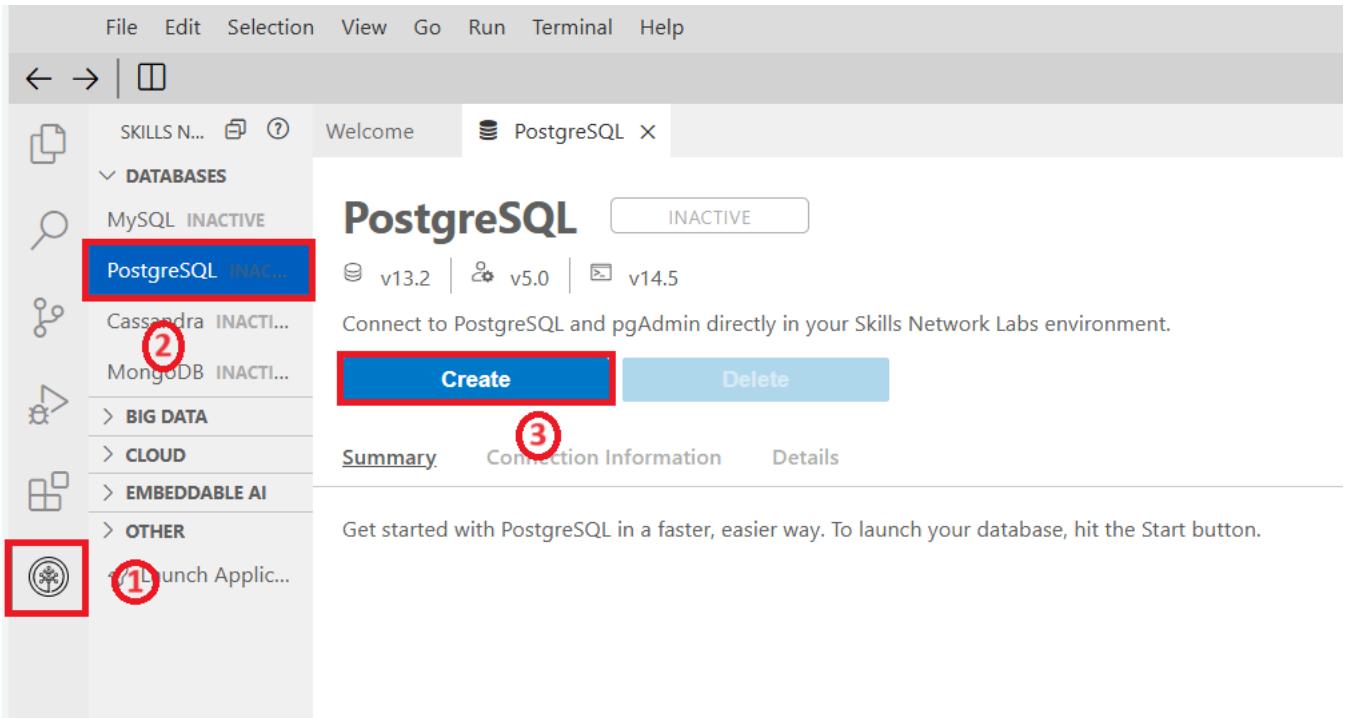
In this example exercise, you will first create a partial ERD of the HR database. Next, you will generate and execute an SQL script to create the partial schema of the HR database from its ERD. Finally, you will load the created database schema with data by using the Restore feature.

### 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 PostgreSQL.
6. You will see the pgAdmin GUI tool.

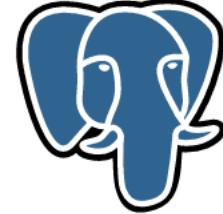
← → ⌂ ⌂ sandipsahajo-5050.theiadocker-27.proxy.cognitive

**pgAdmin** File Object Tools Help

Browser     Dashboard Properties SQL

>  Servers

**Welcome**



**pgAd**

**Manageme**

**Feature rich | Maximi**

pgAdmin is an Open Source ad  
is designed to answer the need:

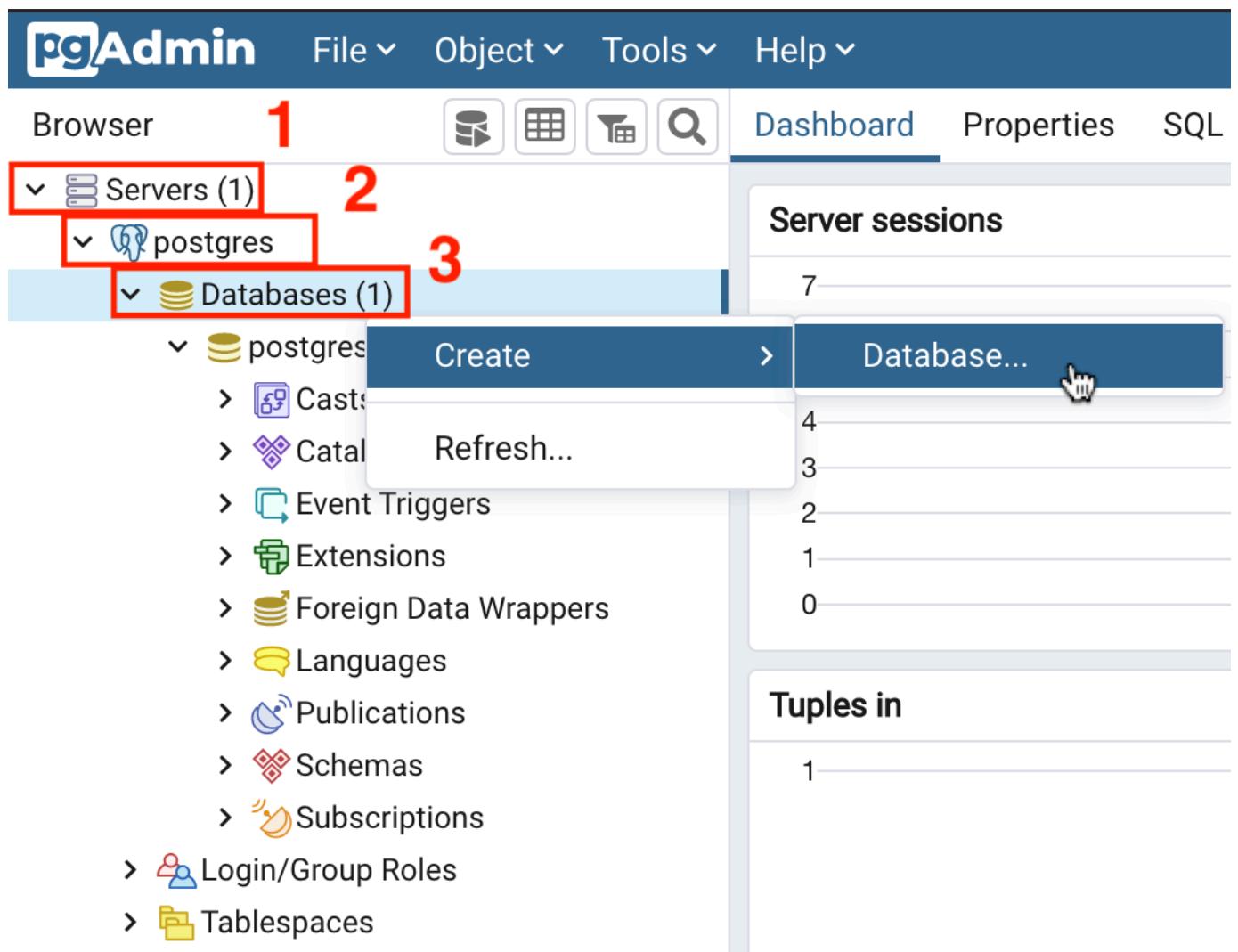
**Quick Links**

**Getting Started**



**PostgreSQL Docum**

7. In the tree-view, expand **Servers** > **postgres** > **Databases**. Enter your PostgreSQL service session password if prompted during the process. Right-click on **Databases** and go to **Create** > **Database**. Type **HR** as the name of the database and click **Save**.



 Create - Database[General](#) [Definition](#) [Security](#) [Parameters](#) [Advanced](#) [SQL](#)

Database

HR

Owner

 postgres

Comment

 Cancel

8. In the tree-view, expand HR. Right-click on HR and select Generate ERD (Beta).

pgAdmin    File ▾    Object ▾    Tools ▾    Help ▾

Browser   

Servers (1)   

postgres    HR

Databases (2)

HR (highlighted with a red box)

Context menu for HR database:

- Create
- Refresh...
- Delete/Drop
- CREATE Script
- Disconnect Database...
- Generate ERD (Beta) (highlighted with a blue box and a cursor icon)
- Maintenance...
- Backup...
- Restore...
- Grant Wizard...
- Search Objects...
- Query Tool
- Properties...

postgres

Casts Catalogs Event Triggers Extensions Foreign Databases Languages Publications Schemas Subscriptions

Login/Group Roles Tablespaces

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.

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser

Servers (1)

postgres

Databases (2)

HR

Casts

Catalogs

Dashboard Properties SQL

Add table

Option Ctrl

The screenshot shows the pgAdmin interface. On the left, a tree view displays a single server named 'postgres' containing two databases: 'HR' and 'Casts'. The 'HR' database is currently selected, indicated by a blue highlight. On the right, a grid-based workspace is visible, and a context menu is open over the 'HR' database entry. The menu includes options like 'Add table' (which is highlighted with a red box), 'Option', and 'Ctrl'. The top navigation bar includes 'File', 'Object', 'Tools', and 'Help' menus, along with tabs for 'Dashboard', 'Properties', and 'SQL'.

## 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/Prec

## New table

General

Columns

### Columns

	Name	Data type
	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

Schema  public

Comment

## Table: jobs (public)

General

Columns

### Columns

	Name	Data type	Length/Precision	Scale
 	job_id	character varying	10	
 	job_title	character varying	35	
 	min_salary	numeric	8	2
 	max_salary	numeric	8	2

▼ [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

## Table: departments (public)

[General](#)
[Columns](#)

### Columns

	Name	Data type	Length/Precision	Scale
	department_id	integer		
	department_name	character varying	30	
	manager_id	integer		
	location_id	integer		

▼ [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

## Table: locations (public)

General Columns

### Columns

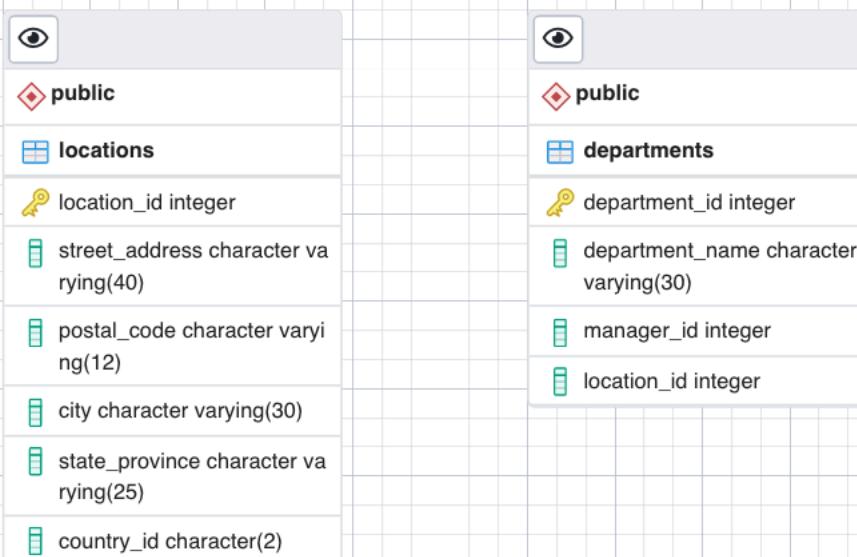
	Name	Data type	Length/Precision
 	location_id	integer	
 	street_address	character varying	40
 	postal_code	character varying	12
 	city	character varying	30
 	state_province	character varying	25
 	country_id	character	2

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

Properties SQL Statistics Dependencies Dependents  Untitled\*



/postgres@localhost



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**.

The screenshot shows a database schema editor interface. At the top, there is a toolbar with various icons. A red box highlights the '1M' button in the toolbar. Below the toolbar, the connection information 'HR/postgres@localhost' is displayed. A tooltip window is open, showing the text 'One-to-Many link' and three keyboard shortcut options: 'Option', 'Ctrl', and 'O'. The main area displays the 'employees' table schema. The table has the following columns:

	public	employees
key	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

The screenshot shows the 'employees' table schema. The table has the following columns:

	public	employees
key	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

The screenshot shows the 'departments' table schema. The table has the following columns:

	public	departments
key	department_id	integer
	department_name	character varying(30)
	manager_id	integer

## One to many relation

### General

Local Table (public) employees

Local Column department\_id

Referenced Table (public) departments

Referenced  
Column department\_id

 Cancel

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

▼ [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

▼ [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

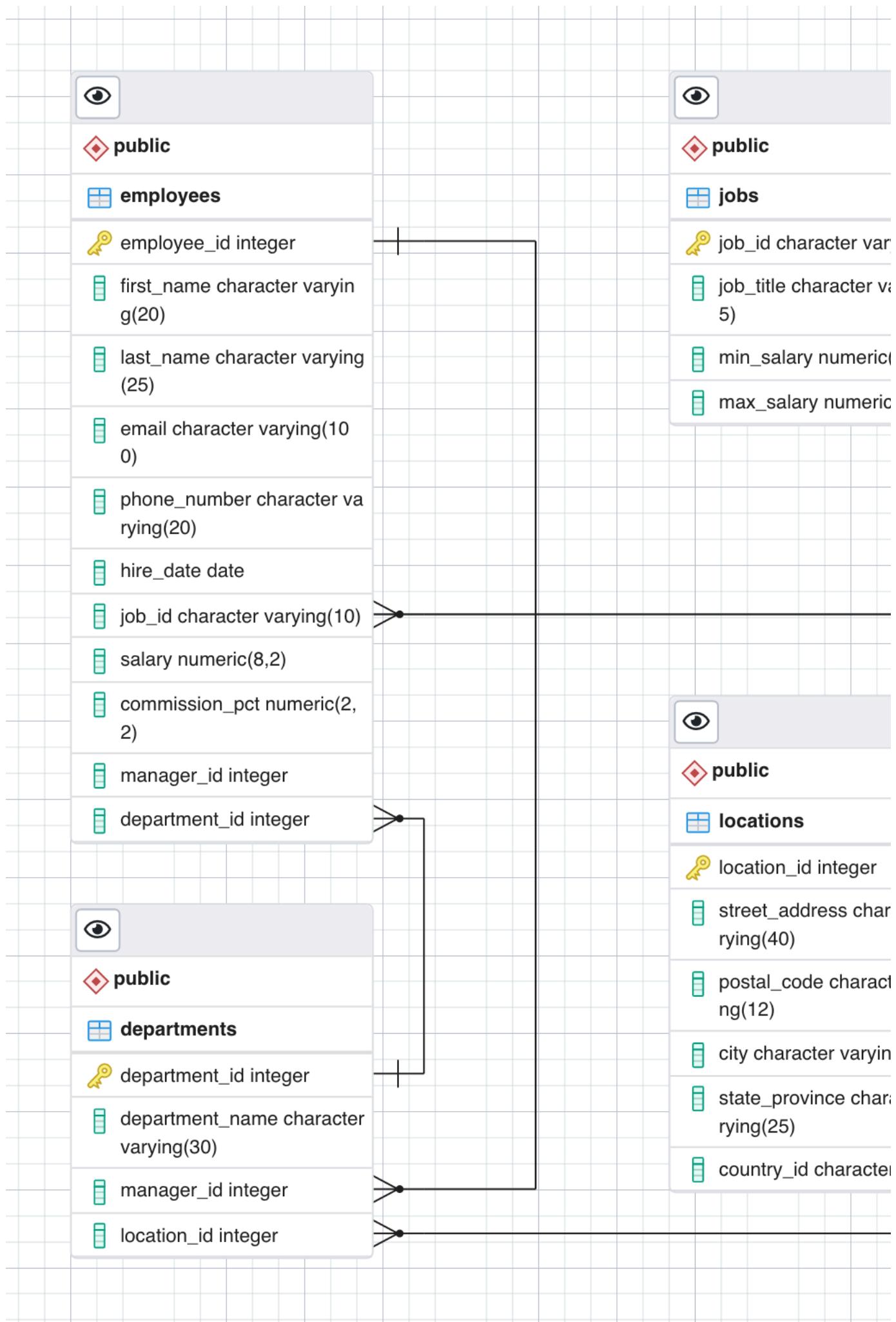
Local Column manager\_id

Referenced Table (public) employees

Referenced Column employee\_id

 Cancel

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**.

The screenshot shows a database diagram interface with two tables displayed:

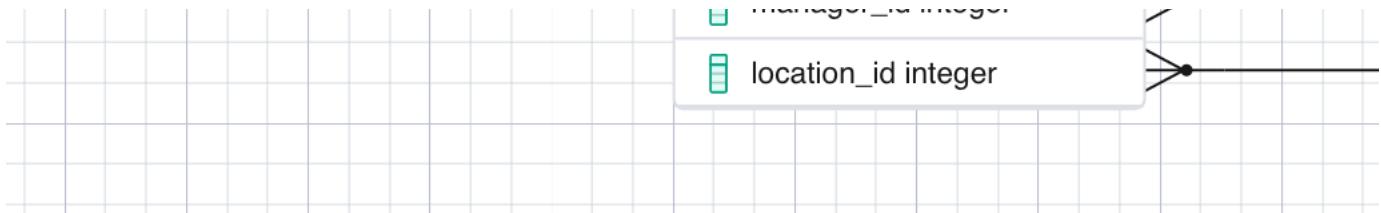
**employees**

- employee\_id integer (Primary Key)
- 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) (Foreign Key to job\_id in departments)
- salary numeric(8,2)
- commission\_pct numeric(2,2)
- manager\_id integer (Foreign Key to manager\_id in departments)
- department\_id integer (Foreign Key to department\_id in departments)

**departments**

- department\_id integer (Primary Key)
- department\_name character varying(30)
- manager\_id integer (Foreign Key to manager\_id in employees)

A red box highlights the "Generate SQL" button in the toolbar.



location\_id integer

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.



Query Editor    Query History

```
1 -- This script was generated by a beta version of the ER
2 -- Please log an issue at https://redmine.postgresql.org
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.

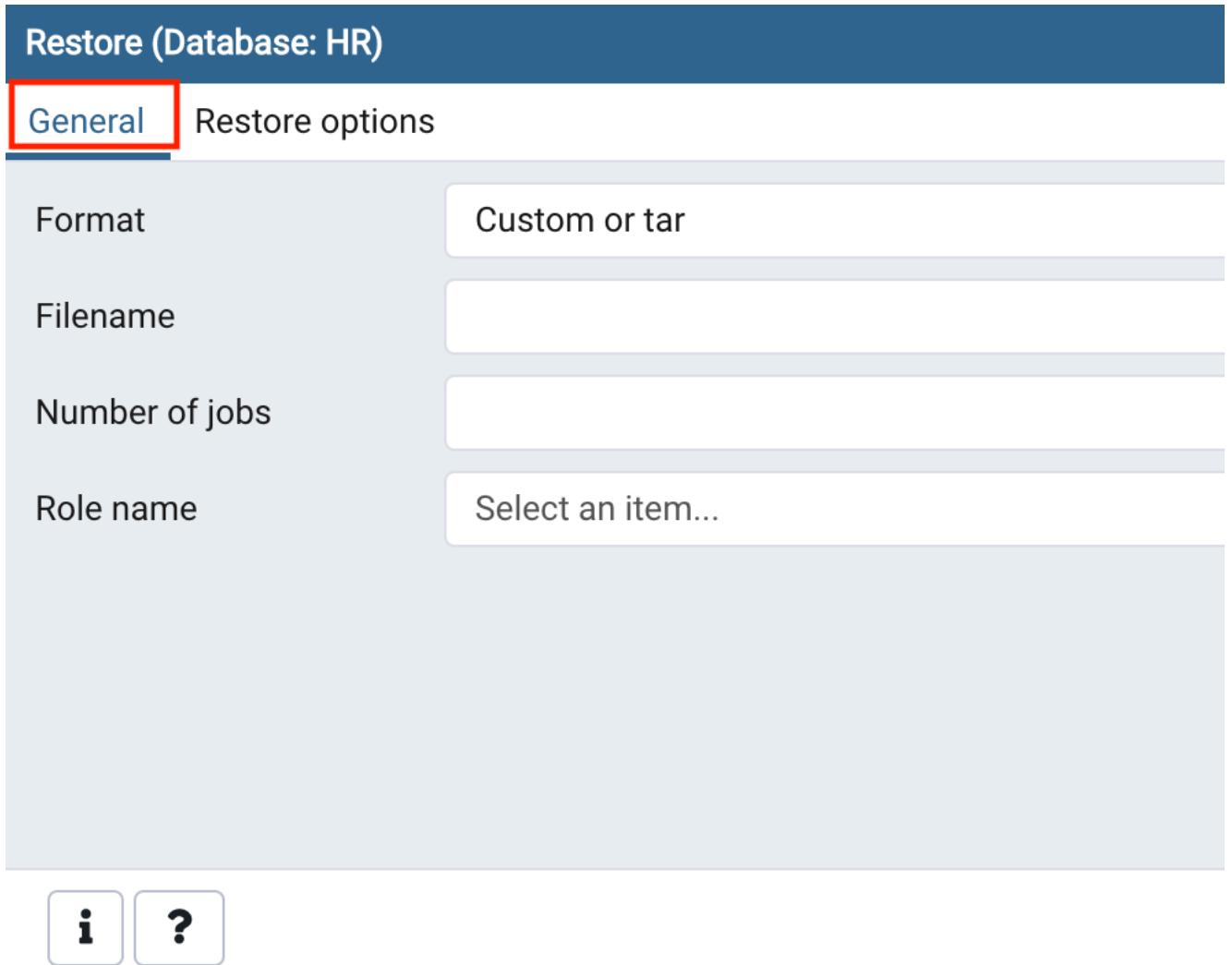
- [HR\\_pgsql\\_dump\\_data\\_for\\_example-exercise.tar](#)

2. Follow the instructions below to import/restore the data:

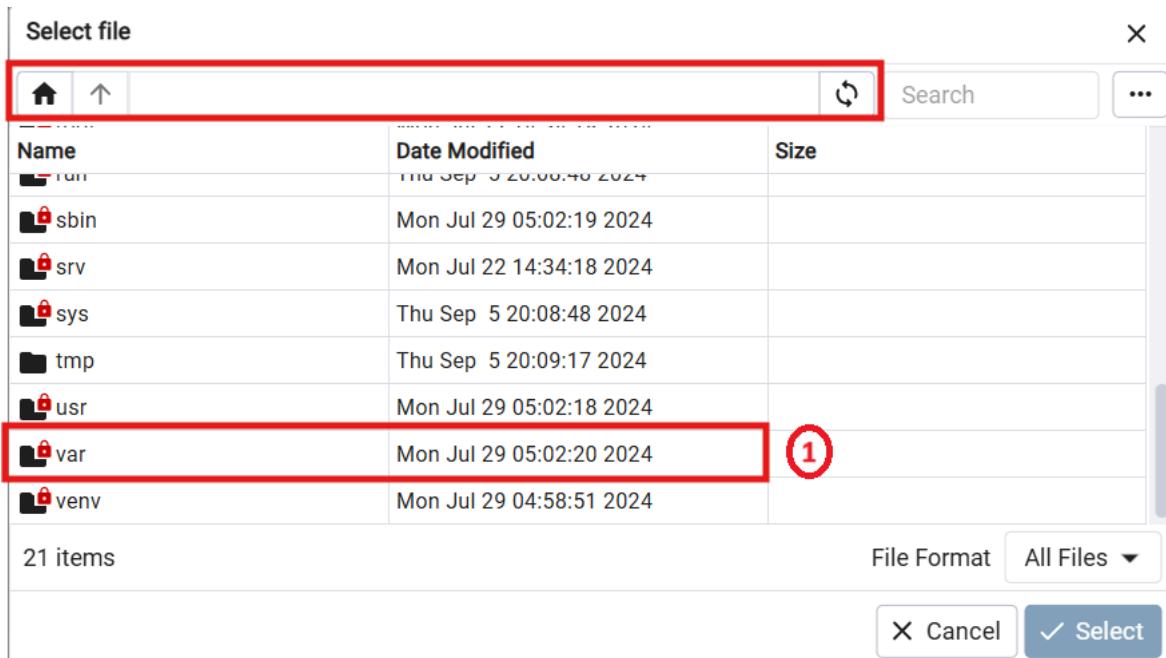
- In the tree-view, expand **HR**. Right-click **HR** and click **Restore**.

The screenshot shows the pgAdmin 4 interface. The top navigation bar includes File, Object, Tools, and Help. Below the header are tabs for Browser, Dashboard, and Properties. The Browser tab is active, displaying a tree-view of database objects. Under the 'Servers' node, the 'postgres' server is expanded, showing the 'Databases' node which contains the 'HR' database, highlighted with a red box. The 'HR' database node is also highlighted with a blue selection bar. A context menu is open over the 'HR' node, listing options: Create, Refresh..., Delete/Drop, CREATE Script, Disconnect Database..., Generate ERD (Beta), Maintenance..., Backup..., Restore... (which is highlighted with a dark blue background), Grant Wizard..., Search Objects..., Query Tool, and Properties... .

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



- 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 lib folder.

Select file X

Home Up /var Search ...

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	
<b>lib</b>	Mon Jul 29 05:02:26 2024	
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.

Select file X

Home Up /var/lib Search ...

Name	Date Modified	Size
misc	Mon Jul 22 14:34:18 2024	
<b>pgadmin</b>	Fri Sep 6 01:00:10 2024	
postfix	Thu Sep 5 20:09:12 2024	
sudo	Mon Jul 29 05:02:20 2024	

4 items File Format All Files ▾

X Cancel ✓ Select

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

Select file X

Home Up /var/lib/pgadmin Search ...

Name	Date Modified	Size
azurecredentialcache	Thu Sep 5 20:08:53 2024	
pgadmin4.db	Fri Sep 6 01:04:34 2024	164.0 kB
sessions	Thu Sep 5 23:43:26 2024	
storage	Thu Sep 5 20:08:53 2024	

1 ...

2
Upload

✓ List View
Grid View

Show Hidden Files

4 items File Format All Files ▾

X Cancel ✓ Select

- o 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/

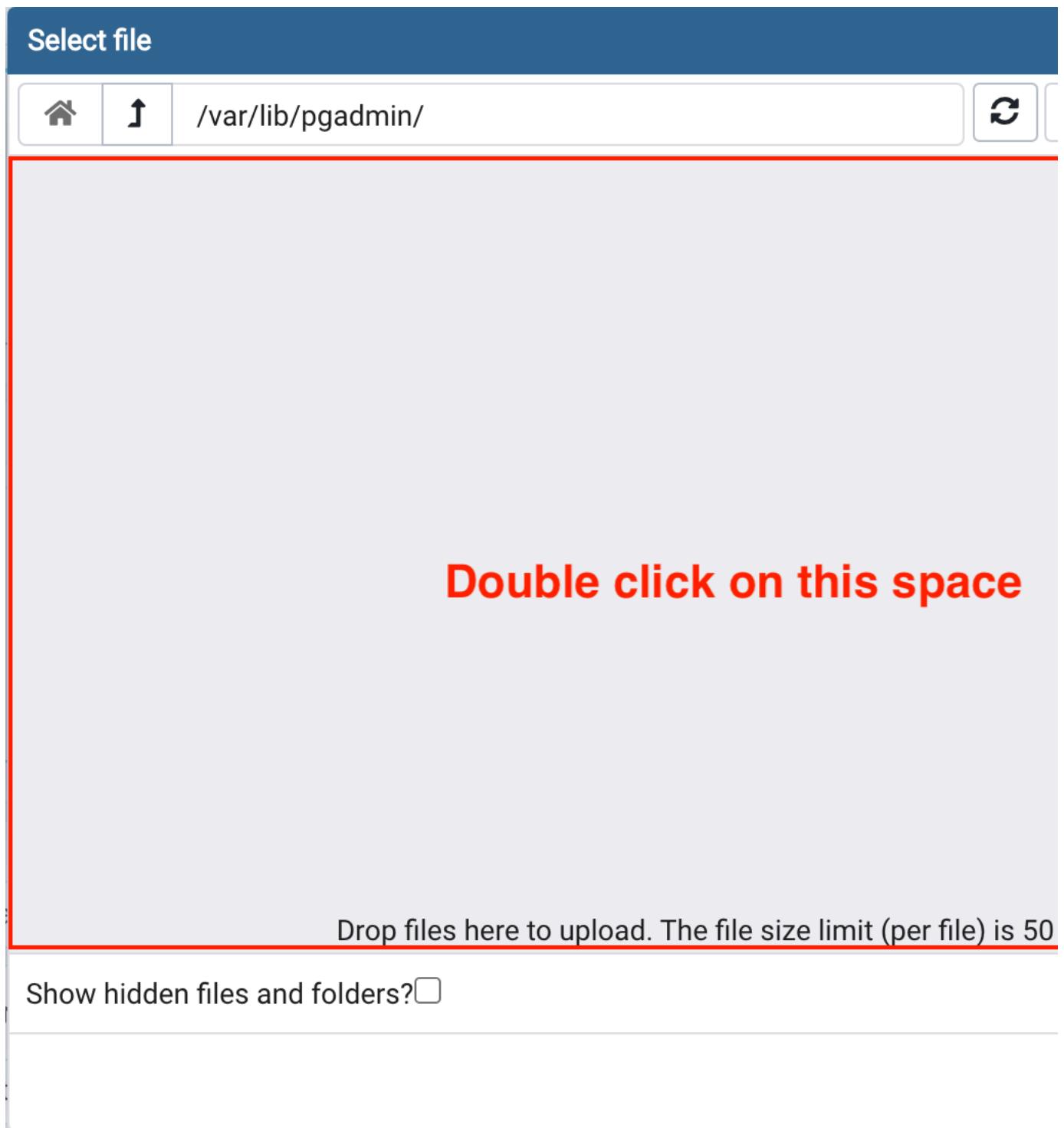
Select file

/var/lib/pgadmin/

Double click on this space

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

Show hidden files and folders?

A screenshot of a web-based file upload interface titled "Select file". At the top, there's a breadcrumb navigation showing the path "/var/lib/pgadmin/". Below the path is a large, empty rectangular area with a red border, intended for file uploads. In the center of this area, the text "Double click on this space" is displayed in red. At the bottom of the upload area, the instruction "Drop files here to upload. The file size limit (per file) is 50" is shown. Below the upload area, there's a checkbox labeled "Show hidden files and folders?".

- o 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

Show hidden files and folders?

- o 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
HR_pgsql_dump_data_for_example-exercise.tar	20.5 kB
pgadmin4.db	156.0 kB
sessions	4.0 kB
storage	4.0 kB

Show hidden files and folders?

- 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_exai
Number of jobs	
Role name	Select an item...

**i** **?**

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

### General    **Restore options**

**Queries**

Include CREATE DATABASE statement	<input type="button" value="No"/>	Clean before restore
Single transaction	<input type="button" value="No"/>	

**Disable**

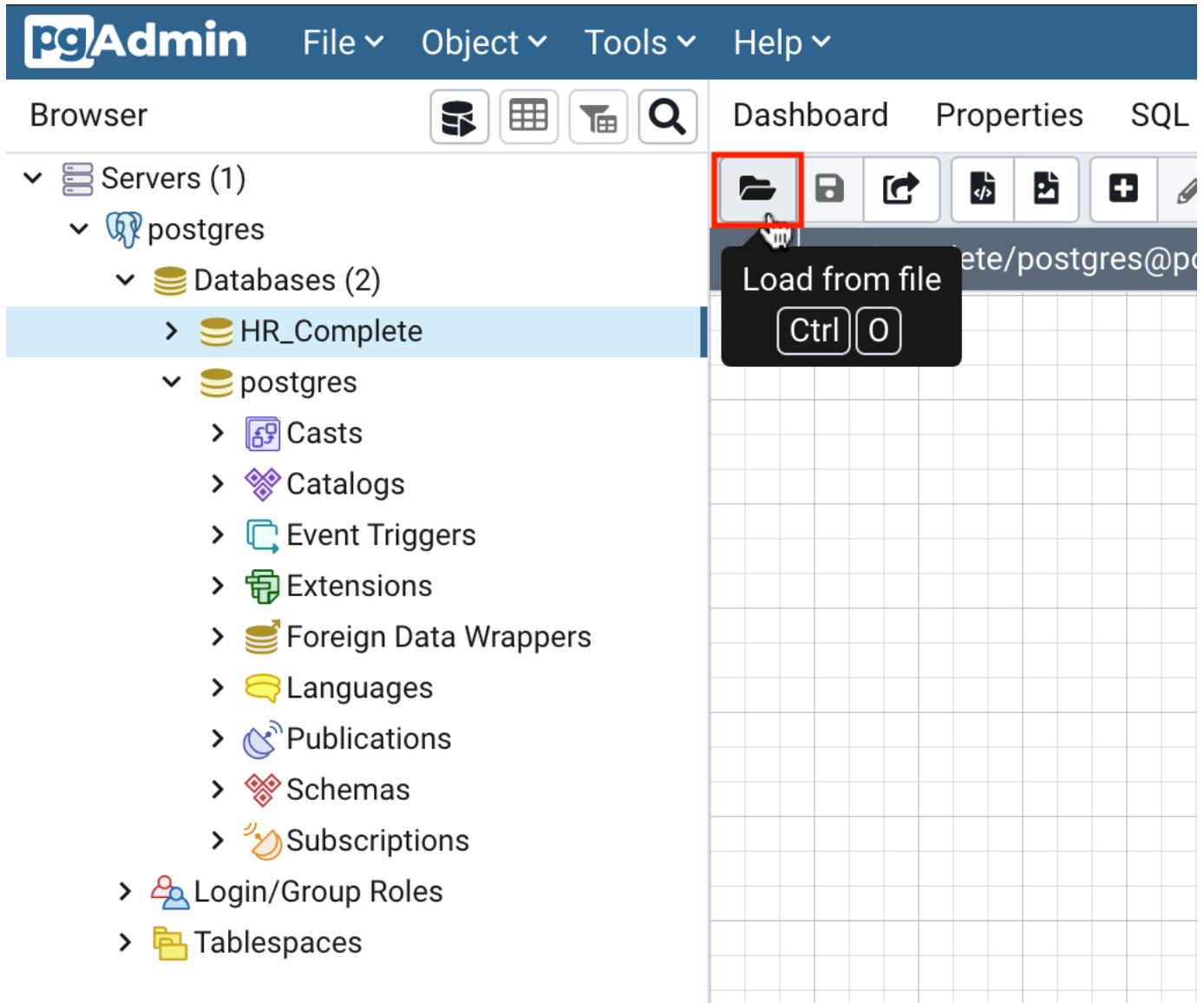
Trigger	<input checked="" type="button" value="Yes"/>	No data for Failed Tables
---------	---	---------------------------

**i** **?**

## 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.

1. 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.
  - o [HR\\_pgsql\\_ERD\\_for\\_practice-exercise.pgerd](#)
2. In pgAdmin, create a new database named **HR\_Complete**.
3. Open the ERD Tool and use **Load from file** to load the [HR\\_pgsql\\_ERD\\_for\\_practice-exercise.pgerd](#) file.



**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   

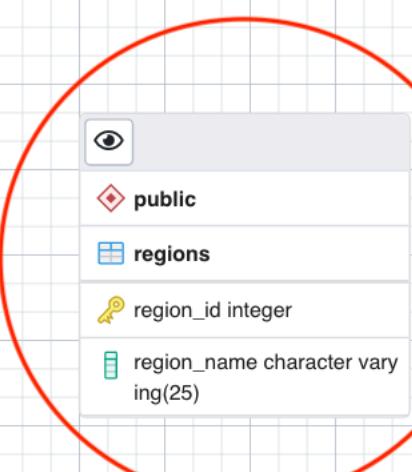
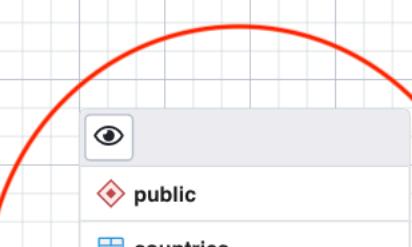
Databases (2)   

HR\_Complete   

postgres    Login/Group Roles    Tablespaces

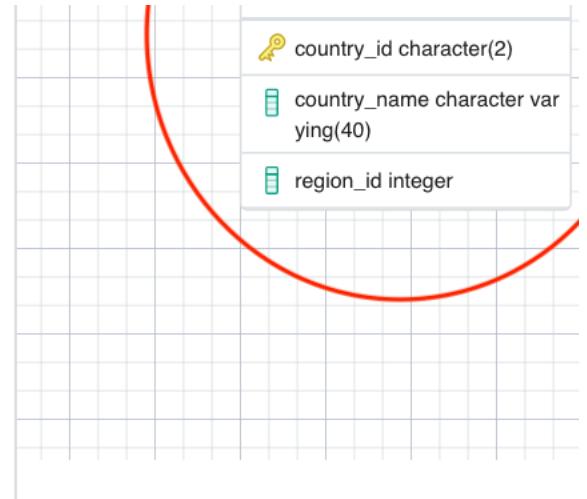
Dashboard    Properties    SQL

HR\_Complete/postgres@pc

	public
	regions
	region_id integer
	region_name character varying(25)

	public
	countries



5. Create the remaining relationships between the tables:

▼ [Click here] Create a relationship between countries and regions

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

## One to many relation

### General

Local Table	(public) countries
Local Column	region_id
Referenced Table	(public) regions
Referenced Column	region_id

X

▼ [Click here] Create a relationship between job\_history and departments

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

## One to many relation

### General

Local Table

(public) job\_history

Local Column

department\_id

Referenced Table

(public) departments

Referenced

department\_id

Column



- ▼ [Click here] Create a relationship between job\_history and employees

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

## One to many relation

### General

Local Table

(public) job\_history

Local Column

employee\_id

Referenced Table

(public) employees

Referenced

employee\_id

Column



- ▼ [Click here] Create a relationship between job\_history and jobs

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

## One to many relation

### General

Local Table

(public) job\_history

Local Column

job\_id

Referenced Table

(public) jobs

Referenced

job\_id

Column



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

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

## One to many relation

### General

Local Table (public) locations

Local Column country\_id

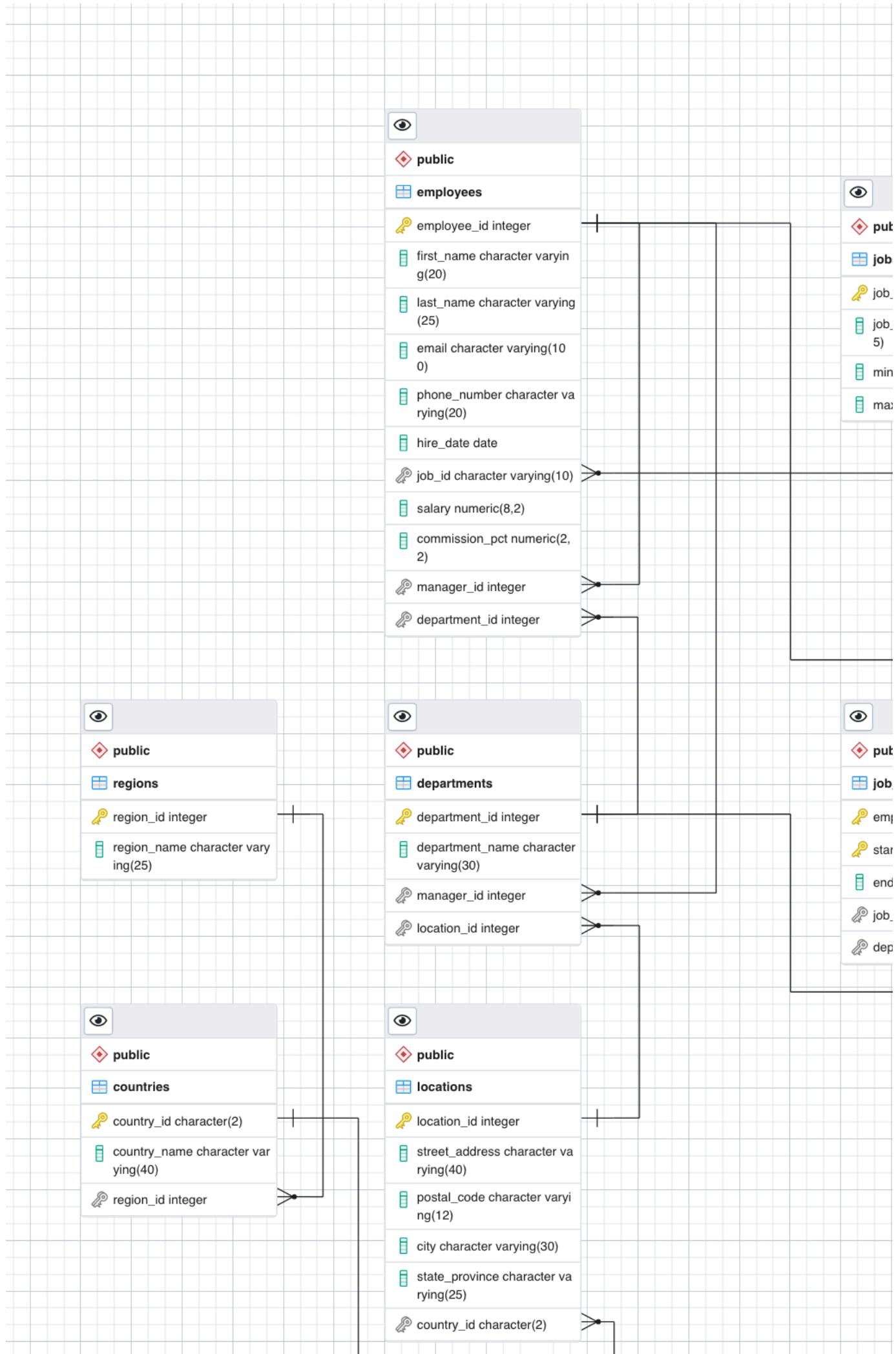
Referenced Table (public) countries

Referenced Column country\_id



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

- 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.

- o [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.