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.

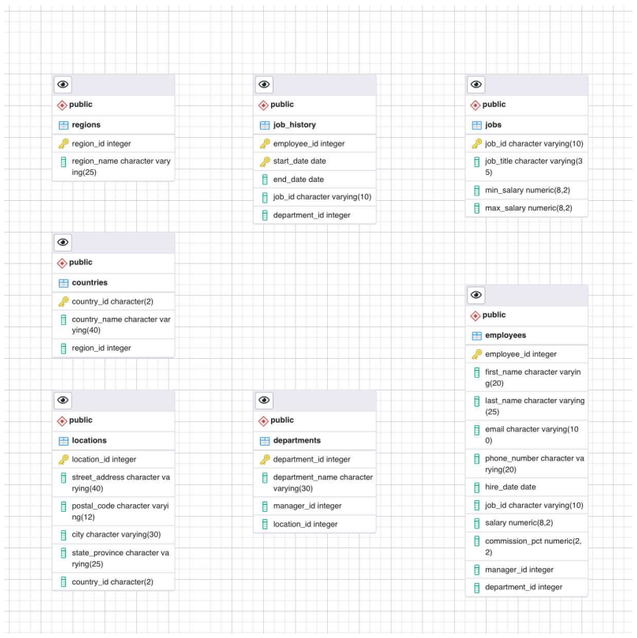
To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

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*.

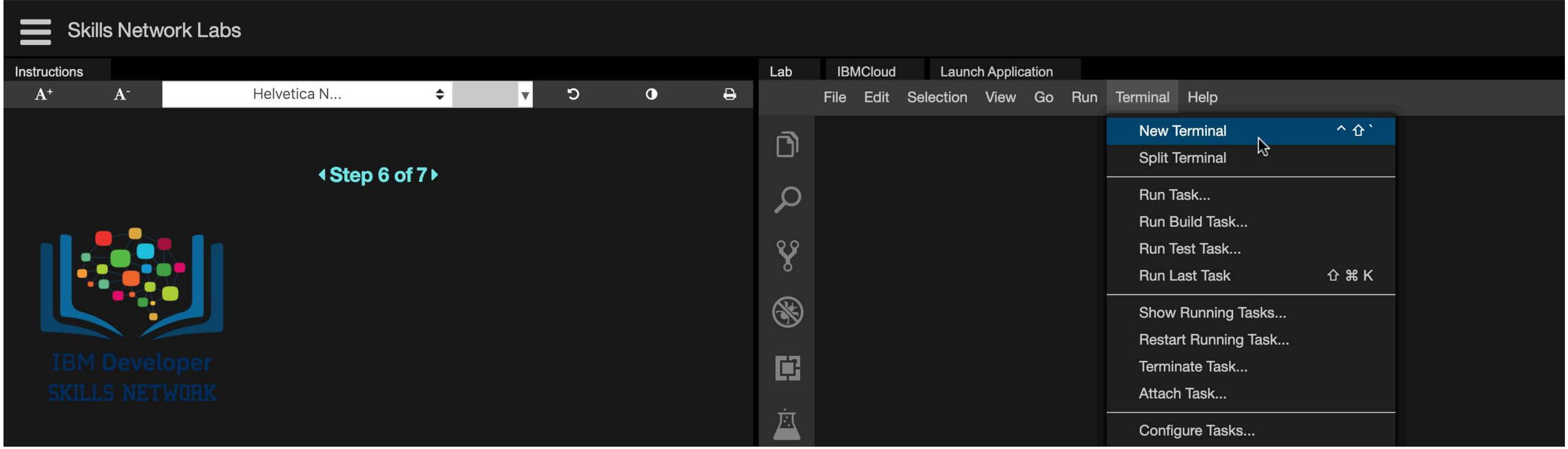
Example Exercise

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.

1. Go to **Terminal > New Terminal** to open a terminal from the side-by-side launched Cloud IDE.



2. Start a PostgreSQL service session in the Cloud IDE using the command given below. Find your PostgreSQL service session password from the highlighted location of the terminal shown in the image below. Note down your PostgreSQL service session password because you may need to use it later in the lab.

```
1. 1
1. start_postgres
```

Copied!

```
theia@theiadocker-sandipsahajo:/home/project$ start_postgres
Starting your Postgres database....
This process can take up to a minute.

Postgres database started, waiting for all services to be ready....
 [/]
Your Postgres database is now ready to use and available with username: postgres password: MTQ5NTItc2FuZGlw

You can access your Postgres database via:
 • The Browser with pgadmin
   • URL: https://sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/
   • Database Password: MTQ5NTItc2FuZGlw
 • CommandLine: psql --username=postgres --host=localhost
theia@theiadocker-sandipsahajo:/home/project$
```

3. Copy your pgAdmin weblink from the highlighted location of the terminal shown in the image below and paste it to a new tab of your web browser.

```
theia@theiadocker-sandipsahajo:/home/project$ start_postgres
Starting your Postgres database....
This process can take up to a minute.

Postgres database started, waiting for all services to be ready....
 [/]
Your Postgres database is now ready to use and available with username: postgres password: MTQ5NTItc2FuZGlw

You can access your Postgres database via:
 • The Browser with pgadmin
   • URL: https://sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/
   • Database Password: MTQ5NTItc2FuZGlw
 • CommandLine: psql --username=postgres --host=localhost
theia@theiadocker-sandipsahajo:/home/project$
```

4. You will see the pgAdmin GUI tool.

sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/

pgAdmin File Object Tools Help

Browser

Servers

Dashboard Properties SQL Statistics Dependencies Dependents

Welcome

pgAdmin

Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural c
is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

Add New Server

Configure pgAdmin

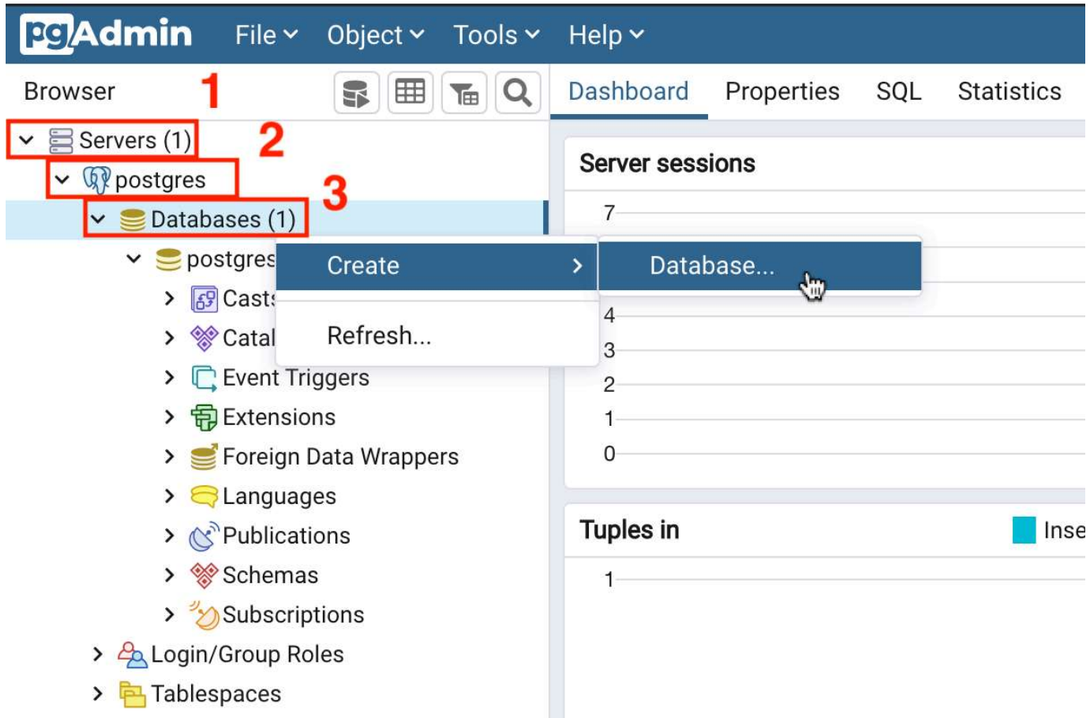
Getting Started

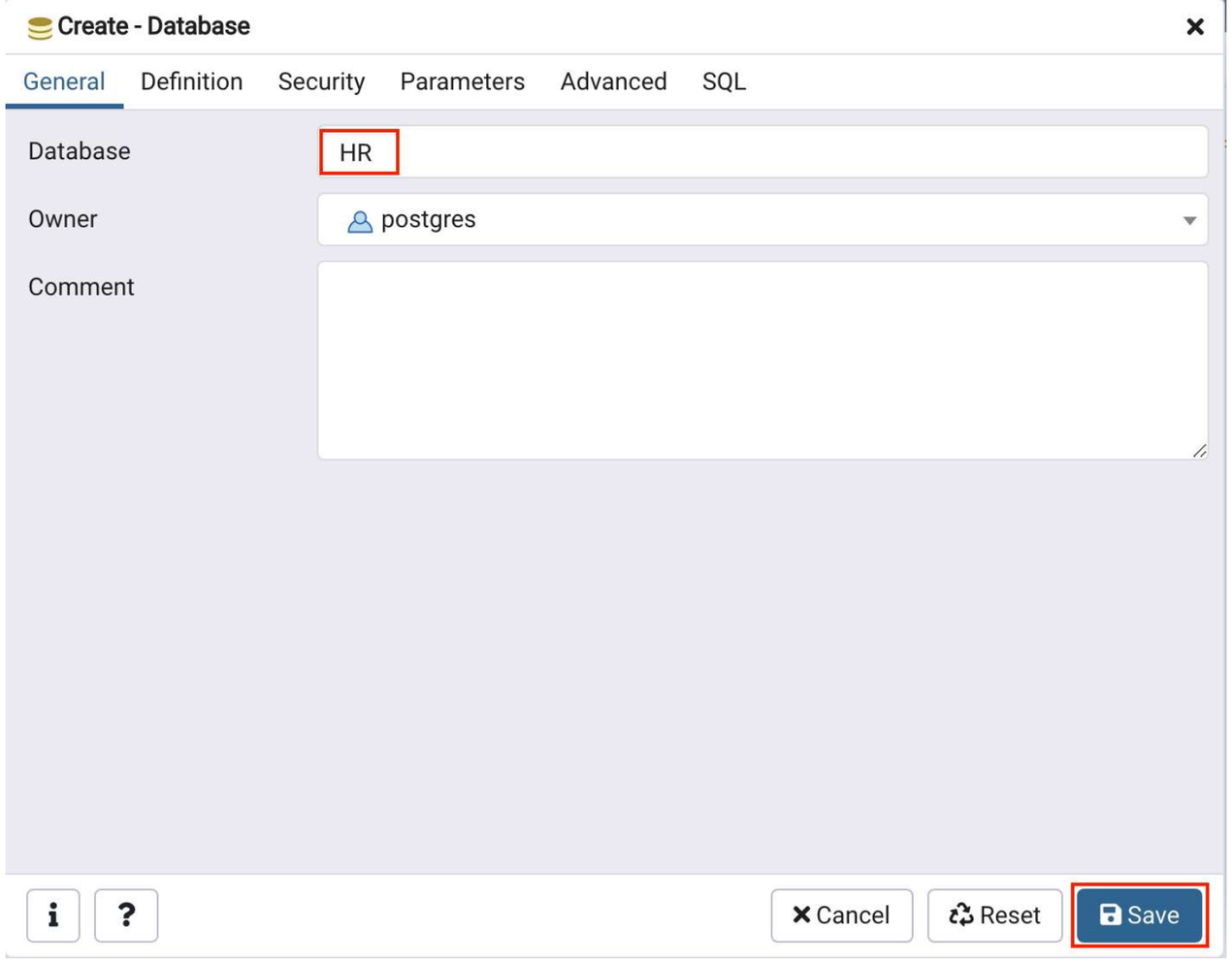
PostgreSQL Documentation

pgAdmin Website

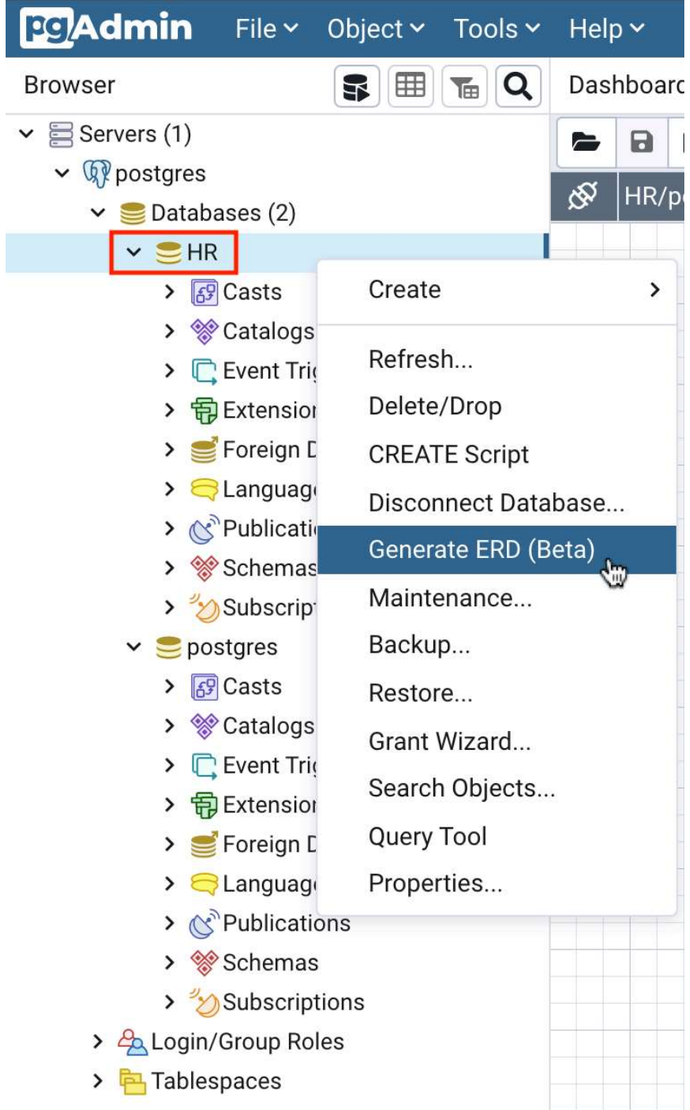
Planet PostgreSQL

5. 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**.

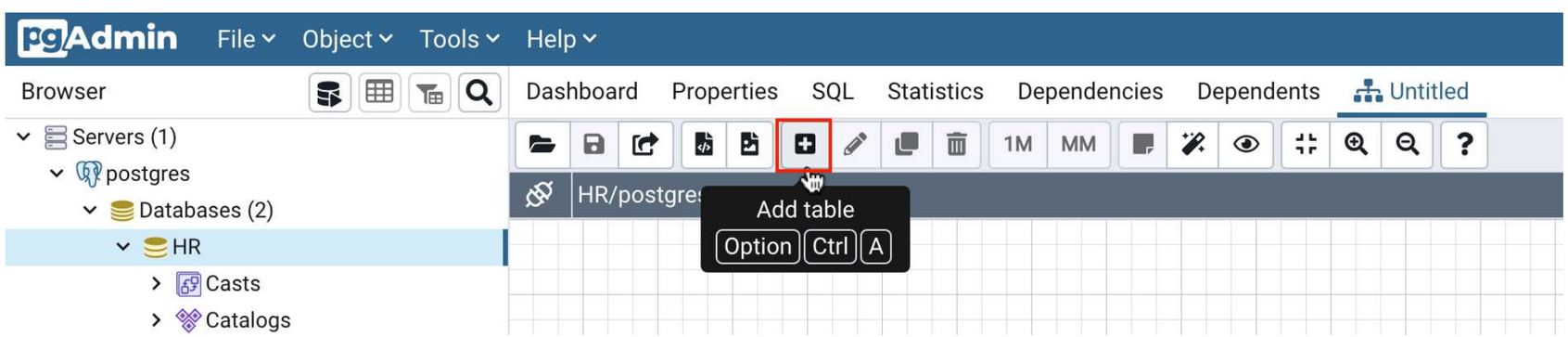


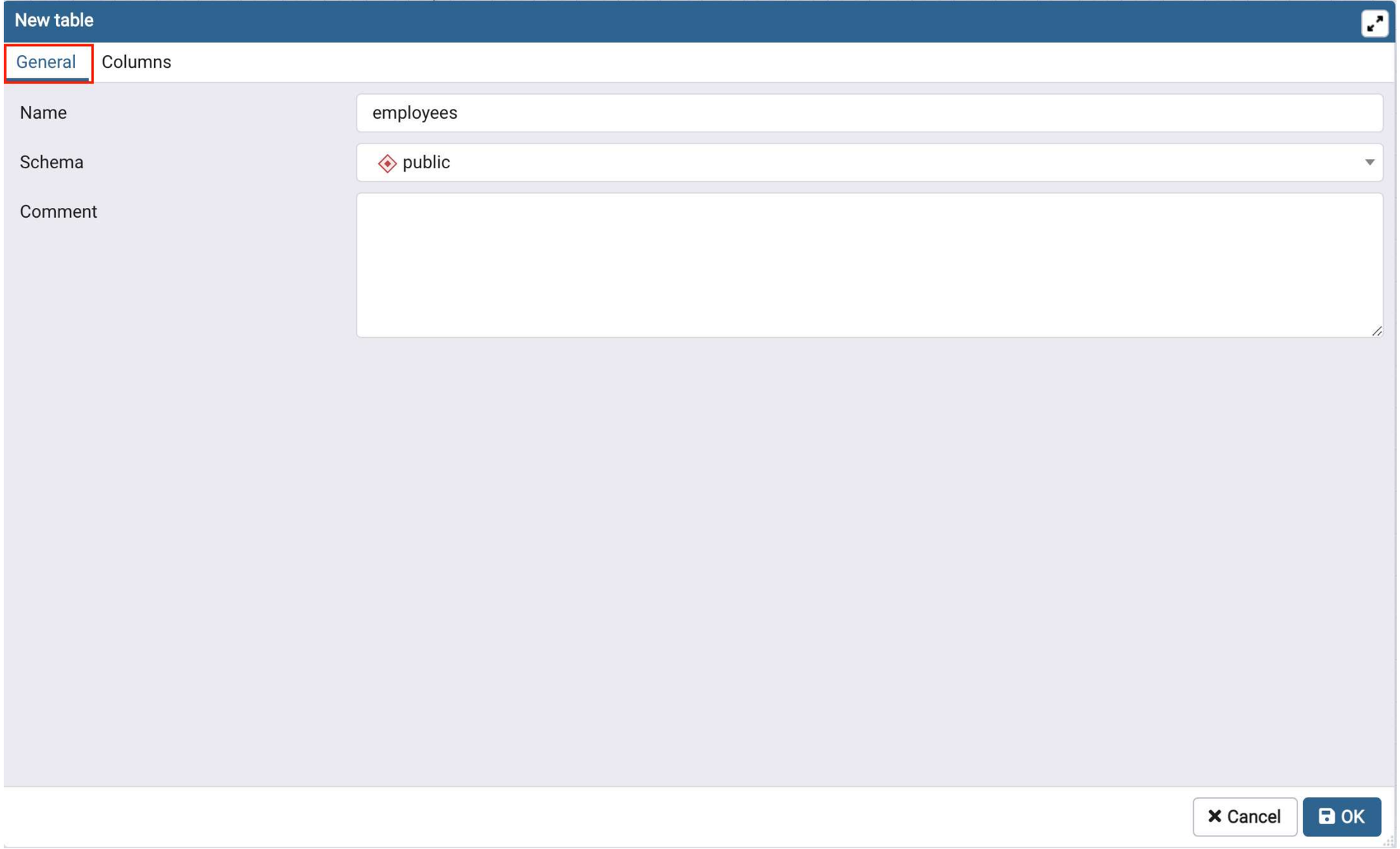


6. In the tree-view, expand **HR**. Right-click on **HR** and select **Generate ERD (Beta)**.



7. 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.





8. 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	Not NULL?	Primary key?

Cancel | OK

New table

General | Columns

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	employee_id	integer			Yes	Yes
✎	🗑	first_name	character varying	20		No	No
✎	🗑	last_name	character varying	25		Yes	No
✎	🗑	email	character varying	100		Yes	No
✎	🗑	phone_number	character varying	20		No	No
✎	🗑	hire_date	date			Yes	No
✎	🗑	job_id	character varying	10		Yes	No
✎	🗑	salary	numeric	8	2	Yes	No
✎	🗑	commission_pct	numeric	2	2	No	No
✎	🗑	manager_id	integer			No	No
✎	🗑	department_id	integer			No	No

✖ Cancel | OK

9. Similarly, create entity diagrams for the other three tables following steps 7 and 8:

▼ [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 | OK

Table: jobs (public)

General | Columns

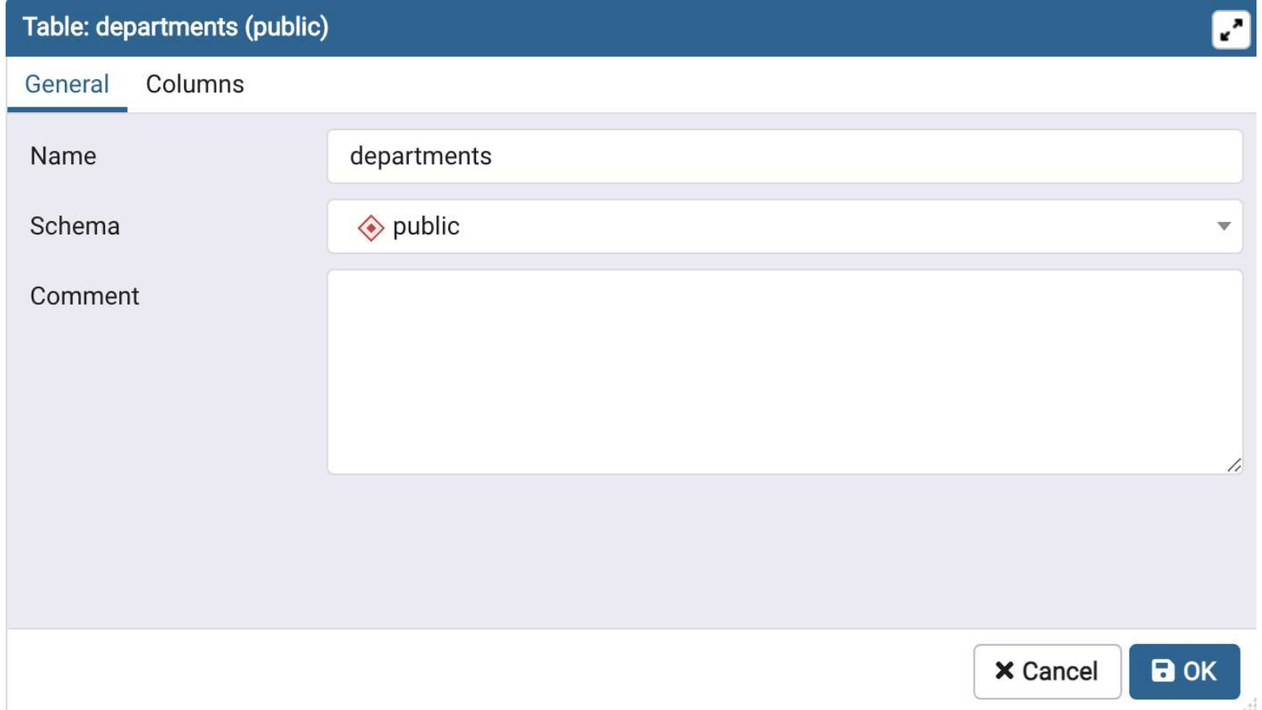
Columns

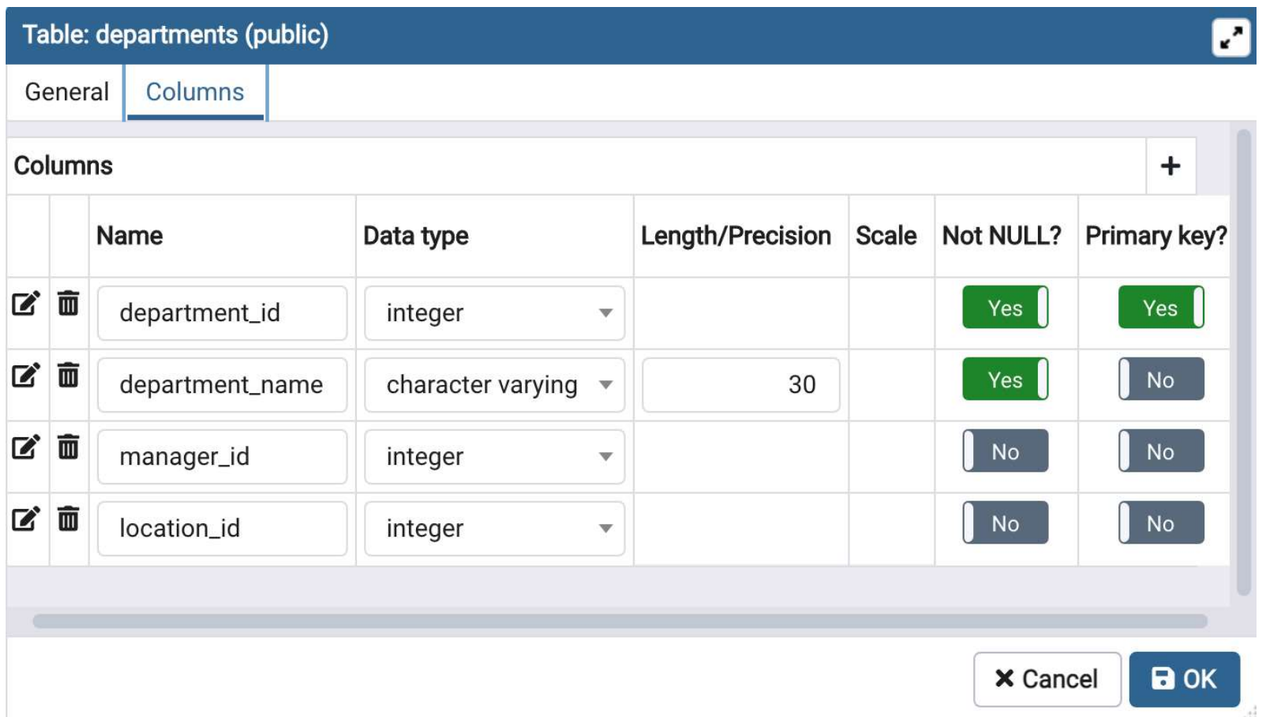
		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	job_id	character varying	10		Yes	Yes
✎	🗑	job_title	character varying	35		Yes	No
✎	🗑	min_salary	numeric	8	2	No	No
✎	🗑	max_salary	numeric	8	2	No	No

Cancel | OK

▼ [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**.





▼ [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 | OK

Table: locations (public)

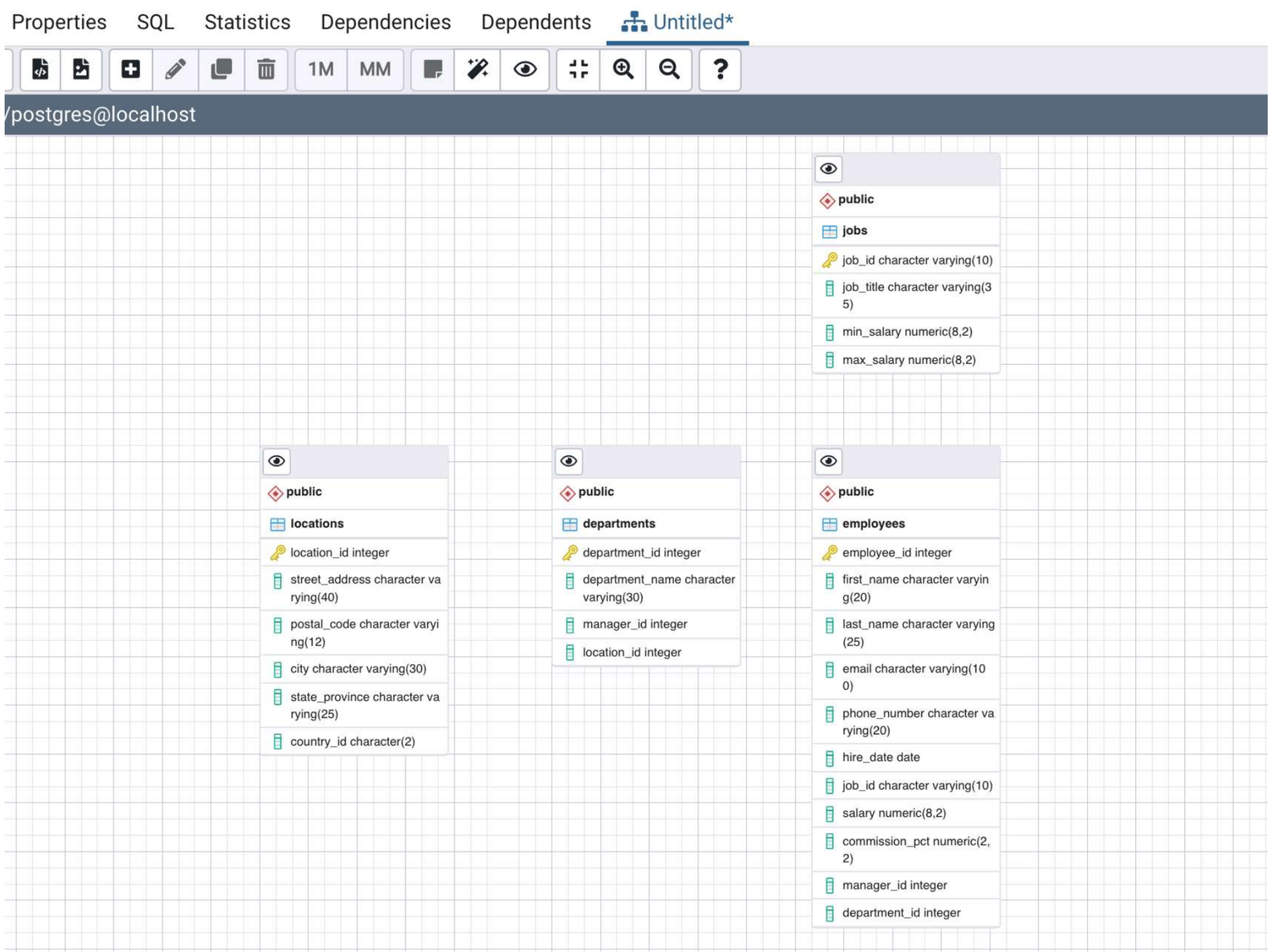
General | Columns

Columns +

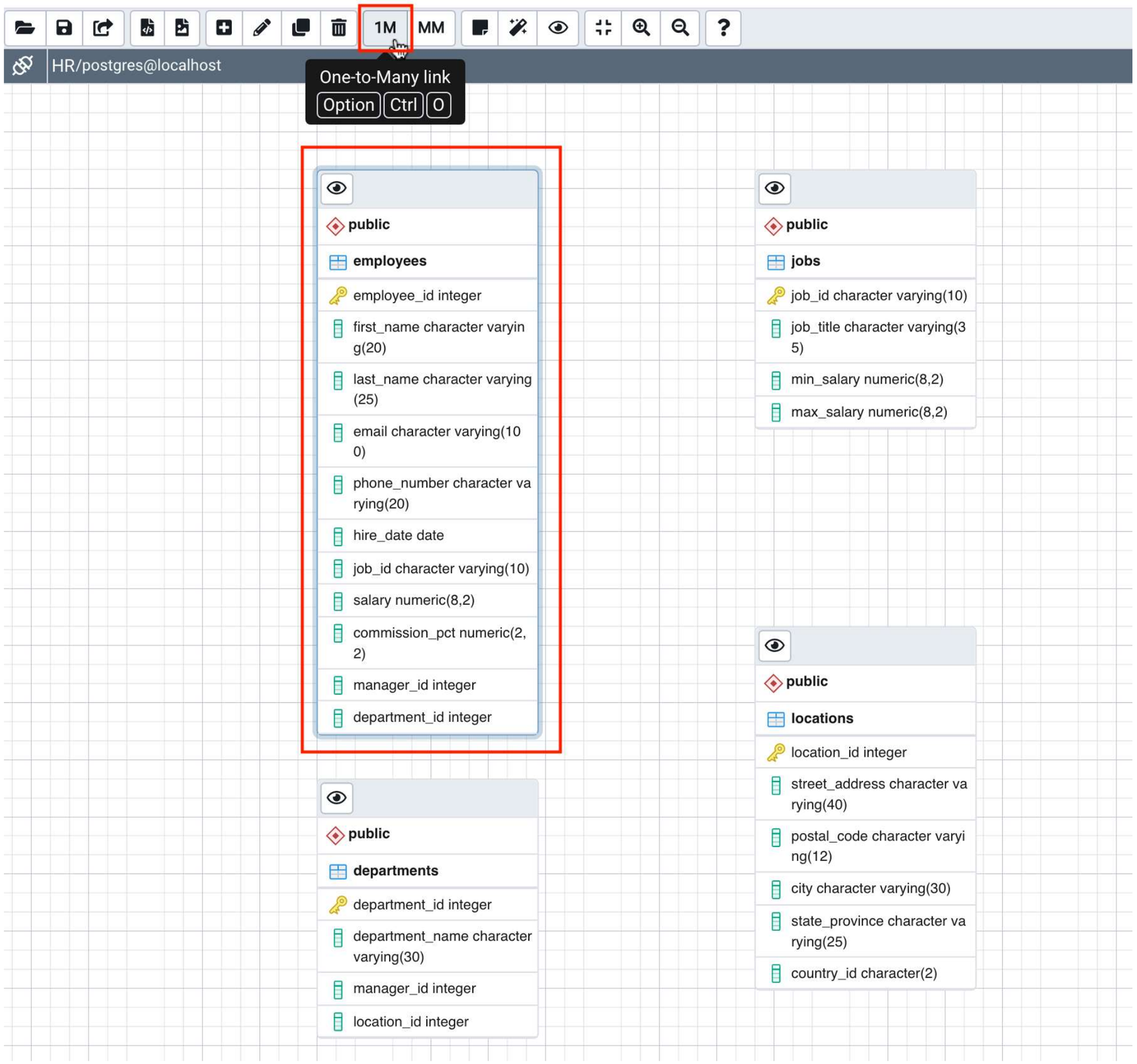
		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	location_id	integer			Yes	Yes
✎	🗑	street_address	character varying	40		No	No
✎	🗑	postal_code	character varying	12		No	No
✎	🗑	city	character varying	30		Yes	No
✎	🗑	state_province	character varying	25		No	No
✎	🗑	country_id	character	2		Yes	No

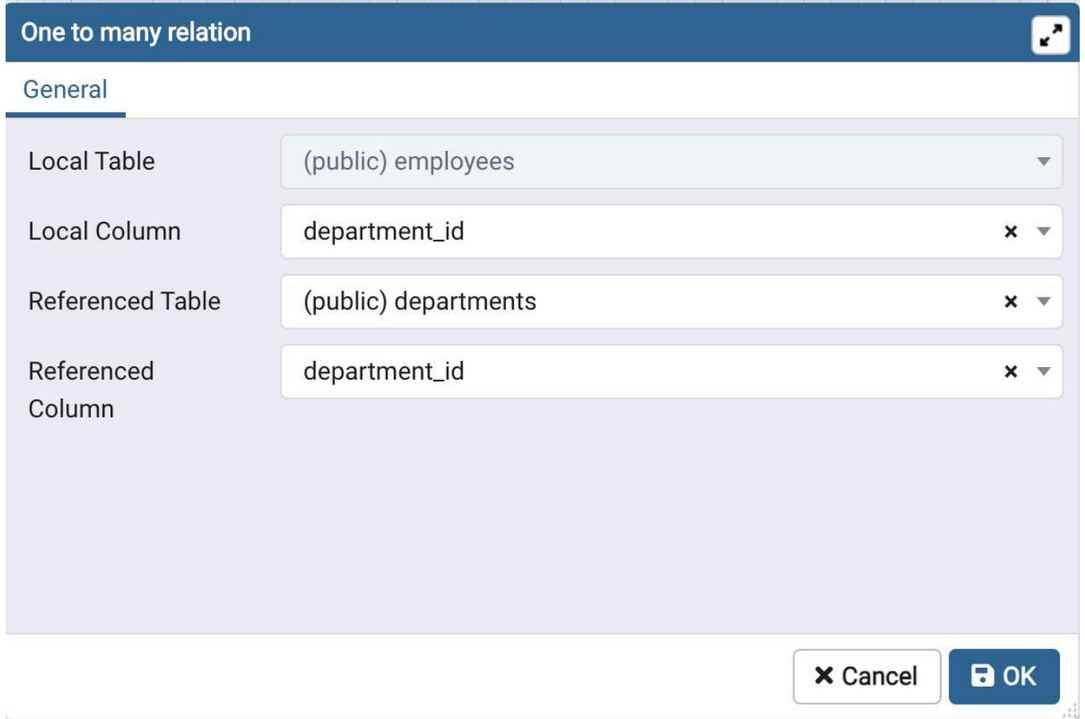
Cancel | OK

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



11. 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**.


1M
MM
HR/postgres@localhost
One-to-Many link
Option
Ctrl
O
public
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
public
jobs
job_id character varying(10)
job_title character varying(35)
min_salary numeric(8,2)
max_salary numeric(8,2)
public
departments
department_id integer
department_name character varying(30)
manager_id integer
location_id integer
public
locations
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)



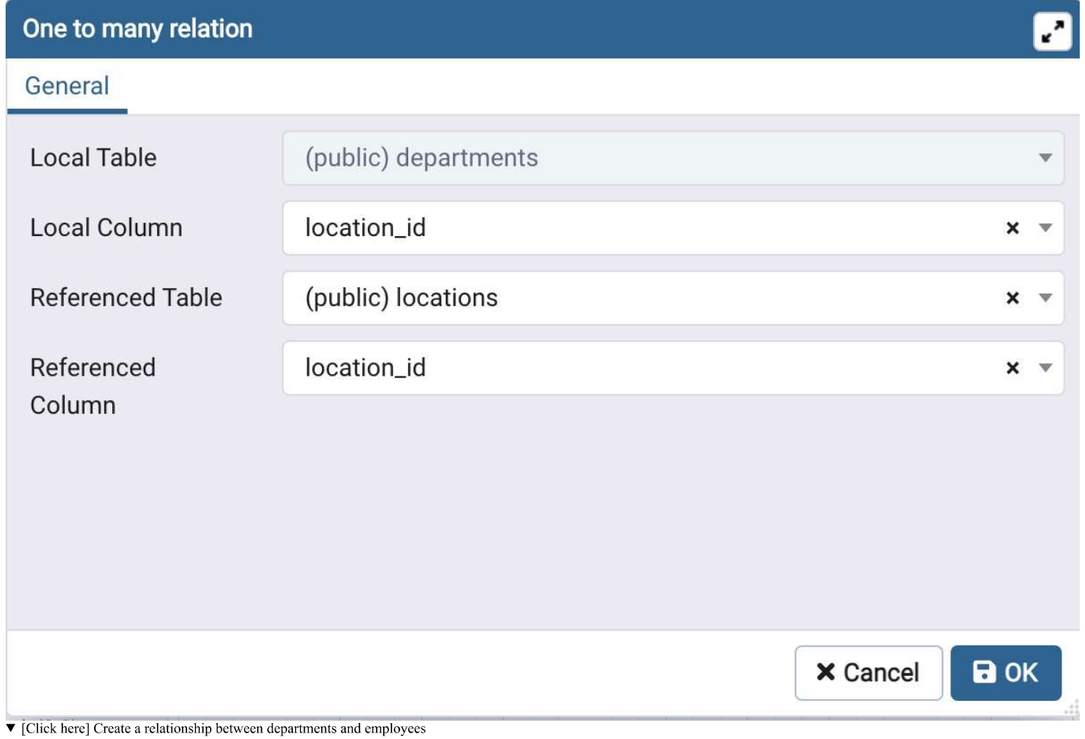
12. Similarly, create the other relationships between the tables following the instructions in step 11:

▼ [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**.

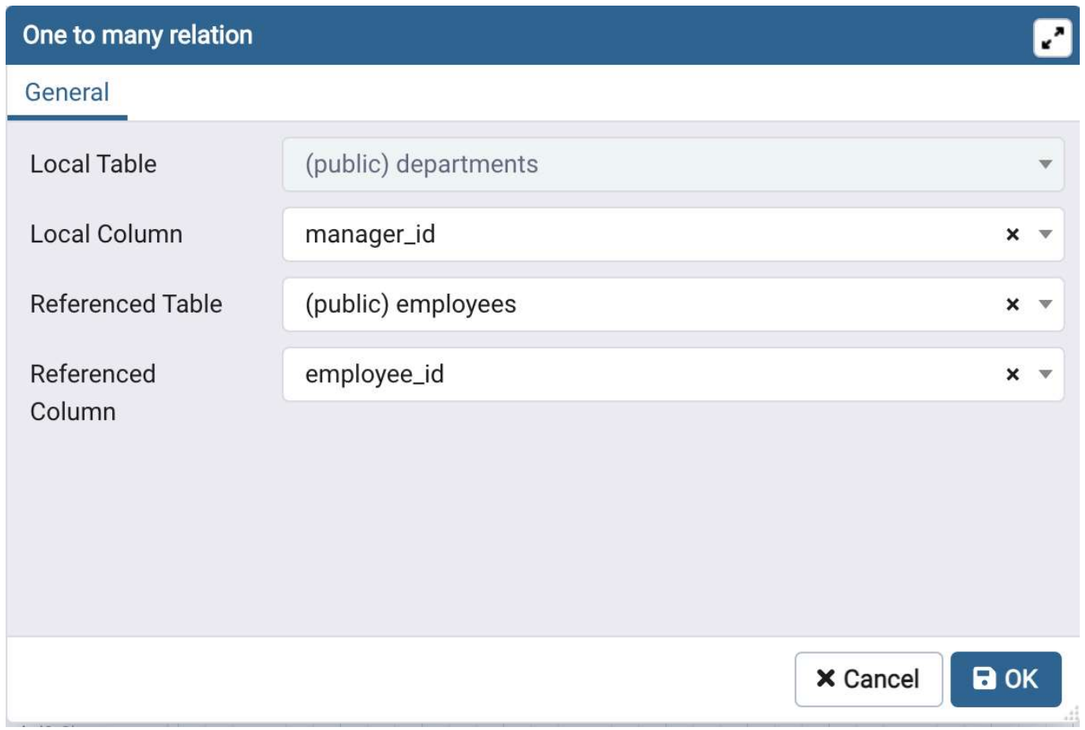
▼ [Click here] Create a relationship between departments and locations

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**.

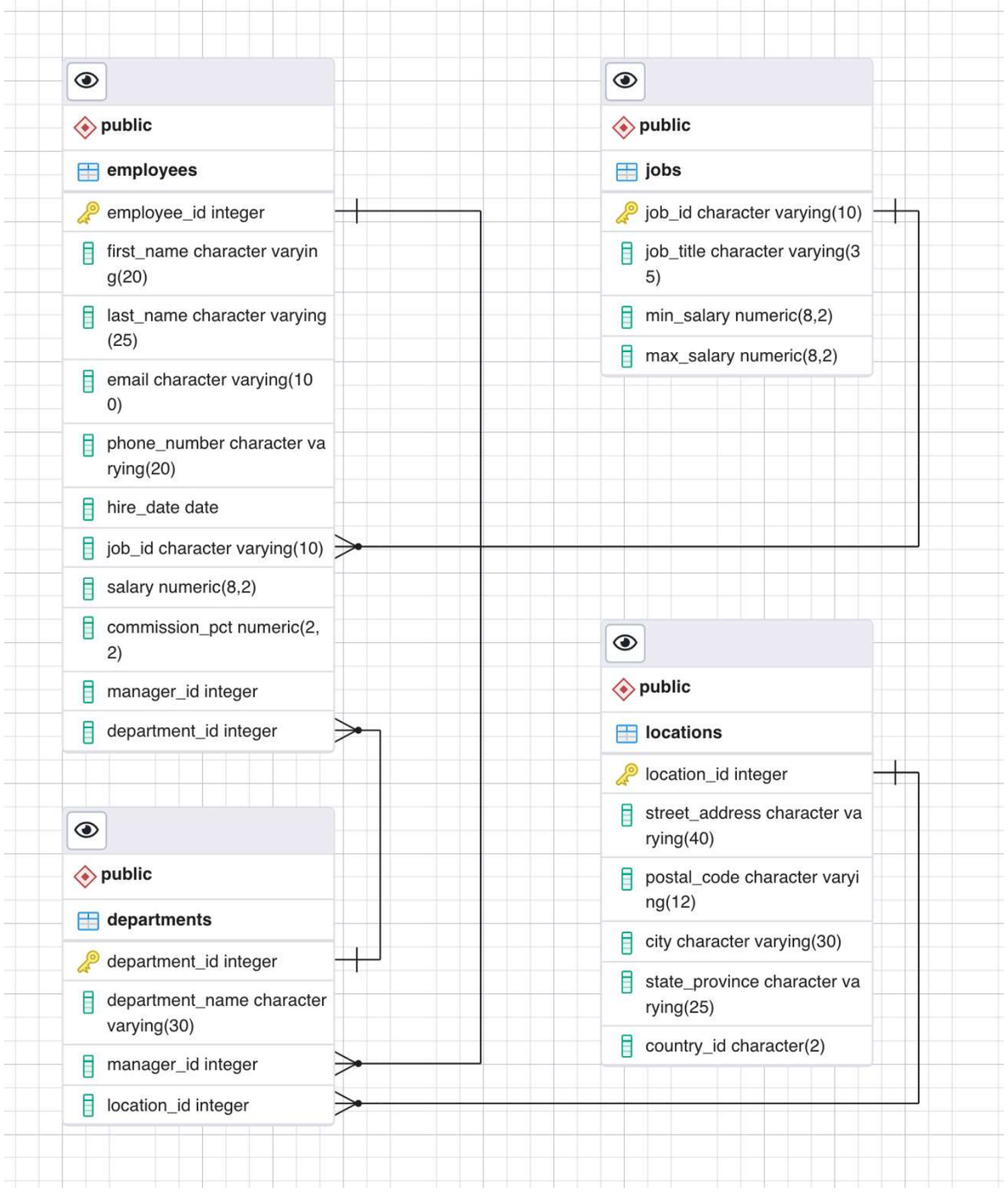


▼ [Click here] Create a relationship between departments and employees

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**.



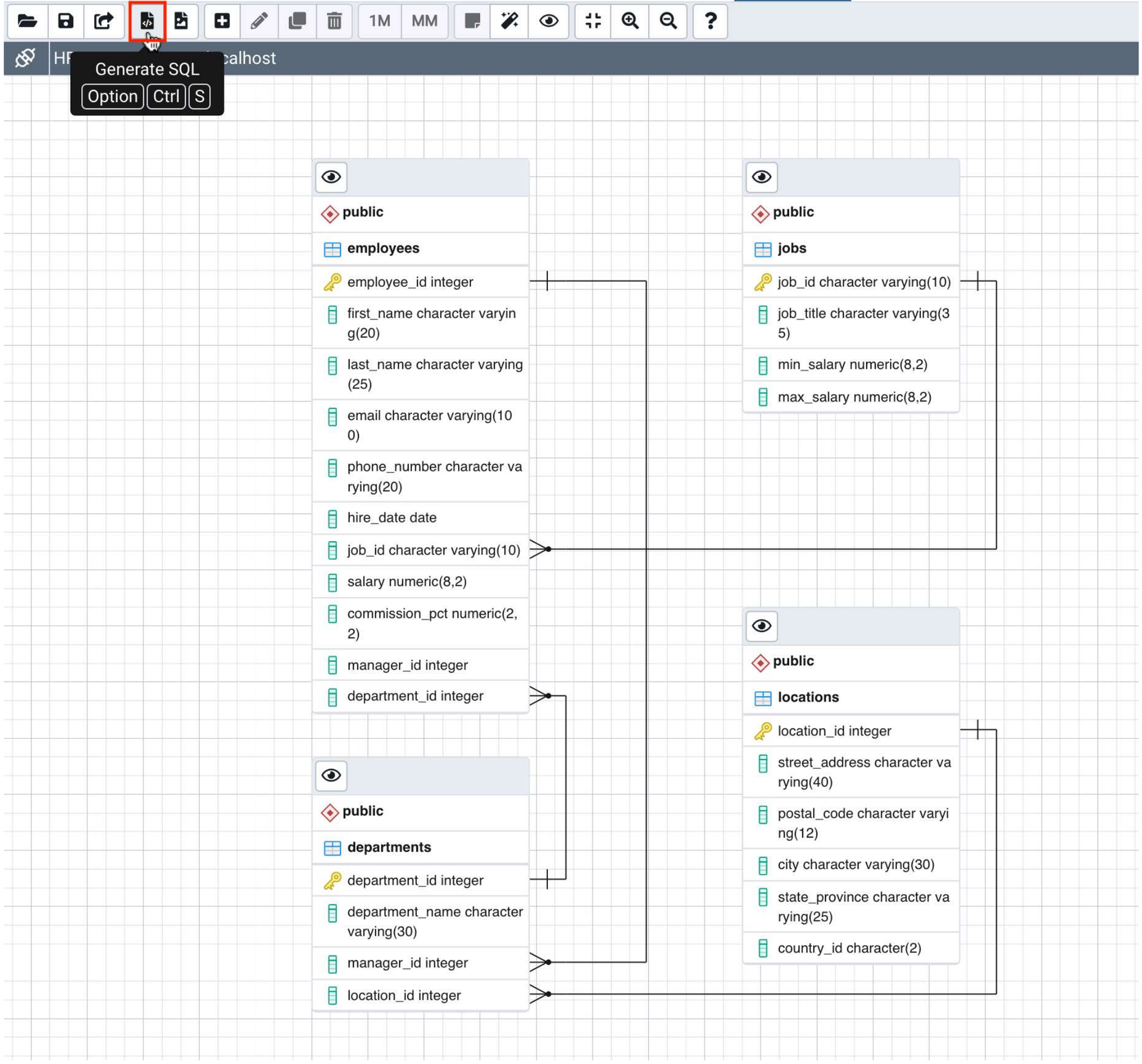
13. After creating all four relationships, you have completed the ERD for this exercise. Proceed to Task B.



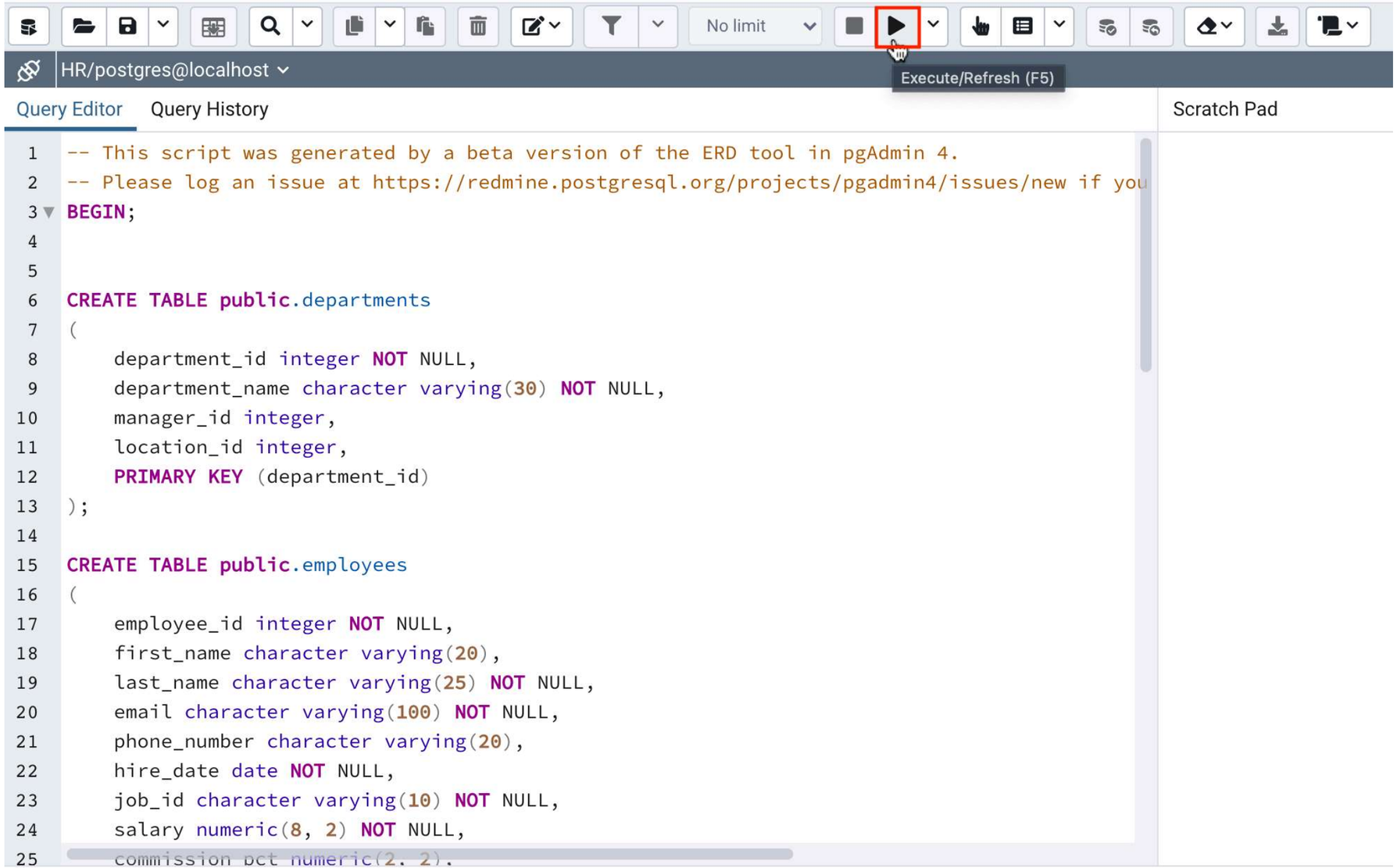
Task B: Generate and execute SQL script from ERD to create the schema

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**.



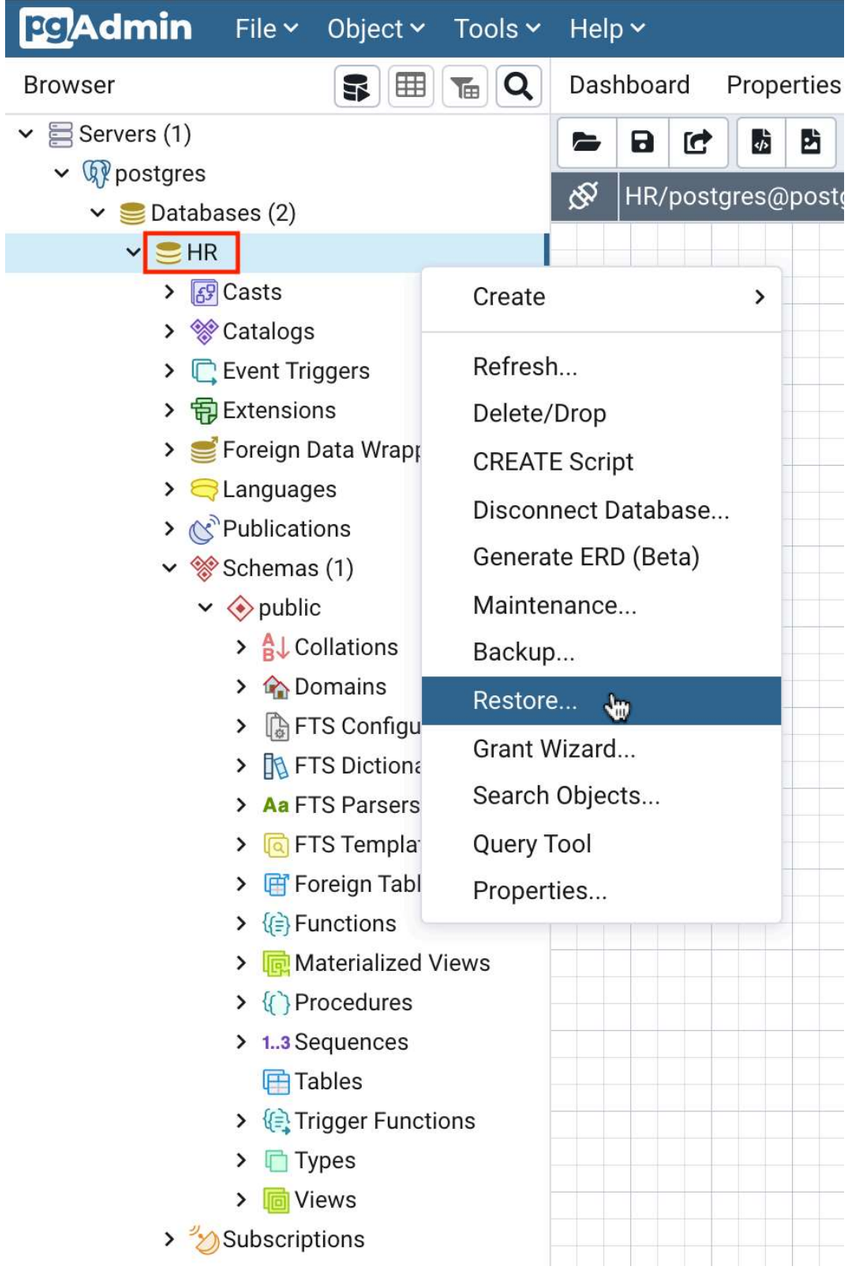
2. A new Query Editor window will open containing a SQL script generated from the ERD. Click **Execute/Refresh** to run the script. Proceed to Task C.



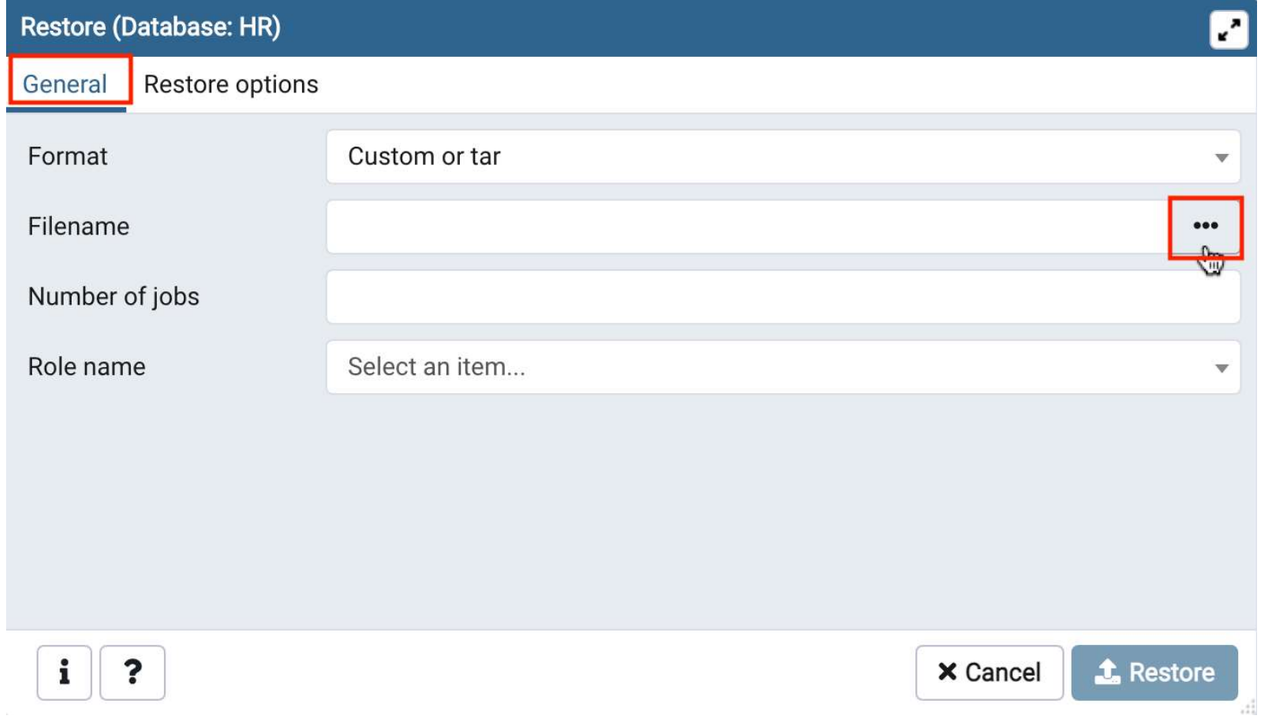
Task C: Load the database schema with data

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.

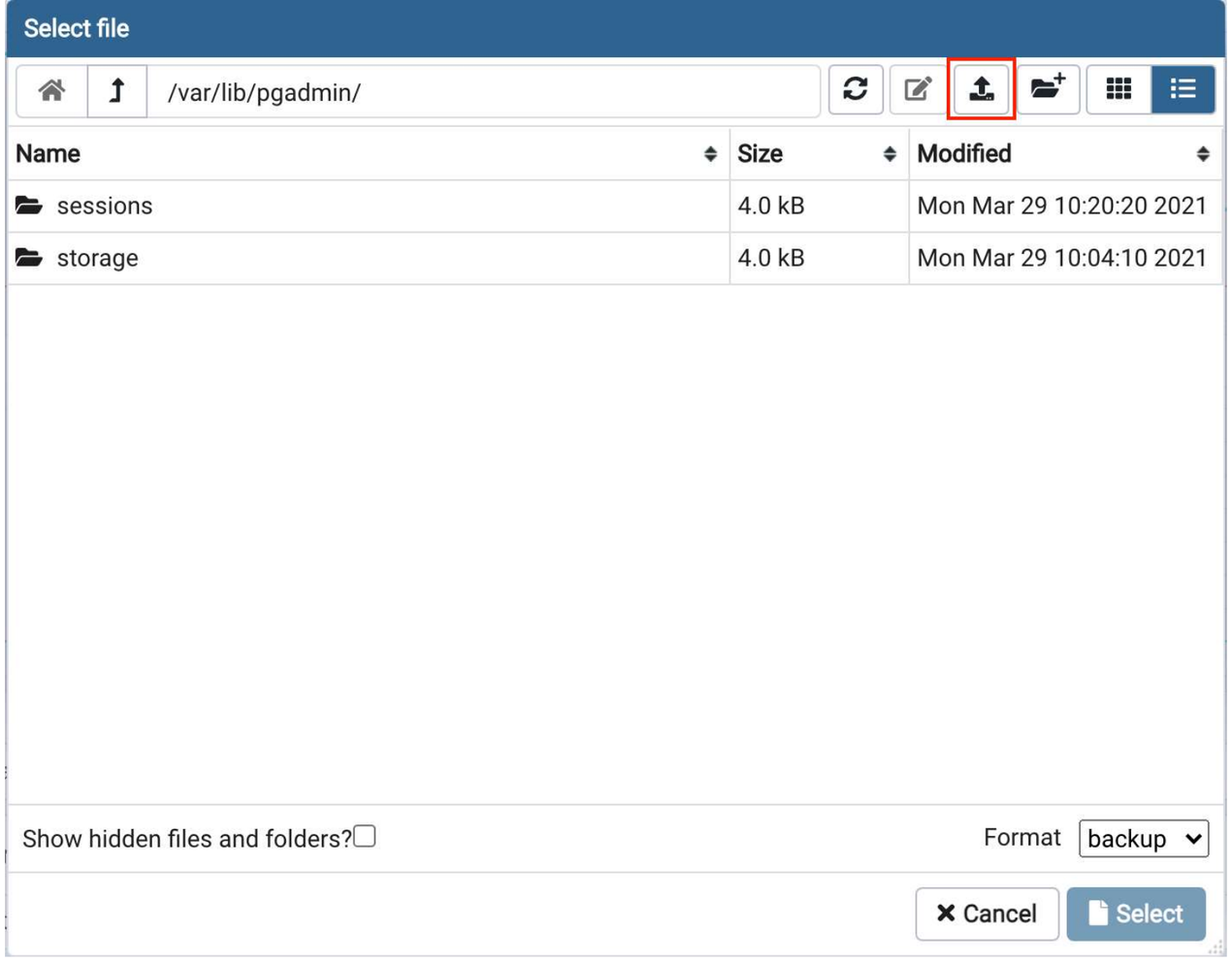
1. Download the **HR_pgsql_dump_data_for_example-exercise.tar** PostgreSQL dump file (containing the partial HR database data) using the link below to your local computer.
   - HR_pgsql_dump_data_for_example-exercise.tar
2. Follow the instructions below to import/restore the data:
   - In the tree-view,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



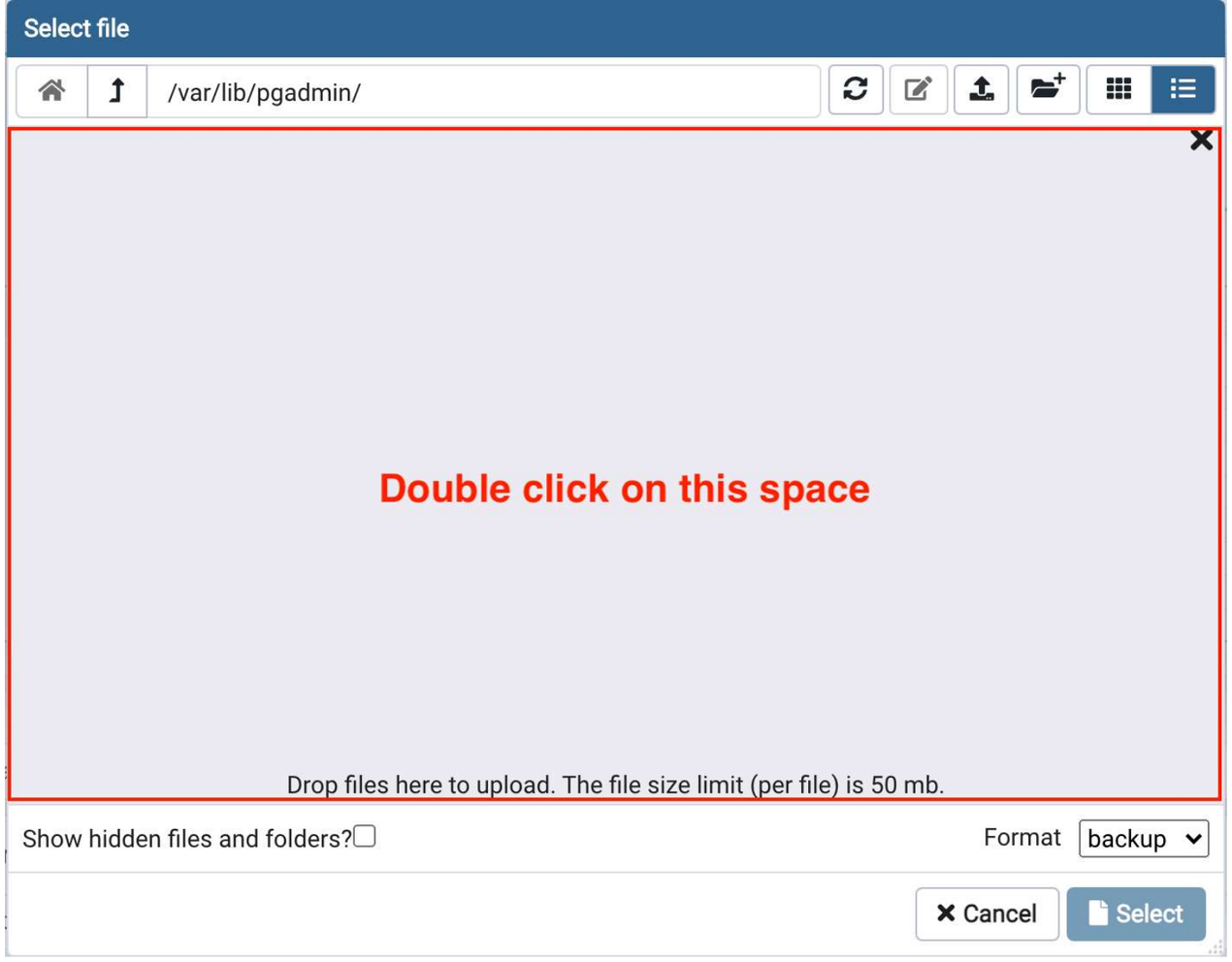
- On the **General** tab, click **Select file** by the Filename box.



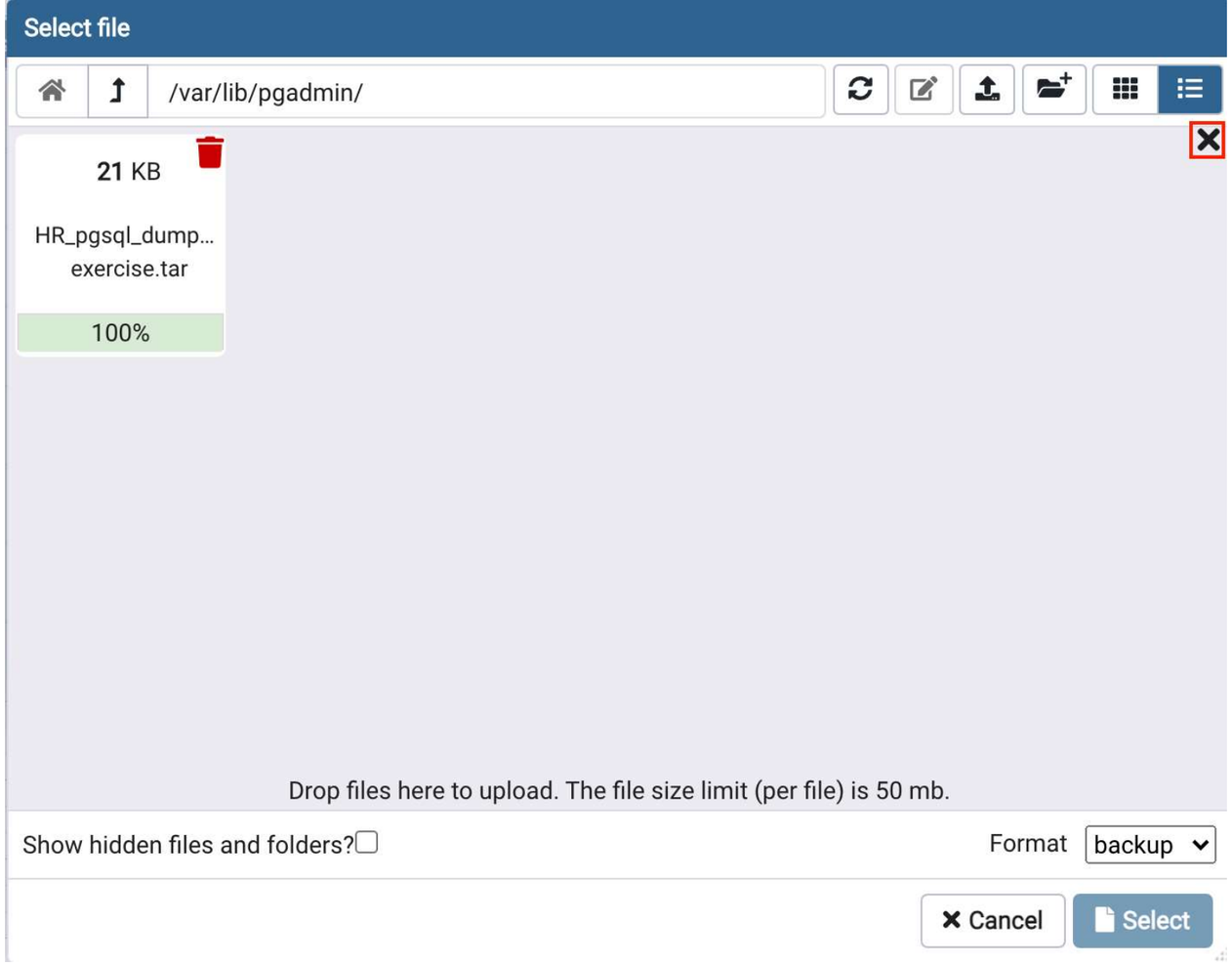
- Click **Upload File**.



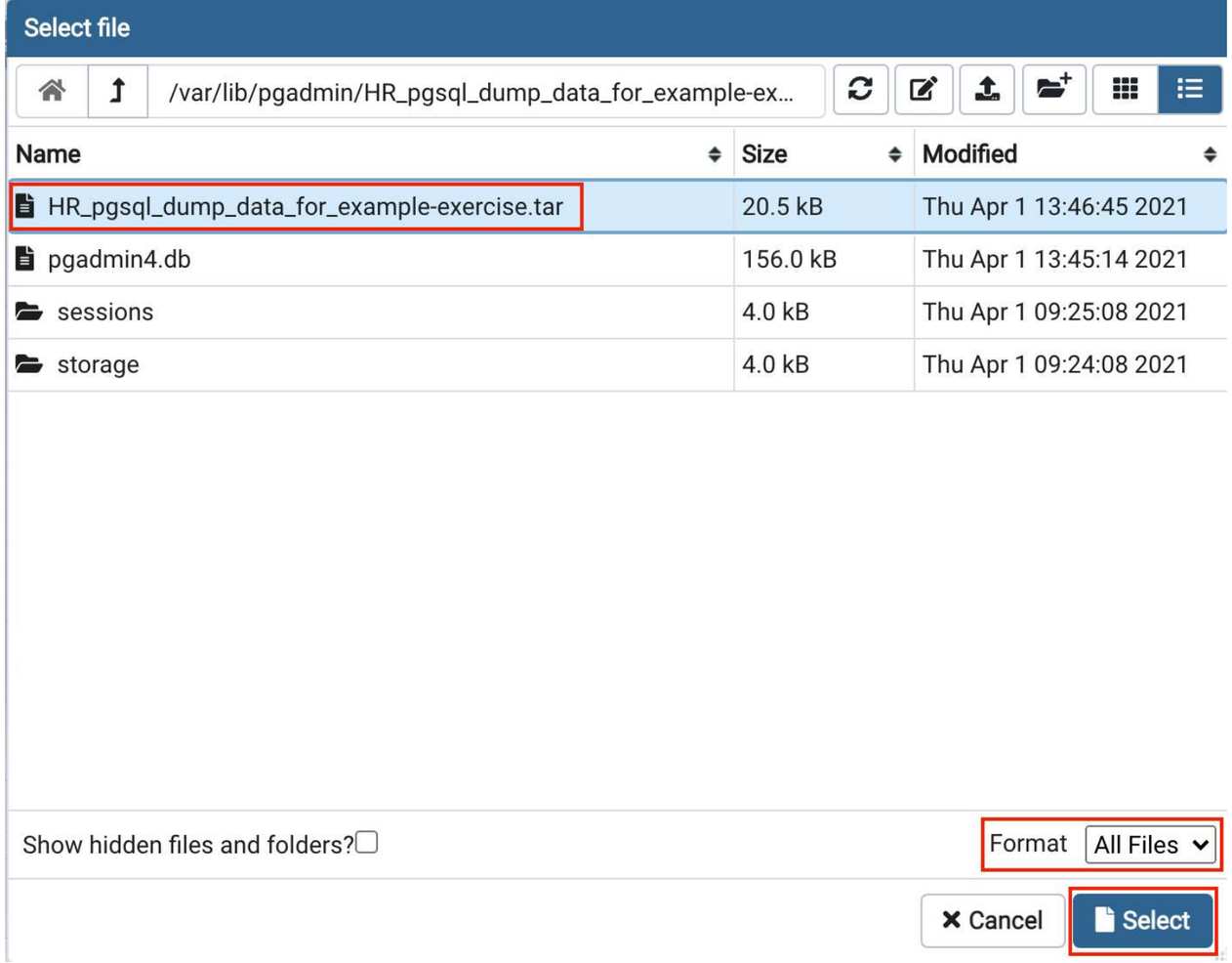
- Double-click on the drop files area and load the **HR_pgsql_dump_data_for_example-exercise.tar** you downloaded earlier on your local computer.



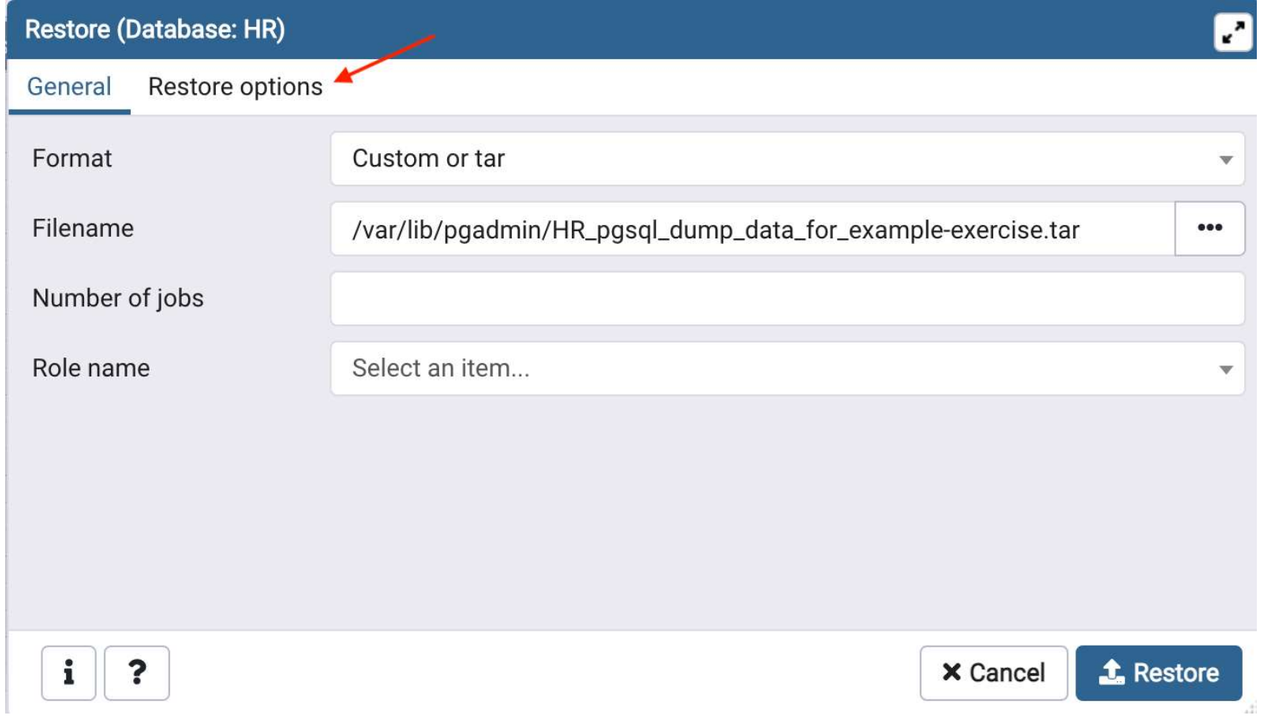
- When the upload is complete, close the drop files area by clicking **X**.



- 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**.



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

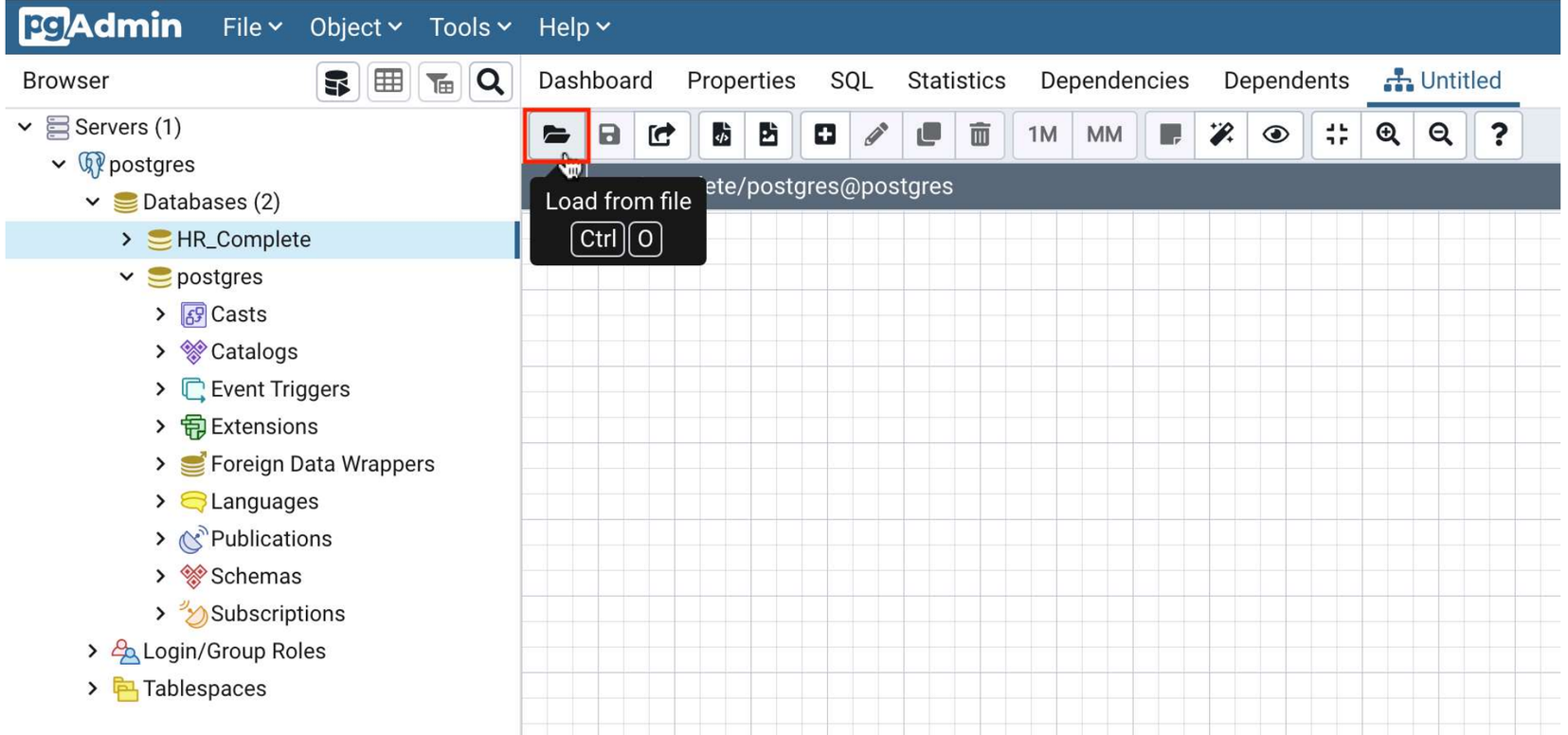


- Under **Disable**, set the **Trigger** option to **Yes**. Then click **Restore**.

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.

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.
   - 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

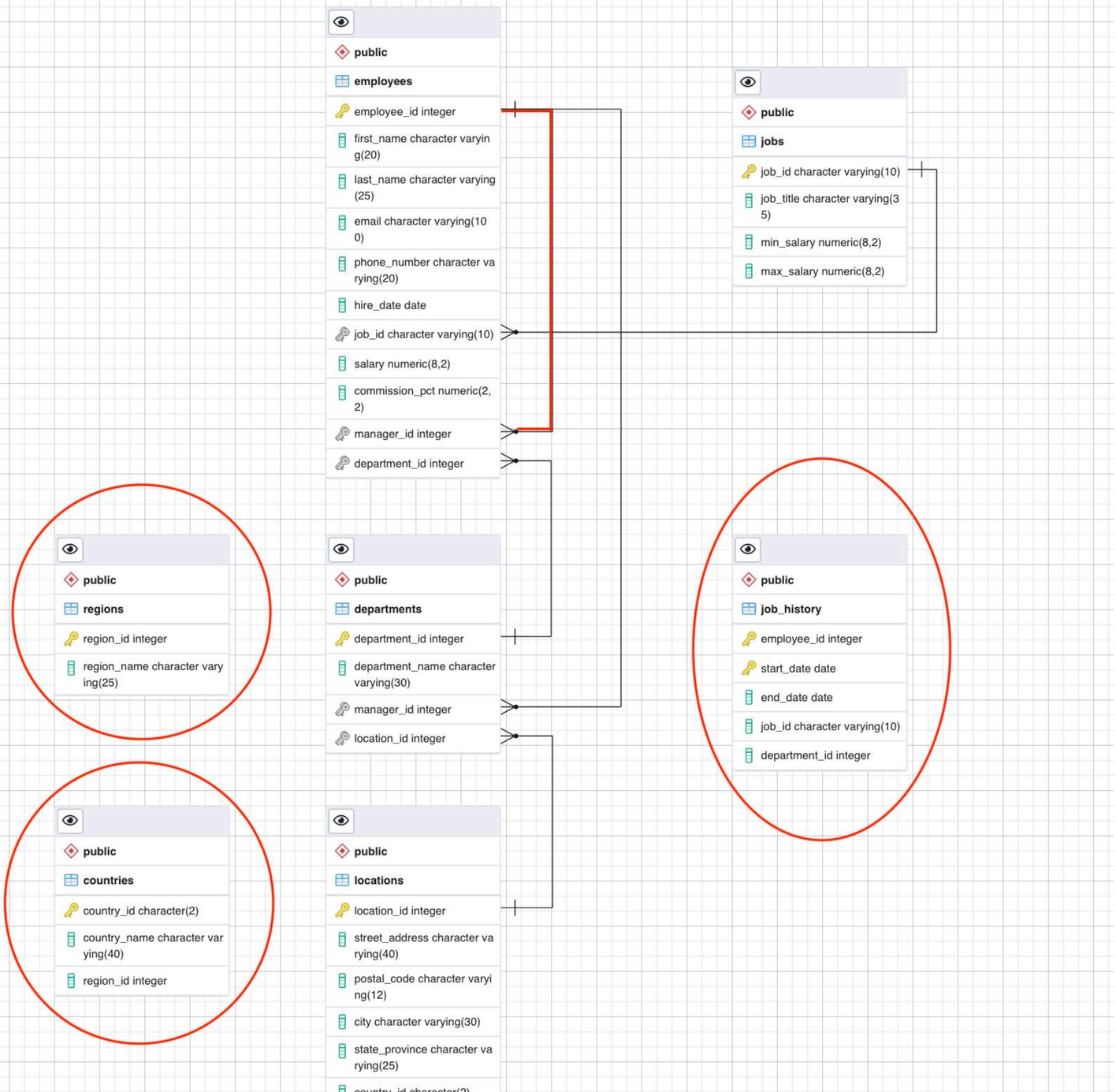
Browser

- Servers (1)
  - postgres
    - Databases (2)
      - HR_Complete
        - Casts
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
      - postgres
    - Login/Group Roles
    - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents HR_pgsql_ERD_for_practice-exercise.pgerd

1M MM

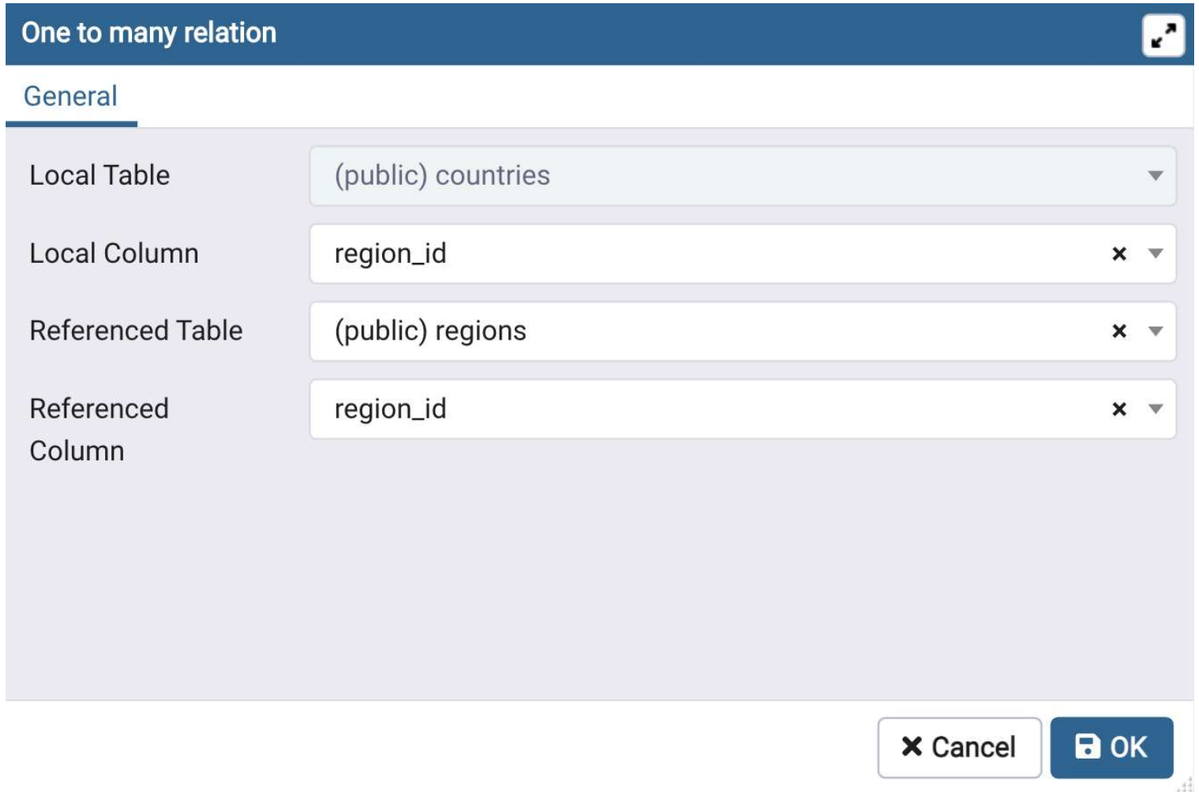
HR_Complete/postgres@postgres



5. Create the remaining relationships between the tables:

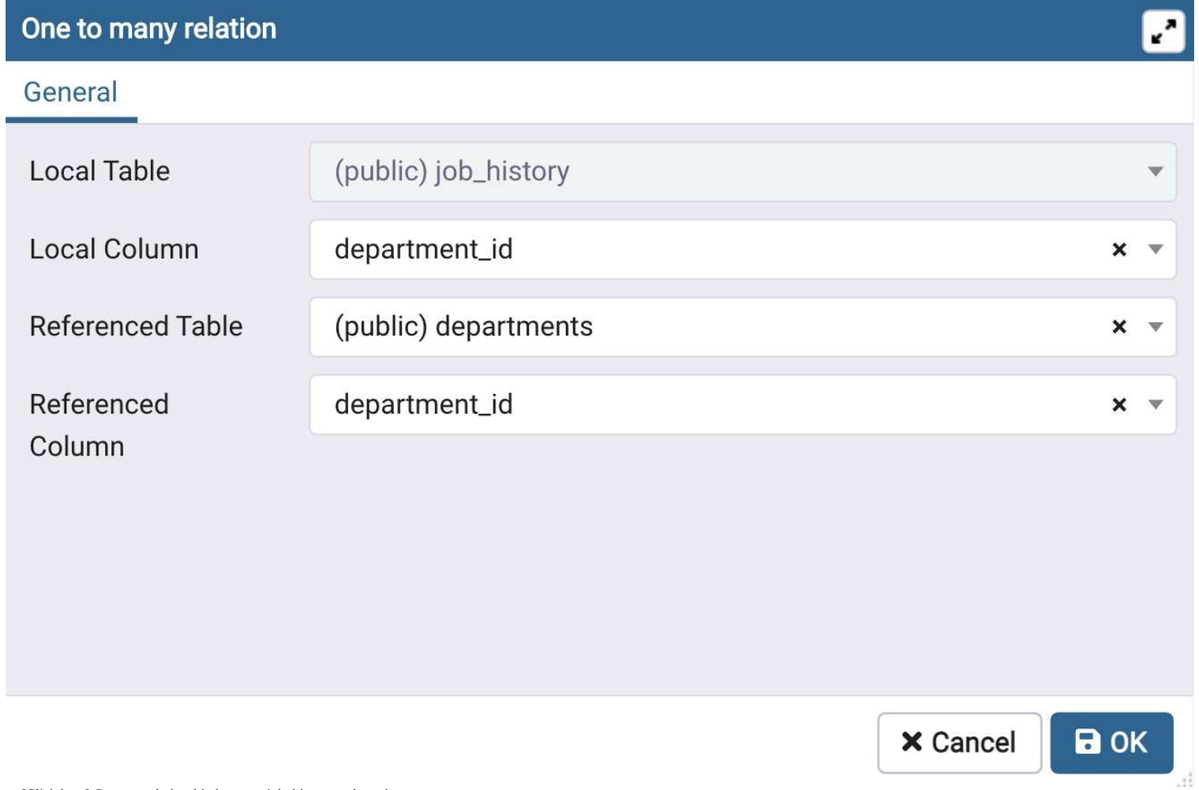
▼ [Click here] Create a relationship between countries and regions

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**.



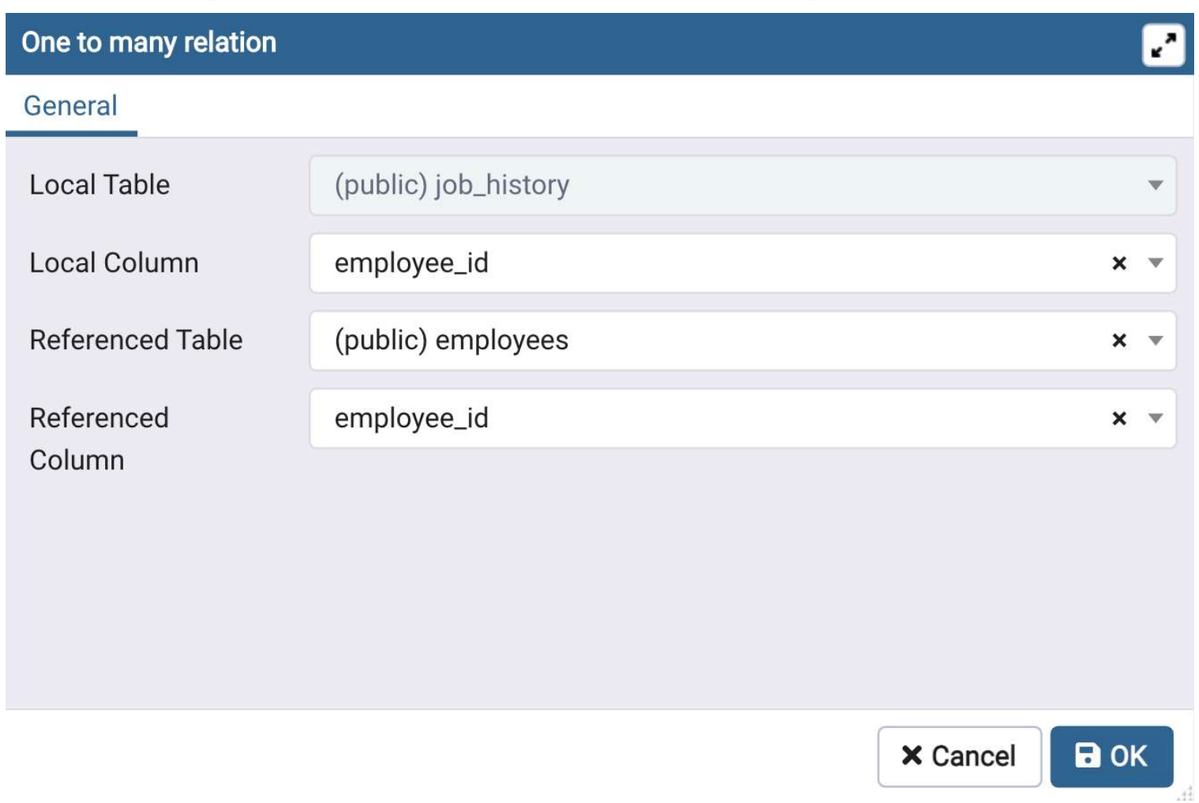
▼ [Click here] Create a relationship between job_history and departments

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**.



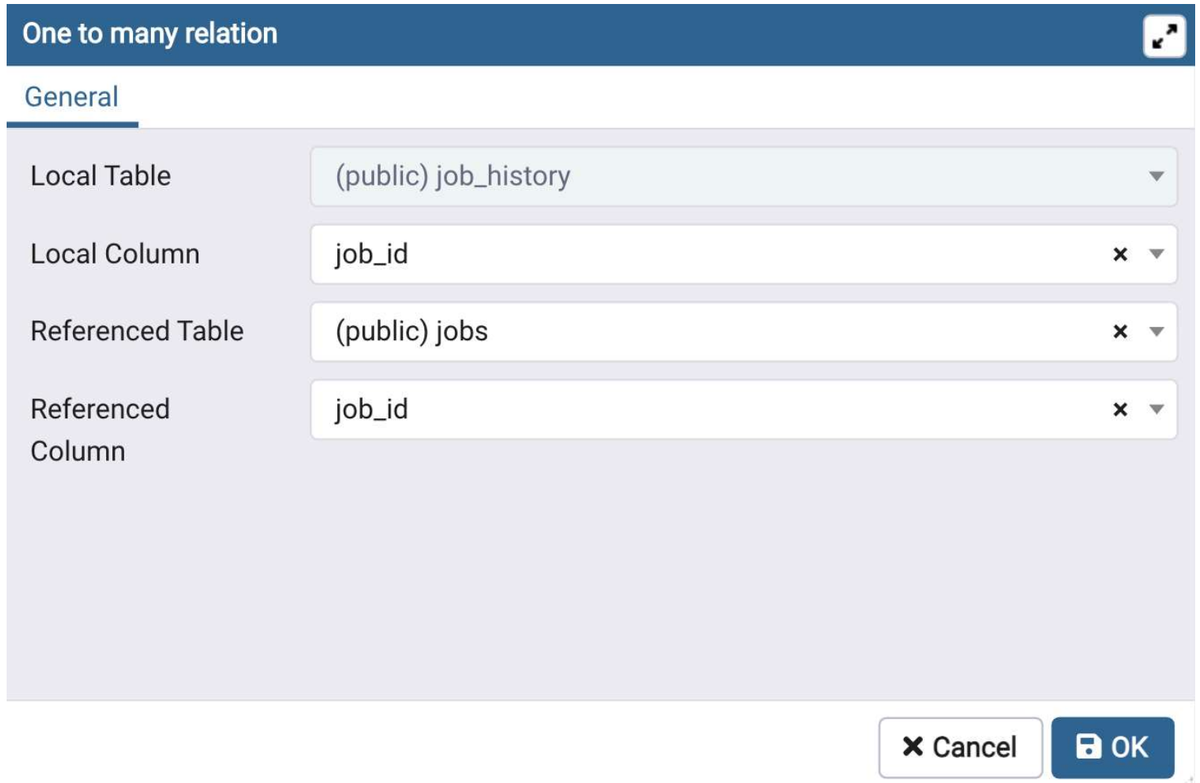
▼ [Click here] Create a relationship between job_history and employees

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**.



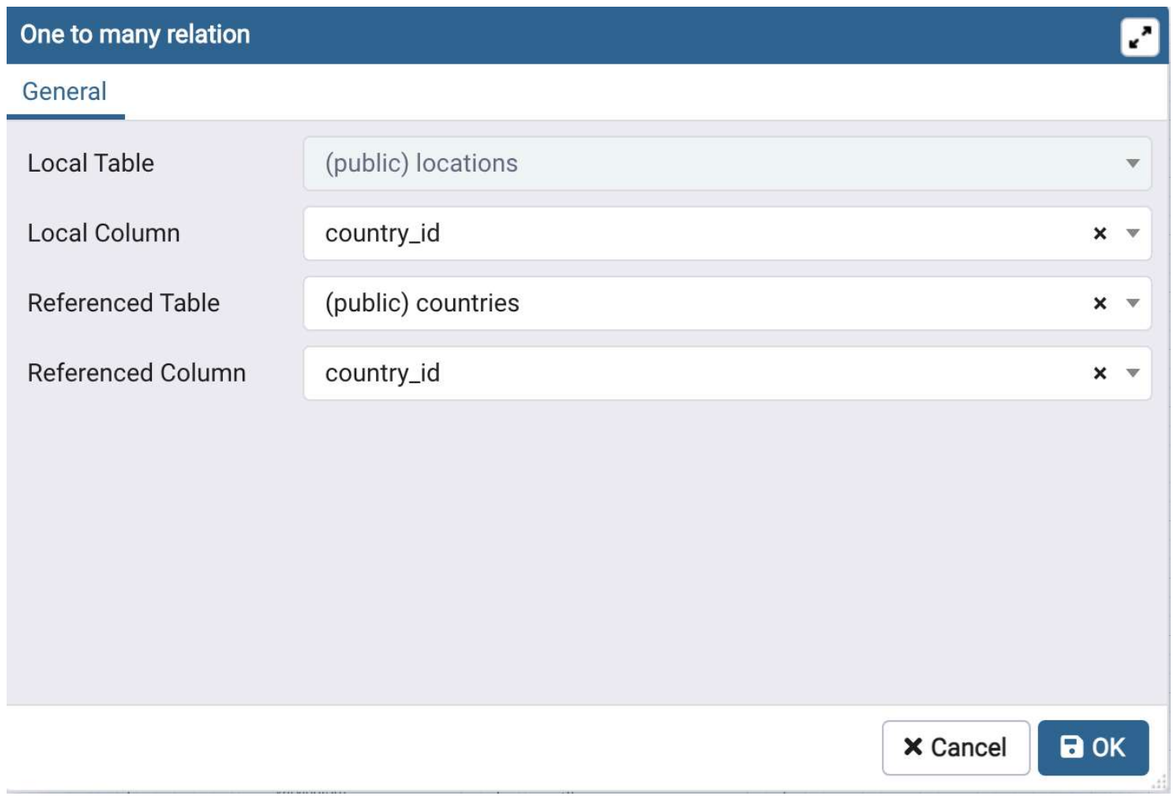
▼ [Click here] Create a relationship between job_history and jobs

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**.



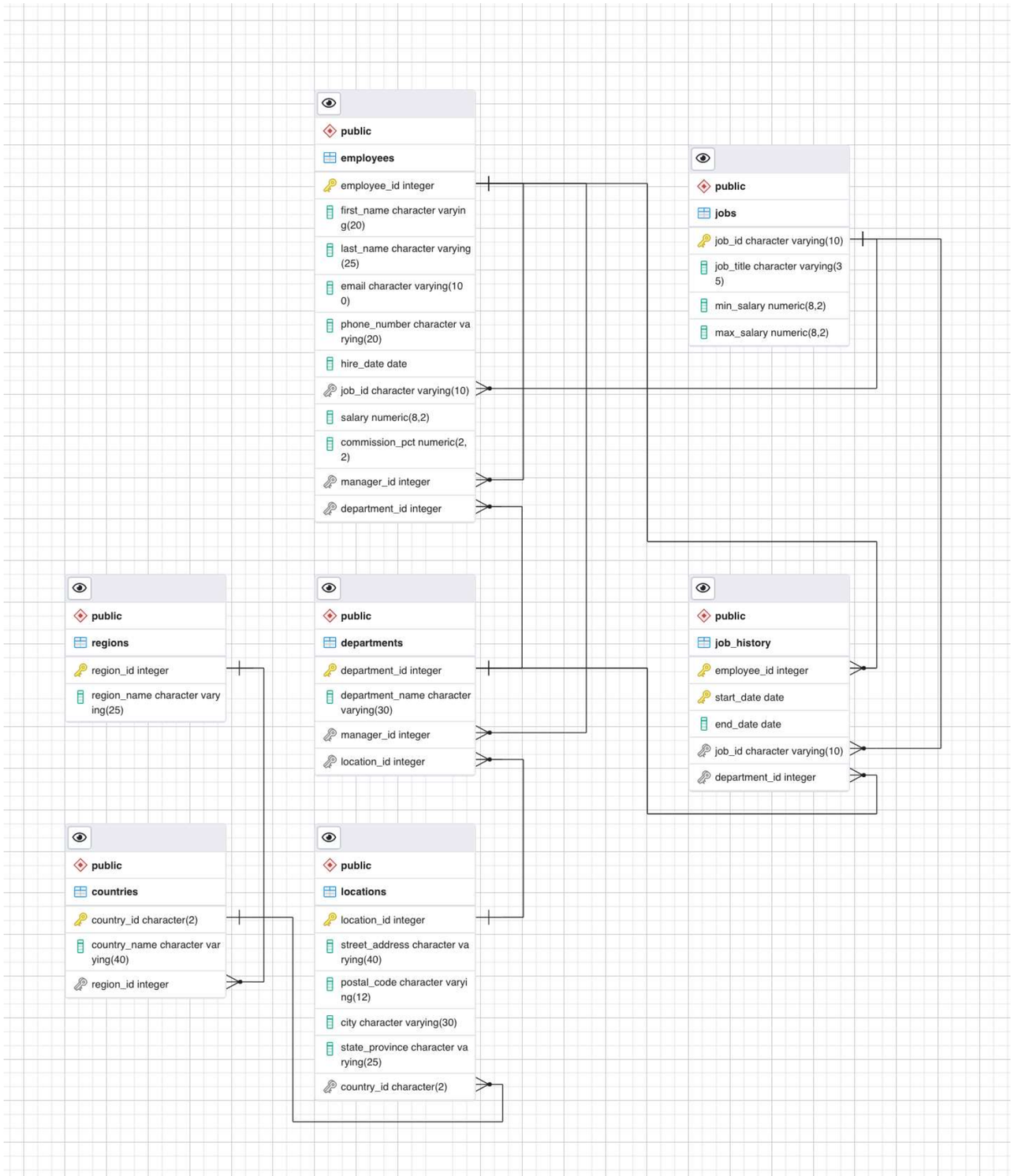
▼ [Click here] Create a relationship between locations and countries

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**.



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