

# 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.



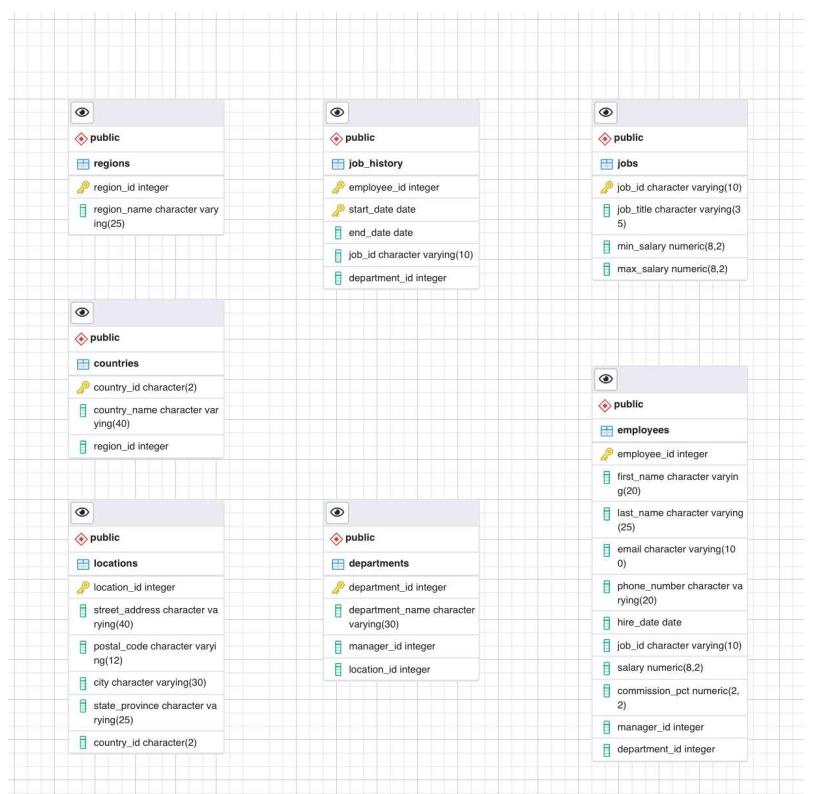
To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

## 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.

IBM Developer  
SKILLS NETWORK

## ◀ Step 6 of 7 ▶

Lab IBMCloud Launch Application

File Edit Selection View Go Run Terminal Help



New Terminal

Split Terminal

Run Task...

Run Build Task...

Run Test Task...

Run Last Task

^ ⌘ ⌘

Show Running Tasks...

Restart Running Task...

Terminate Task...

Attach Task...

Configure Tasks...

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.

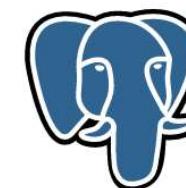
Postgres database started, waiting for all services to be ready....
[/>
Your Postgres database is now ready to use and available with username: postgres password: MTQ5NTItc2FuZGlw

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.



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

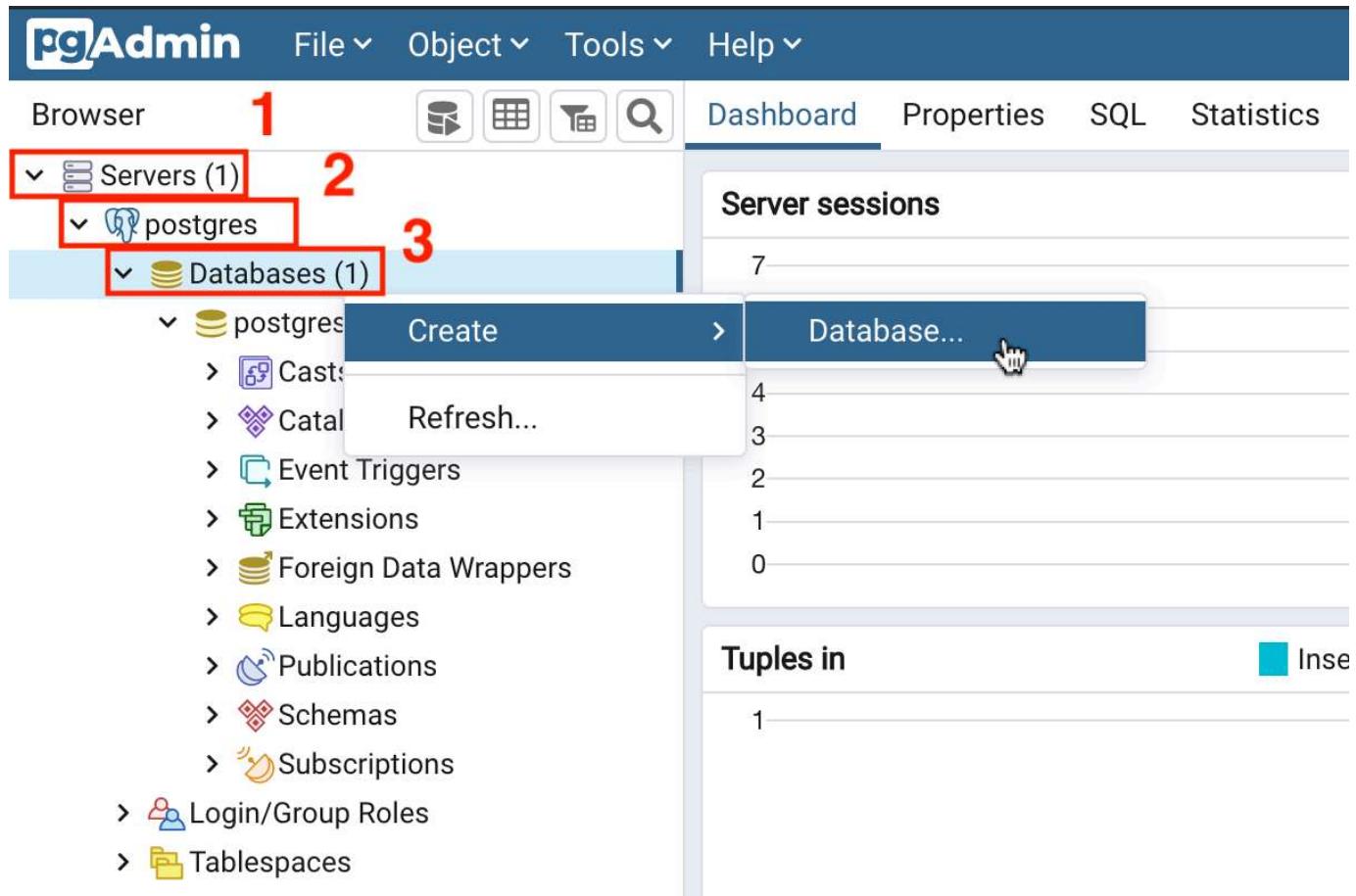


pgAdmin Website



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**.





General Definition Security Parameters Advanced SQL

Database

HR

Owner

postgres

Comment



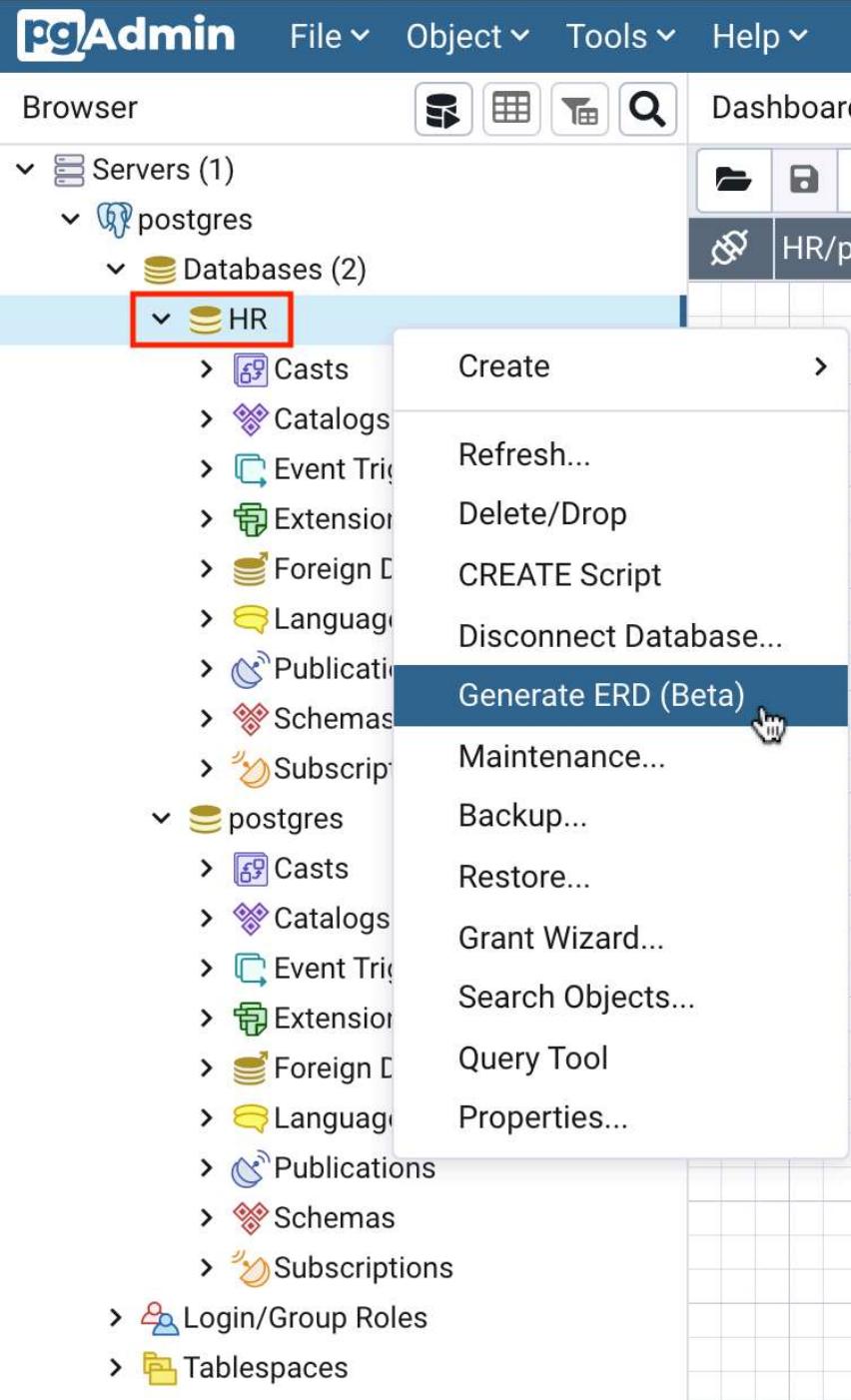
Cancel

Reset

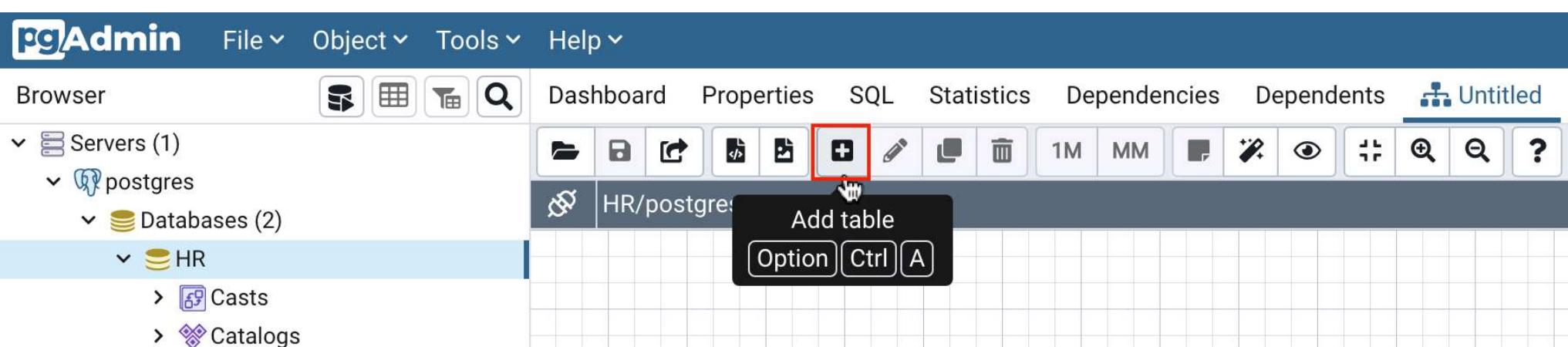
Save

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

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.



## New table



General Columns

Name

employees

Schema

◆ public

Comment

Cancel

OK

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**.

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**

General **Columns**

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary 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				

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	

**X Cancel** **OK**

Table: jobs (public)

General Columns

**Columns** **+**

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input type="checkbox"/> <input type="button" value="Delete"/>	job_id	character varying	10		<input type="checkbox"/> Yes	<input type="checkbox"/> Yes
<input type="checkbox"/> <input type="button" value="Delete"/>	job_title	character varying	35		<input type="checkbox"/> Yes	<input type="checkbox"/> No
<input type="checkbox"/> <input type="button" value="Delete"/>	min_salary	numeric	8	2	<input type="checkbox"/> No	<input type="checkbox"/> No
<input type="checkbox"/> <input type="button" value="Delete"/>	max_salary	numeric	8	2	<input type="checkbox"/> No	<input type="checkbox"/> No

**X 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**.

## Table: departments (public)



General Columns

Name	departments
Schema	public
Comment	

Cancel OK

Table: departments (public)						
General		Columns				
Columns						
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		department_id	integer		<span>Yes</span>	<span>Yes</span>
		department_name	character varying	30	<span>Yes</span>	<span>No</span>
		manager_id	integer		<span>No</span>	<span>No</span>
		location_id	integer		<span>No</span>	<span>No</span>

Cancel 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	

Table: locations (public)

General Columns

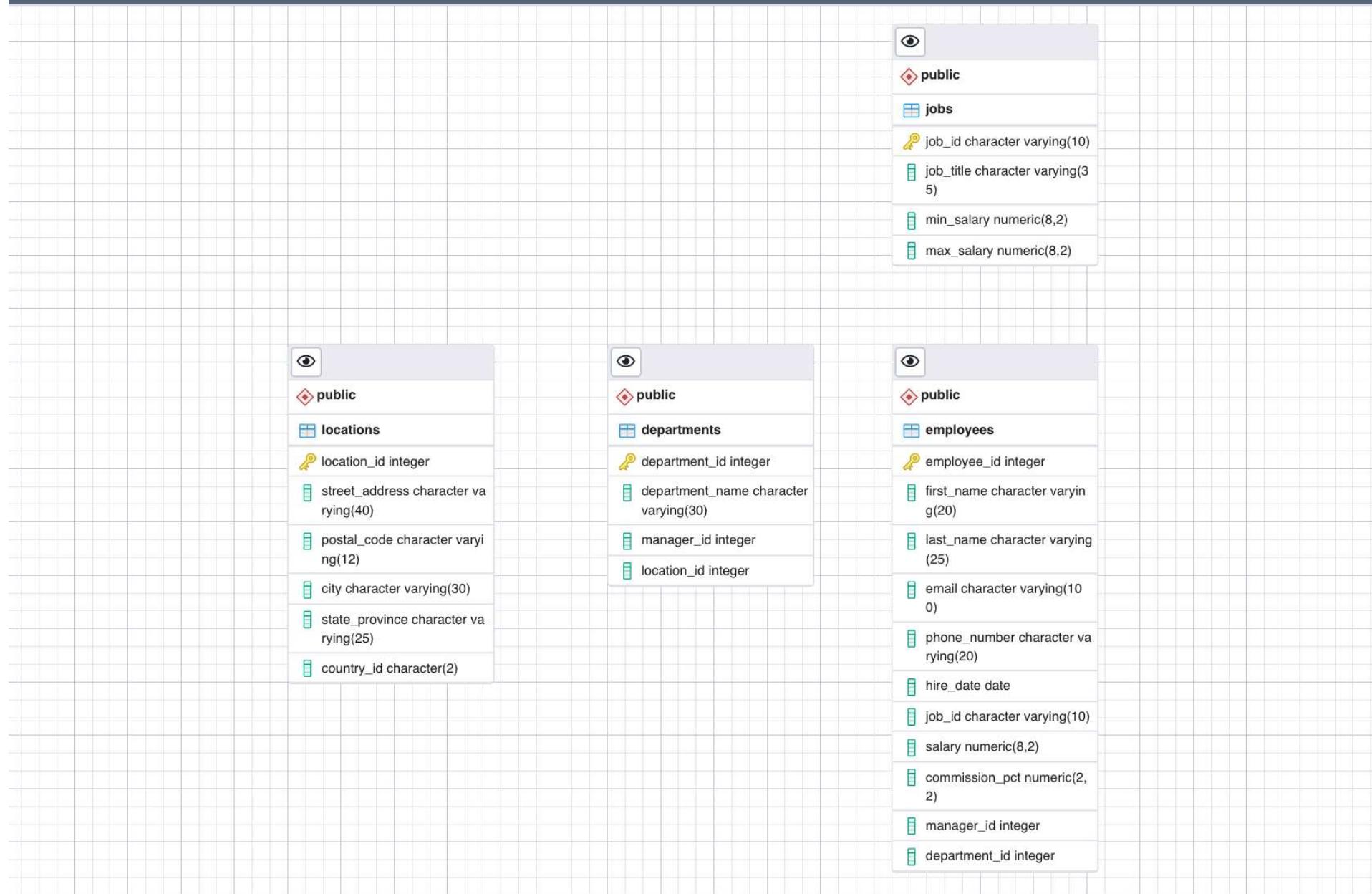
Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input type="checkbox"/> <input type="button" value="Delete"/>	location_id	integer			<input type="checkbox" value="Yes"/>	<input type="checkbox" value="Yes"/>
<input type="checkbox"/> <input type="button" value="Delete"/>	street_address	character varying	40		<input type="checkbox" value="No"/>	<input type="checkbox" value="No"/>
<input type="checkbox"/> <input type="button" value="Delete"/>	postal_code	character varying	12		<input type="checkbox" value="No"/>	<input type="checkbox" value="No"/>
<input type="checkbox"/> <input type="button" value="Delete"/>	city	character varying	30		<input type="checkbox" value="Yes"/>	<input type="checkbox" value="No"/>
<input type="checkbox"/> <input type="button" value="Delete"/>	state_province	character varying	25		<input type="checkbox" value="No"/>	<input type="checkbox" value="No"/>
<input type="checkbox"/> <input type="button" value="Delete"/>	country_id	character	2		<input type="checkbox" value="Yes"/>	<input type="checkbox" value="No"/>

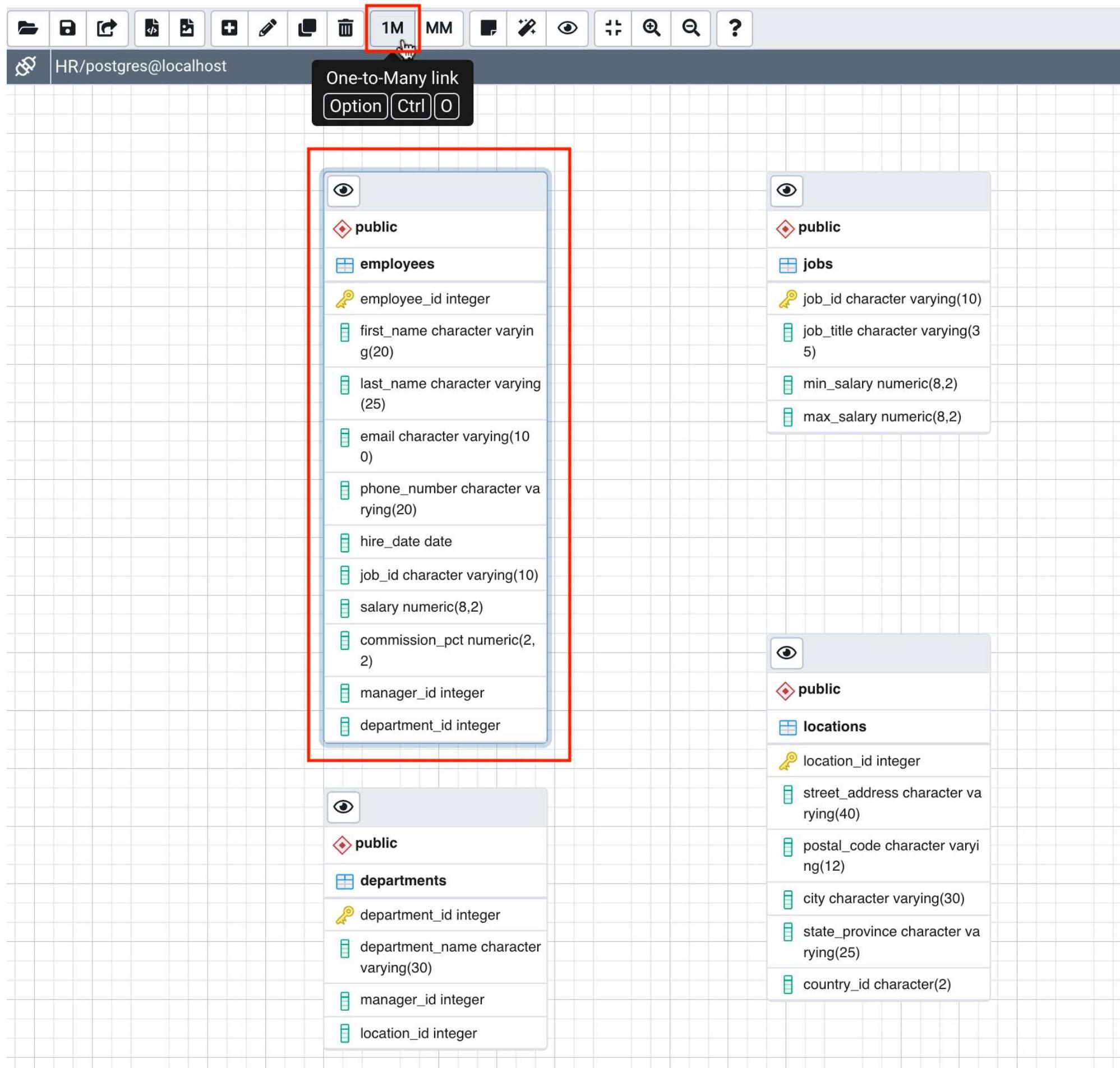
10. After creating all four entity diagrams, the entities of the ERD are complete.



/postgres@localhost



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**.



## 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 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**.

## 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**.

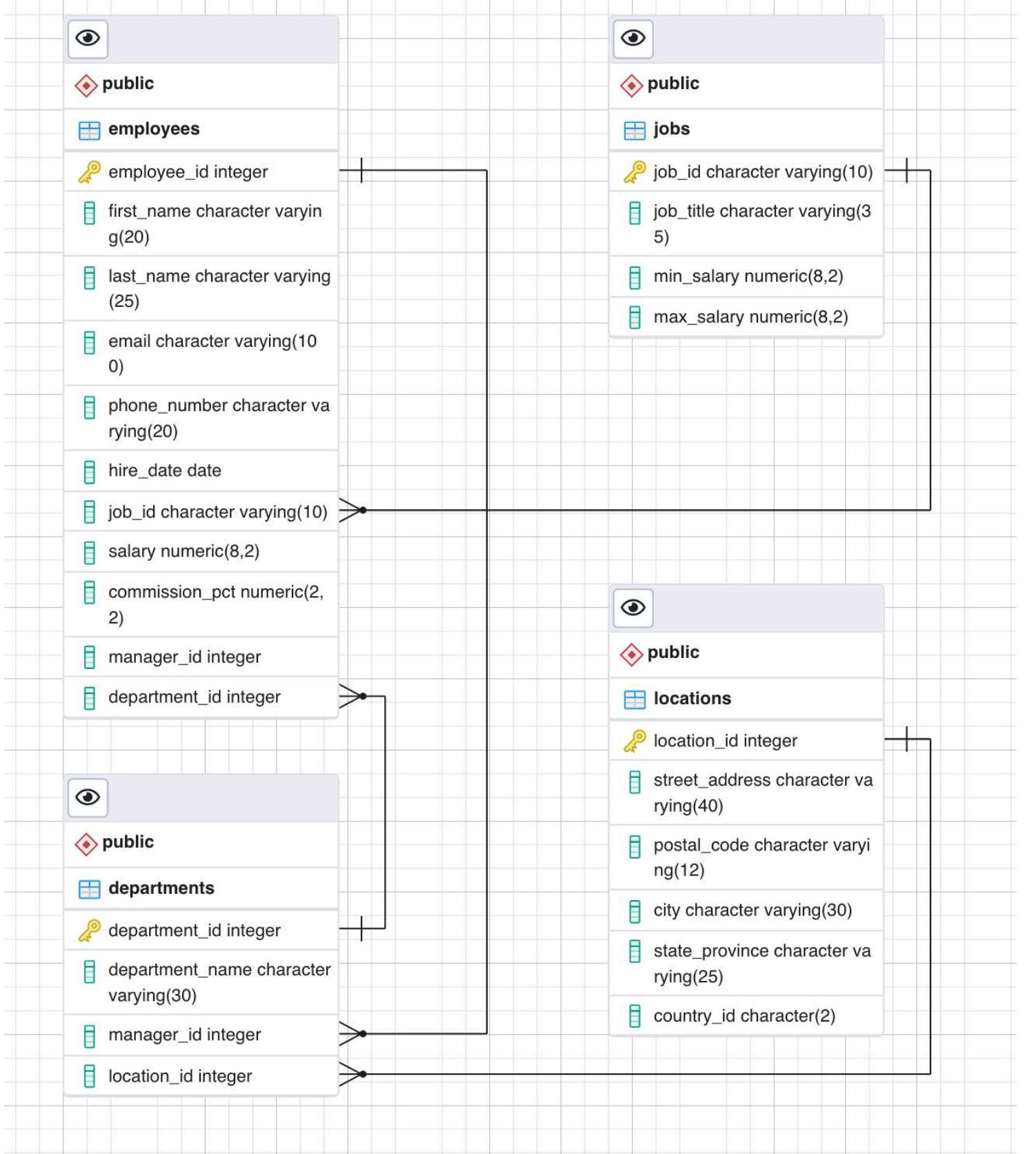
## One to many relation

### General

Local Table	(public) departments
Local Column	manager_id
Referenced Table	(public) employees
Referenced Column	employee_id

**Cancel** **OK**

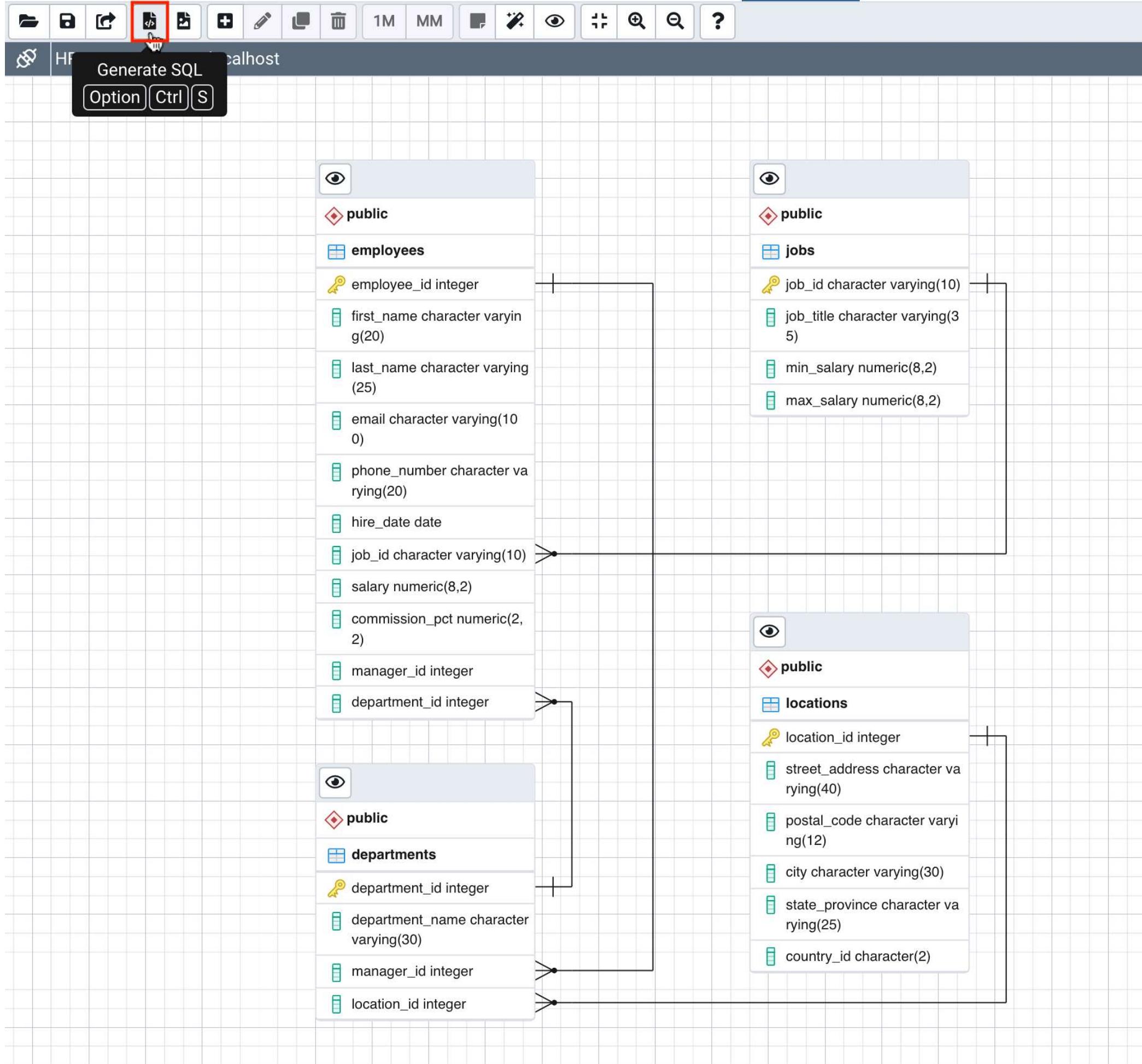
13. After creating all four relationships, you have completed the ERD for this exercise. Proceed to Task B.



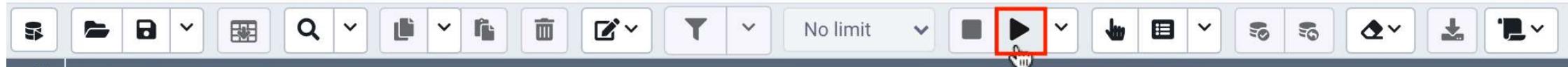
## Task B: Generate and execute SQL script from ERD to create the schema

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.



HR/postgres@localhost

Execute/Refresh (F5)

Query Editor Query History

Scratch Pad

```
1 -- This script was generated by a beta version of the ERD tool in pgAdmin 4.
2 -- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you
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**.

1. Click the **Restore** button in the toolbar.

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

3. Click the **Open** button.

4. Click the **OK** button.

5. Click the **OK** button again to start the restore process.

pgAdmin File Object Tools Help

Browser

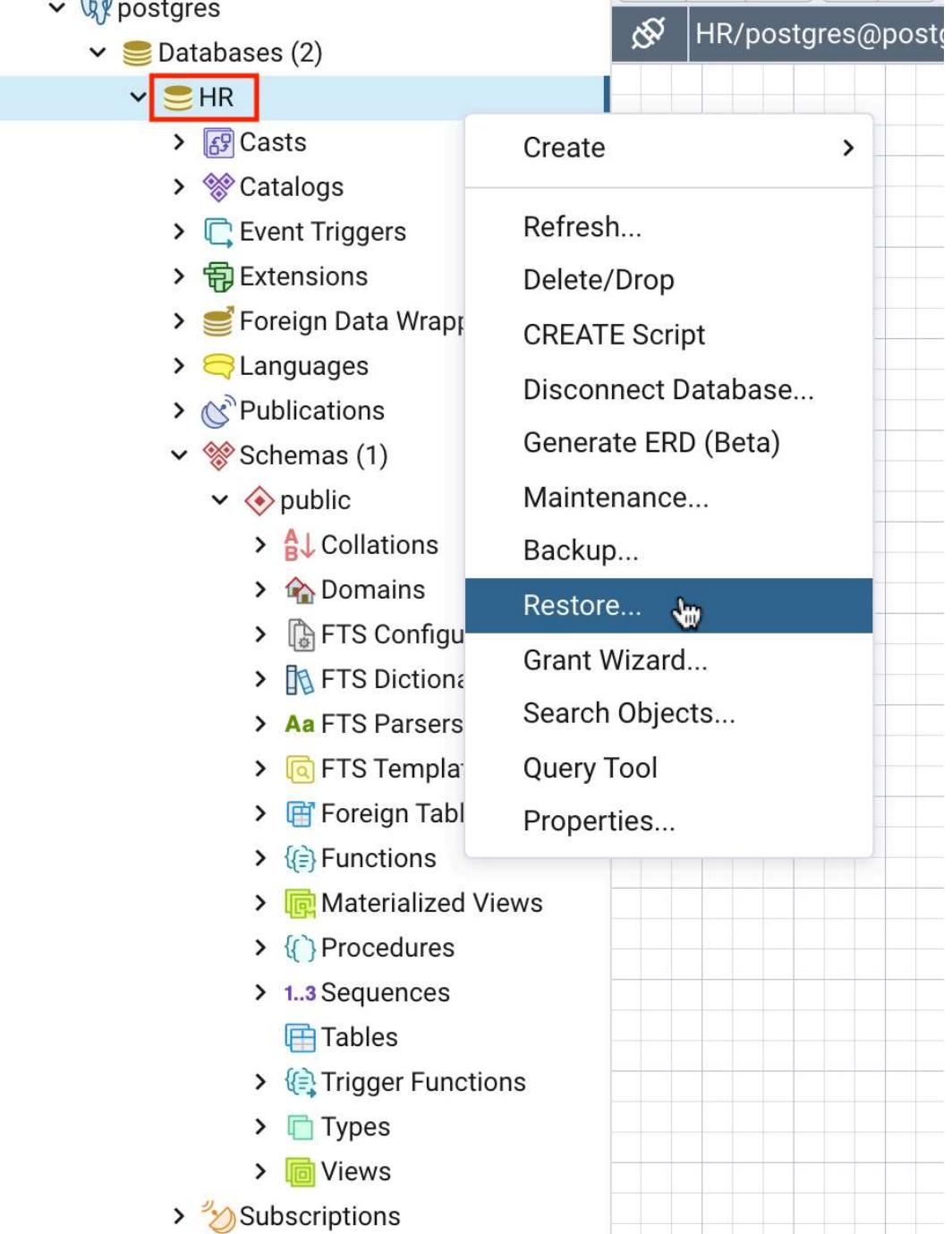
Servers (1) postres Databases (2) HR

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

Dashboard Properties

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.



General Restore options

Format

Custom or tar

Filename



Number of jobs

Role name

Select an item...



Cancel Restore

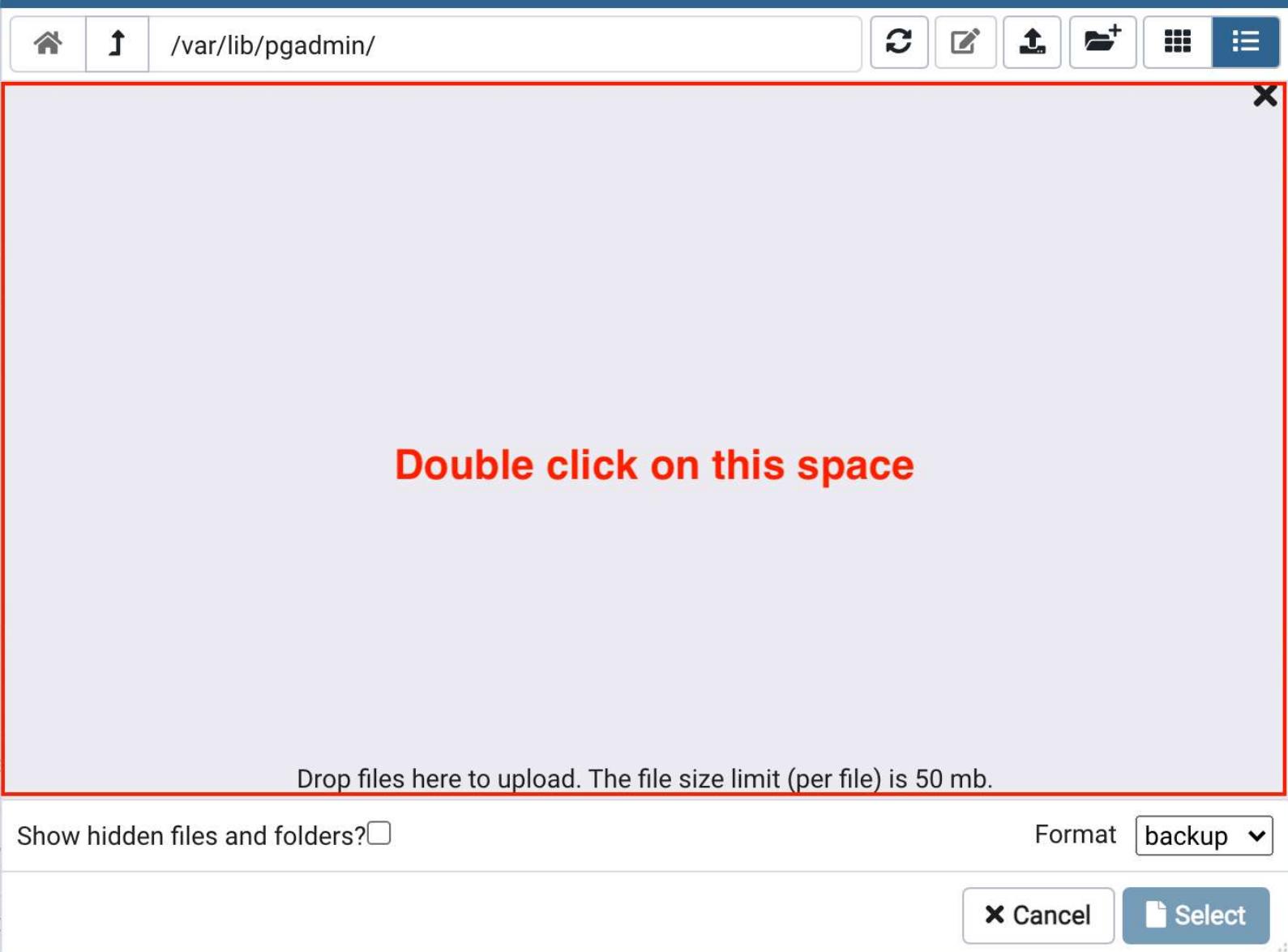
- Click **Upload File**.

## Select file

The screenshot shows a file manager interface with the following details:

- Toolbar:** Includes icons for Home, Up, Refresh, Create, Upload (highlighted with a red box), Delete, Grid, and List.
- Path:** /var/lib/pgadmin/
- File List:** A table with columns: Name, Size, and Modified. It contains two entries:
  - sessions**: 4.0 kB, Mon Mar 29 10:20:20 2021
  - storage**: 4.0 kB, Mon Mar 29 10:04:10 2021
- Bottom Left:** A checkbox labeled "Show hidden files and folders?".
- Bottom Center:** A "Format" dropdown set to "backup".
- Bottom Right:** Buttons for "Cancel" and "Select".

- 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.

21 KB

HR\_pgsql\_dump...  
exercise.tar

100%

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

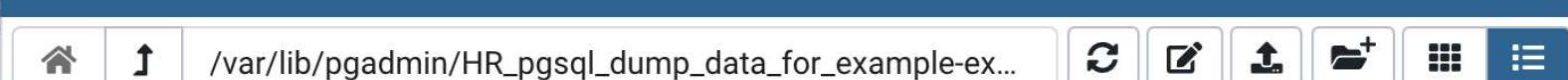
Show hidden files and folders?

Format backup

Cancel Select

- 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



Name	Size	Modified
HR_pgsql_dump_data_for_example-exercise.tar	20.5 kB	Thu Apr 1 13:46:45 2021
pgadmin4.db	156.0 kB	Thu Apr 1 13:45:14 2021
sessions	4.0 kB	Thu Apr 1 09:25:08 2021
storage	4.0 kB	Thu Apr 1 09:24:08 2021

Show hidden files and folders?

Format All Files

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...

**Cancel**  **Restore**

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

General **Restore options**

**Queries**

Include CREATE DATABASE statement:  No

Clean before restore:  No

Single transaction:  No

**Disable**

Trigger:  Yes

No data for Failed Tables:  No

**Cancel**  **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.

pgAdmin File Object Tools Help

Browser

Servers (1) **postgres**

Databases (2) **HR\_Complete**

postgres

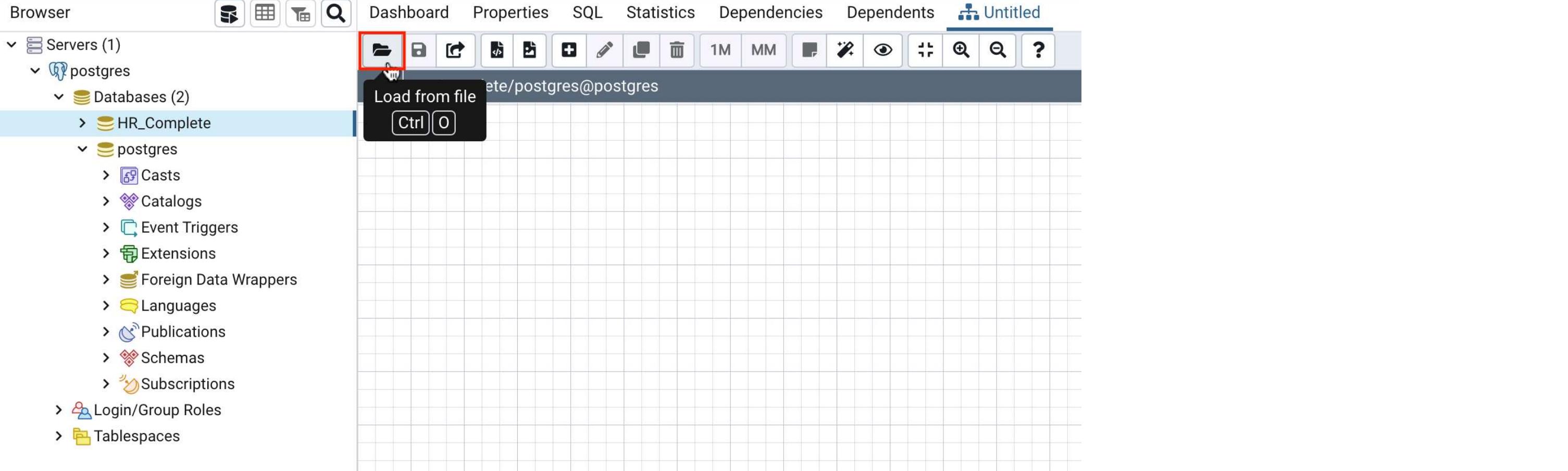
- Casts
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas
- Subscriptions

Login/Group Roles

Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Untitled

Load from file Ctrl O



**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.

## Servers (1)

postgres

## Databases (2)

HR\_Complete

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

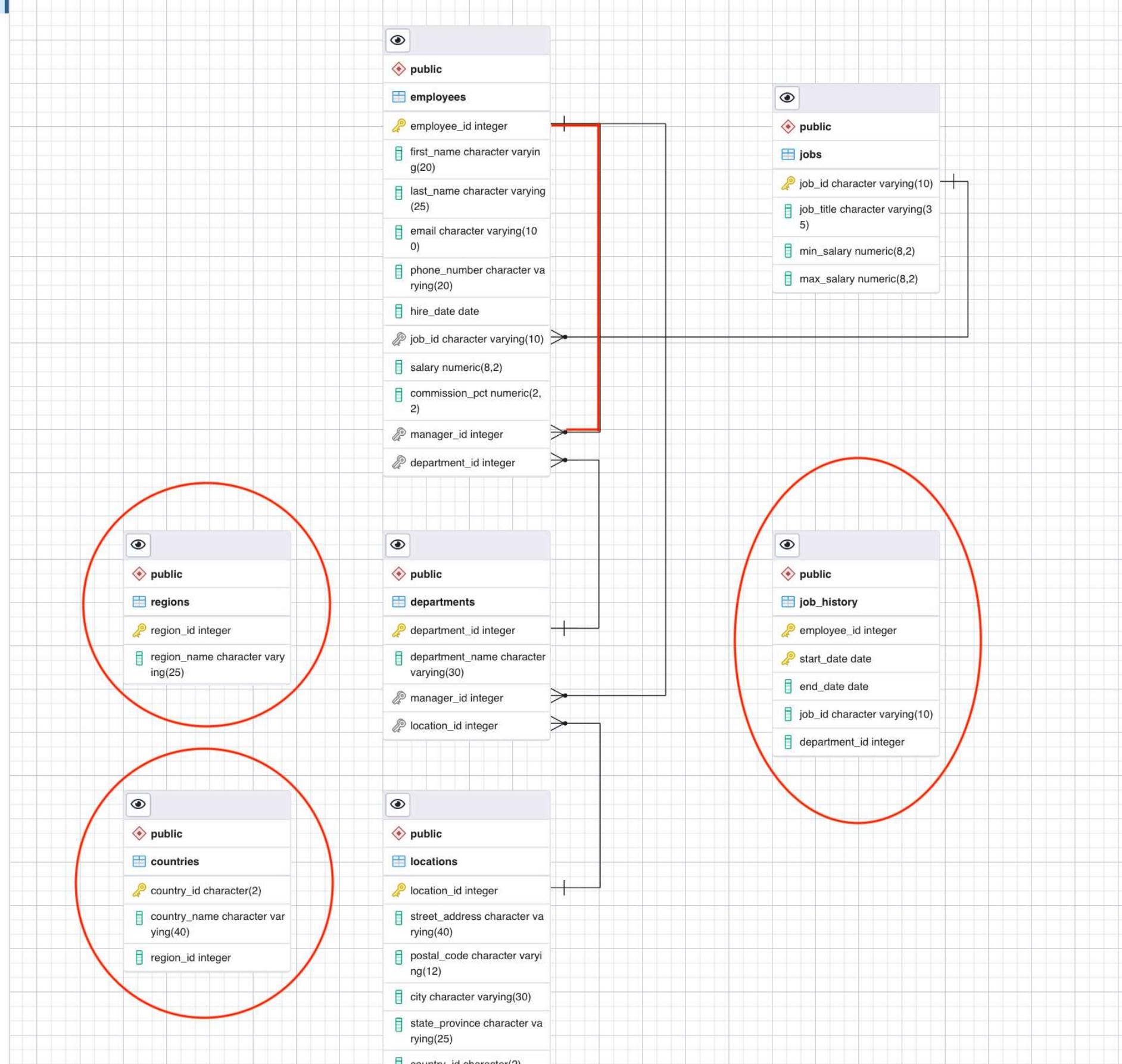
Schemas

Subscriptions

postgres

Login/Group Roles

Tablespaces



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

**Cancel** **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**.

## One to many relation

### General

Local Table (public) job\_history

Local Column department\_id

Referenced Table (public) departments

Referenced Column department\_id

Cancel  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**.

## One to many relation

### General

Local Table (public) job\_history

Local Column employee\_id

Referenced Table (public) employees

Referenced Column employee\_id

Cancel  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**.

## One to many relation

### General

Local Table	(public) job_history
Local Column	job_id
Referenced Table	(public) jobs
Referenced Column	job_id

**Cancel** **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**.

## One to many relation

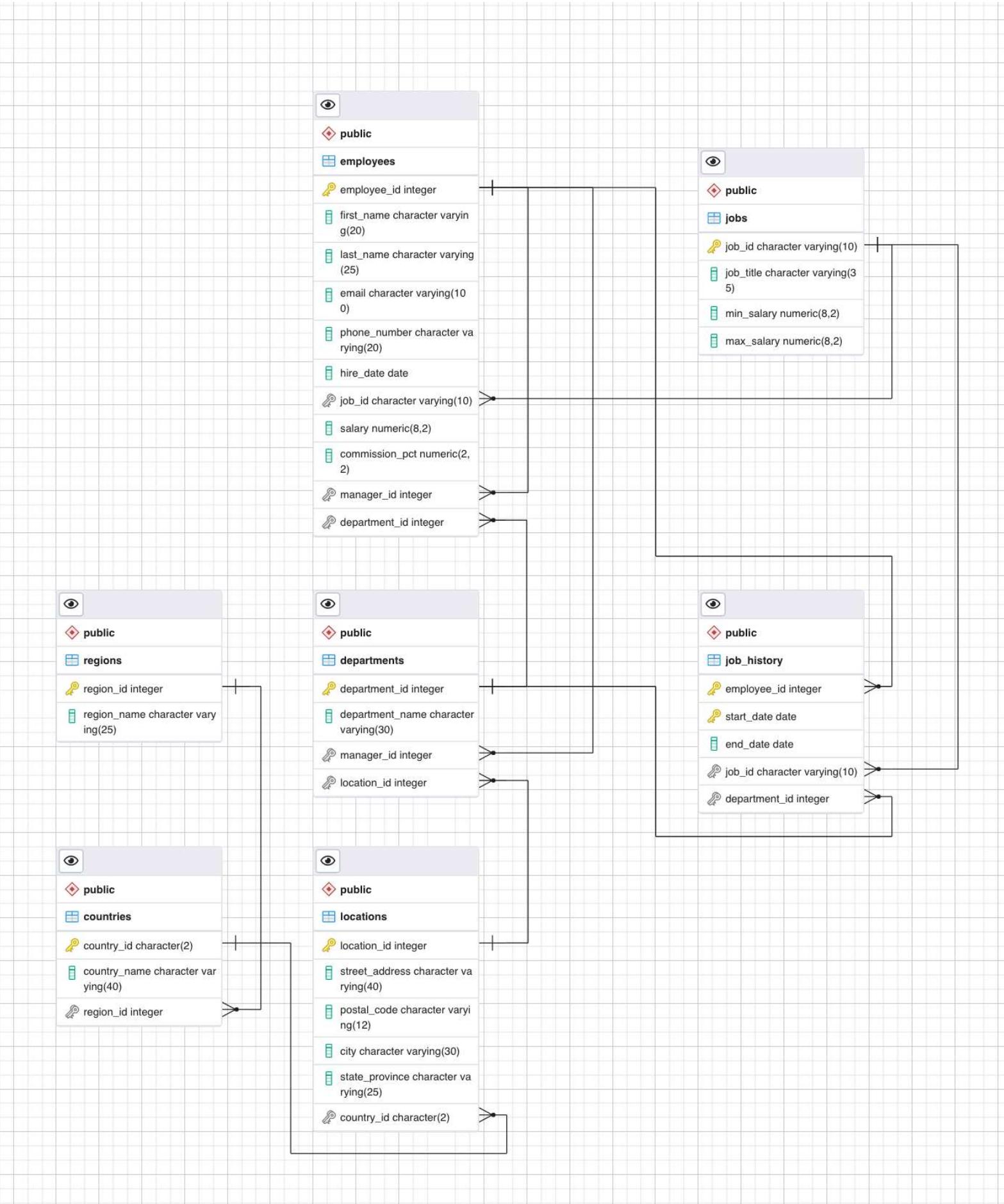
### General

Local Table	(public) locations
Local Column	country_id
Referenced Table	(public) countries
Referenced Column	country_id

**Cancel** **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](#)

## 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.