



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.



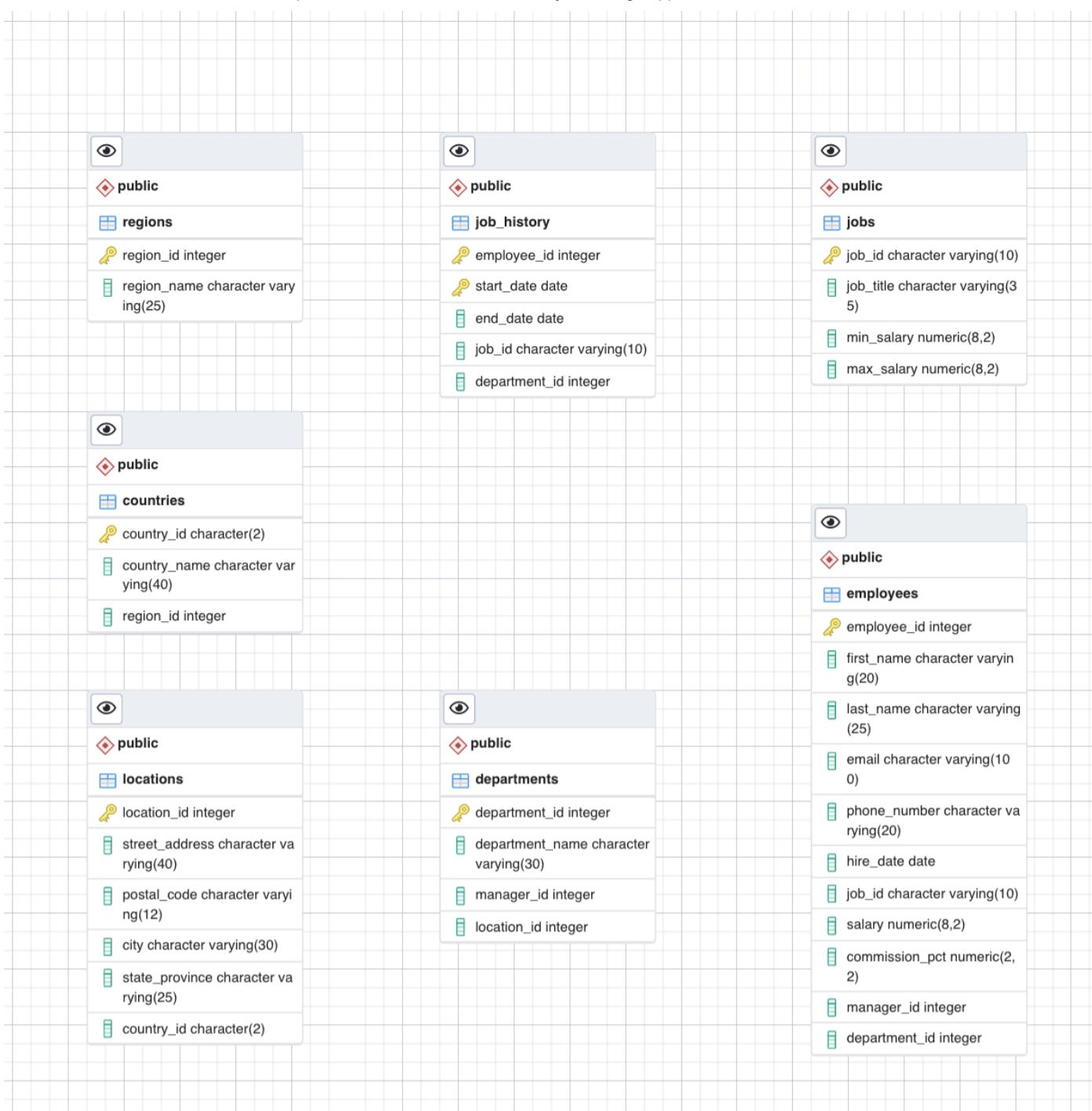
To complete this lab you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

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, so 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*.

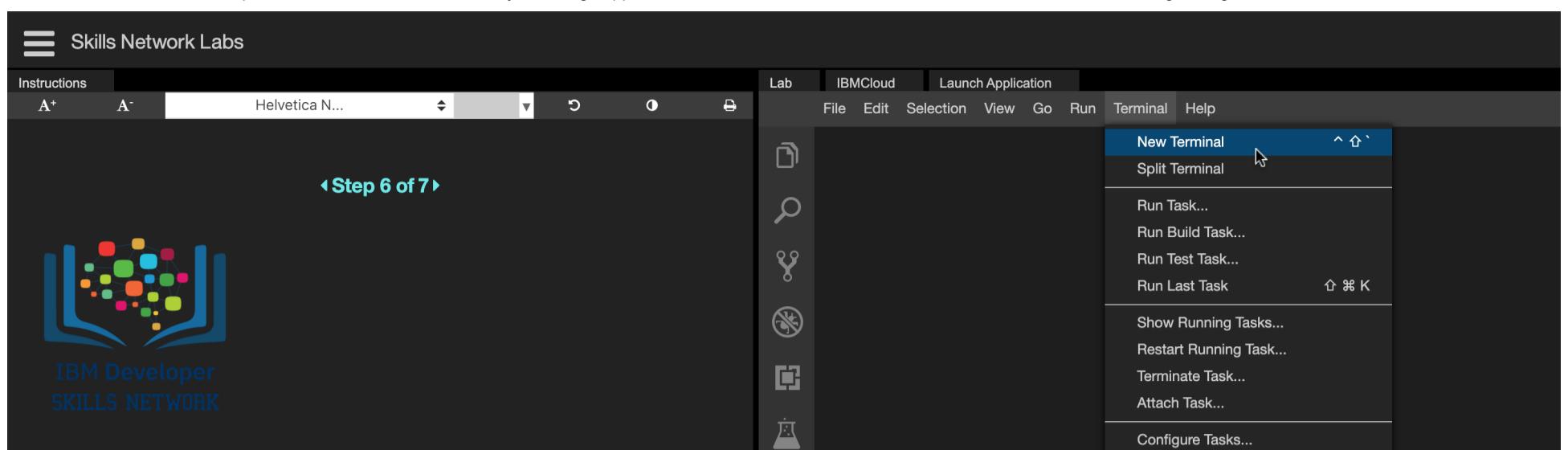
Example Exercise

In this example exercise through different tasks, first you will 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 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 below in the terminal. 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.

```
start_postgres
```

```
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$
```

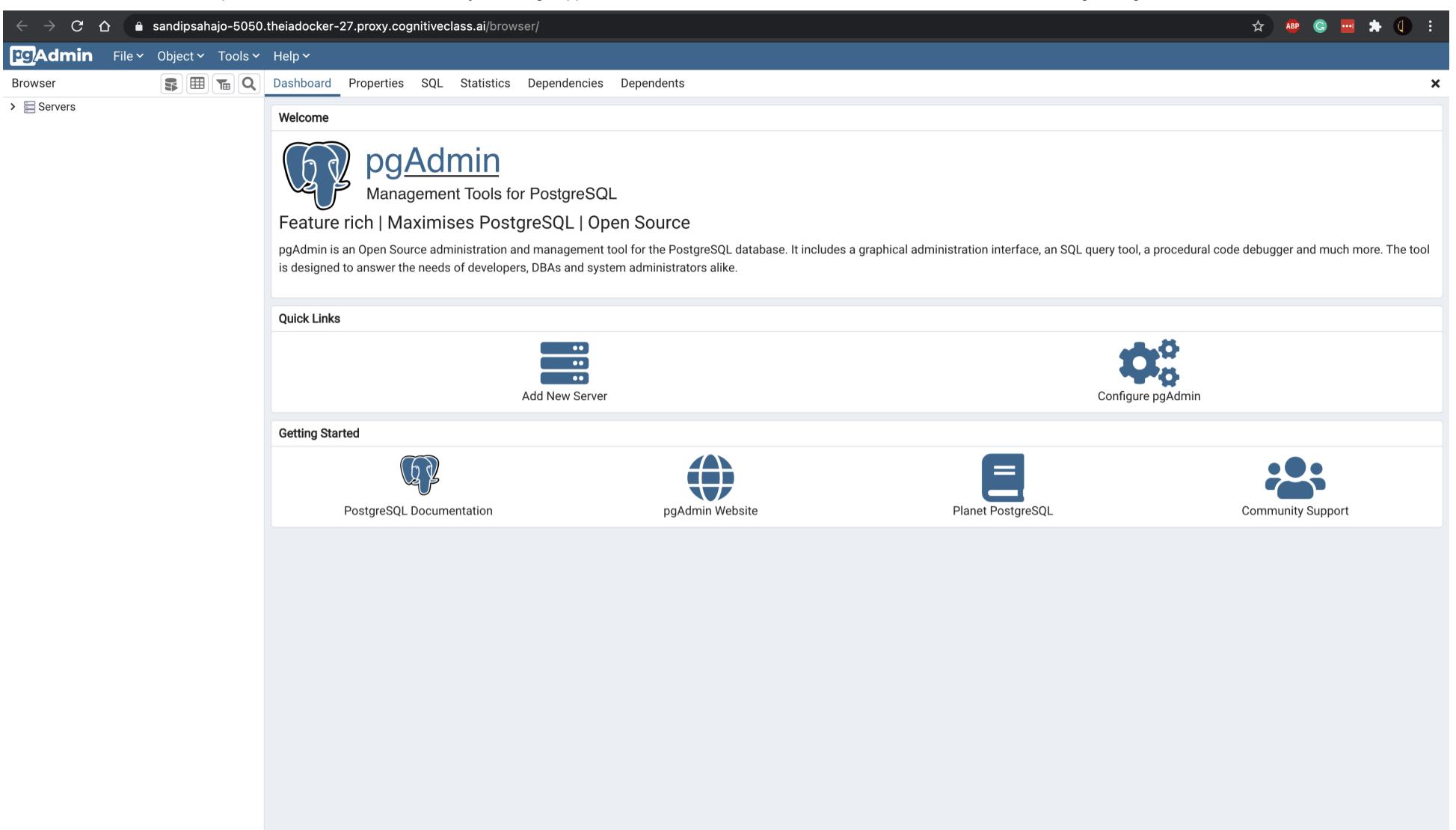
3. Copy your pgAdmin weblink from the highlighted location of the terminal shown in the image below and paste it to a new tab of your web browser.

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

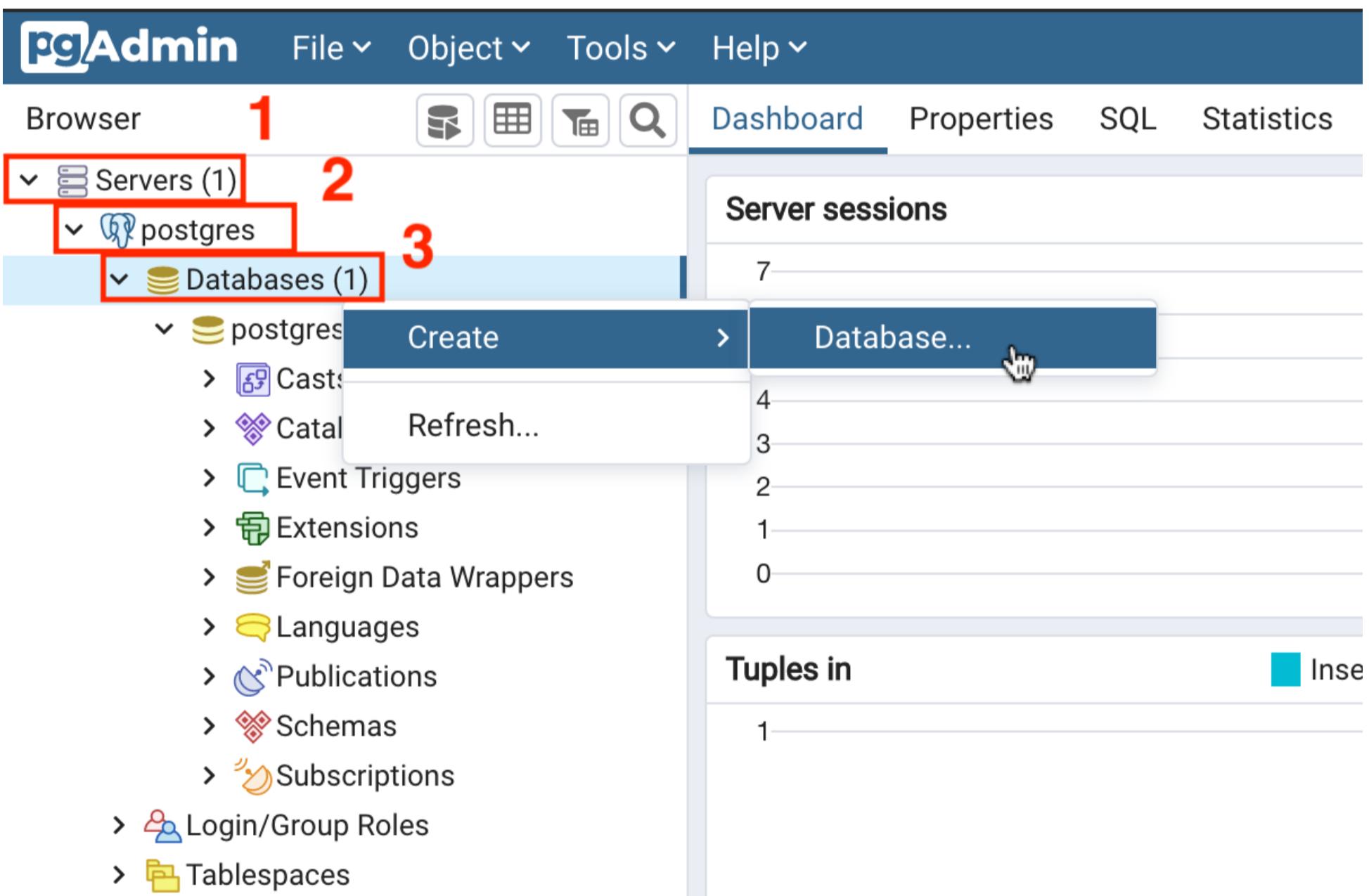
Postgres database started, waiting for all services to be ready....
[/>
Your Postgres database is now ready to use and available with username: postgres password: MTQ5NTItc2FuZGlw

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.



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 name of the database and click **Save**.



Create - Database

X

General Definition Security Parameters Advanced SQL

Database **HR**

Owner **postgres**

Comment

i **?** **Cancel** **Reset** **Save**

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

The screenshot shows the pgAdmin interface. In the top navigation bar, there are tabs for 'File', 'Object', 'Tools', and 'Help'. Below the navigation bar is a toolbar with icons for 'New Query', 'New Object', 'New Table', and 'Search'. On the left, the 'Browser' panel displays a tree structure of database objects. Under 'Servers (1) > postgres > Databases (2)', the 'HR' database is selected and highlighted with a red box. A context menu is open over the 'HR' database, listing various options: 'Create', 'Refresh...', 'Delete/Drop', 'CREATE Script', 'Disconnect Database...', 'Generate ERD (Beta)' (which is highlighted with a blue background and a cursor icon), 'Maintenance...', 'Backup...', 'Restore...', 'Grant Wizard...', 'Search Objects...', 'Query Tool', and 'Properties...'. To the right of the context menu, there is a 'Dashboard' area with several cards.

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

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents Untitled

Servers (1) postres Databases (2) HR

Add table Option Ctrl A

New table General Columns

Name: employees Schema: public Comment:

Cancel OK

8. Switch to the **Columns** tab and click the **Add new row** button 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

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	employee_id	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	first_name	character varying	20		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	last_name	character varying	25		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	email	character varying	100		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	phone_number	character varying	20		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	hire_date	date			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	job_id	character varying	10		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	salary	numeric	8	2	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	commission_pct	numeric	2	2	<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	manager_id	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	department_id	integer			<input type="checkbox"/> No	<input type="checkbox"/> 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 name of the table. Don't click **OK**. Switch to tab **Columns** and click the **Add new row** button 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 name of the table. Don't click **OK**. Switch to tab **Columns** and click the **Add new row** button 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			Yes	Yes
	department_name	character varying	30		Yes	No
	manager_id	integer			No	No
	location_id	integer			No	No

X 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 name of the table. Don't click **OK**. Switch to tab **Columns** and click the **Add new row** button 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

X Cancel**OK**

Table: locations (public)

General Columns

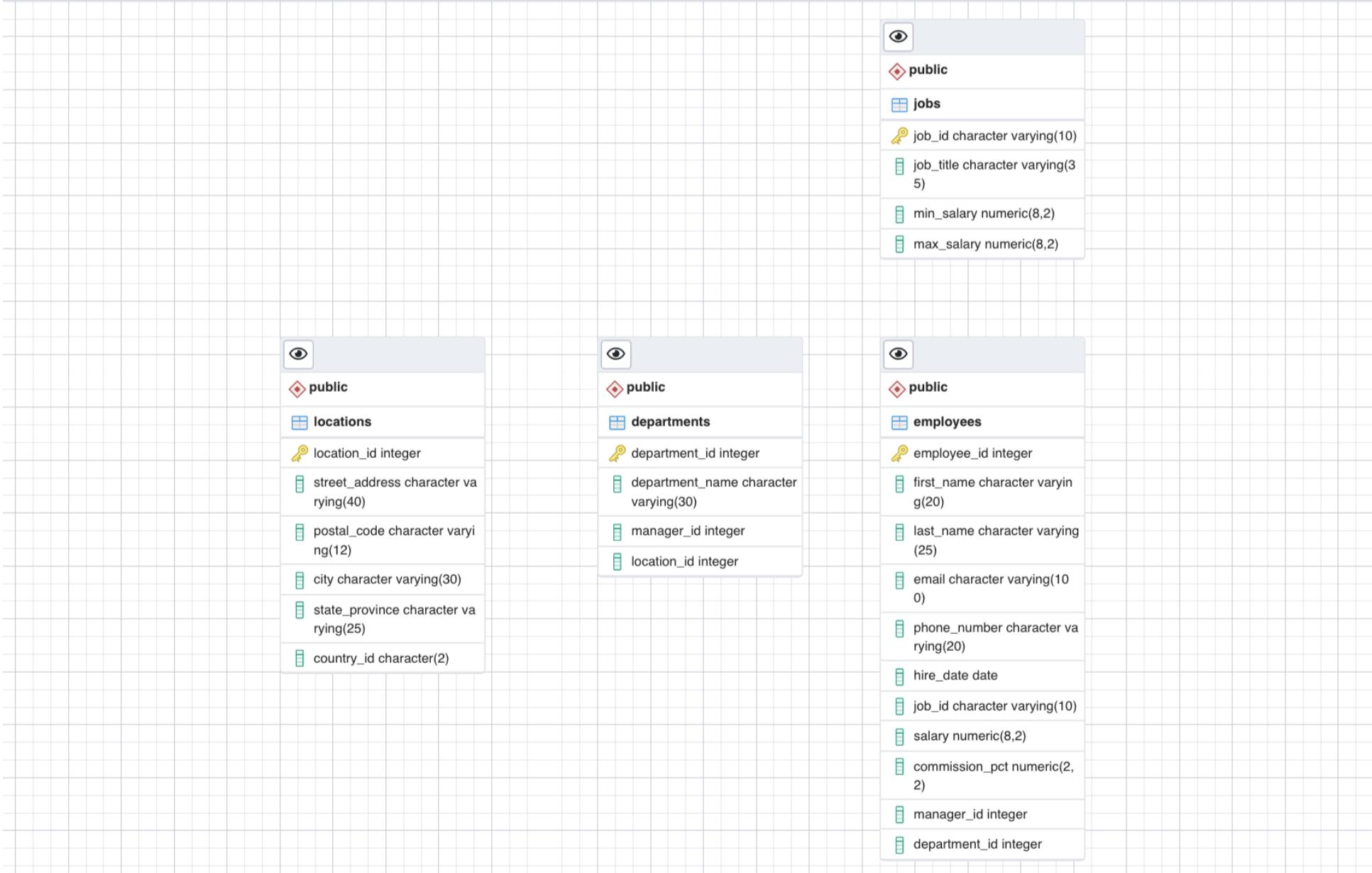
Columns					
	Name	Data type	Length/Precision	Scale	Not NULL? Primary key?
<input checked="" type="checkbox"/>	location_id	integer			<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	street_address	character varying	40		<input type="checkbox"/>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

 Cancel OK

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

Properties SQL Statistics Dependencies Dependents 

/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 the **One-to-Many link** button. 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 viewer interface with the following details:

Toolbar: Includes icons for file operations (New, Open, Save, Print, etc.), a magnifying glass, and other database management tools. The "1M" button is highlighted with a red box.

Connection: HR/postgres@localhost

Link Type: One-to-Many link

Buttons: Option, Ctrl, O

Tables:

- employees** (highlighted with a red border):
 - public
 - employee_id integer (PK)
 - 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
- jobs**:
 - public
 - job_id character varying(10)
 - job_title character varying(35)
 - min_salary numeric(8,2)
 - max_salary numeric(8,2)
- departments**:
 - public
 - department_id integer (PK)
 - department_name character varying(30)
 - manager_id integer
 - location_id integer
- locations**:
 - public
 - location_id integer (PK)
 - street_address character varying(40)
 - postal_code character varying(12)
 - city character varying(30)
 - state_province character varying(25)
 - country_id character(2)

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 the **One-to-Many link** button. 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 the **One-to-Many link** button. 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 the **One-to-Many link** button. 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**.

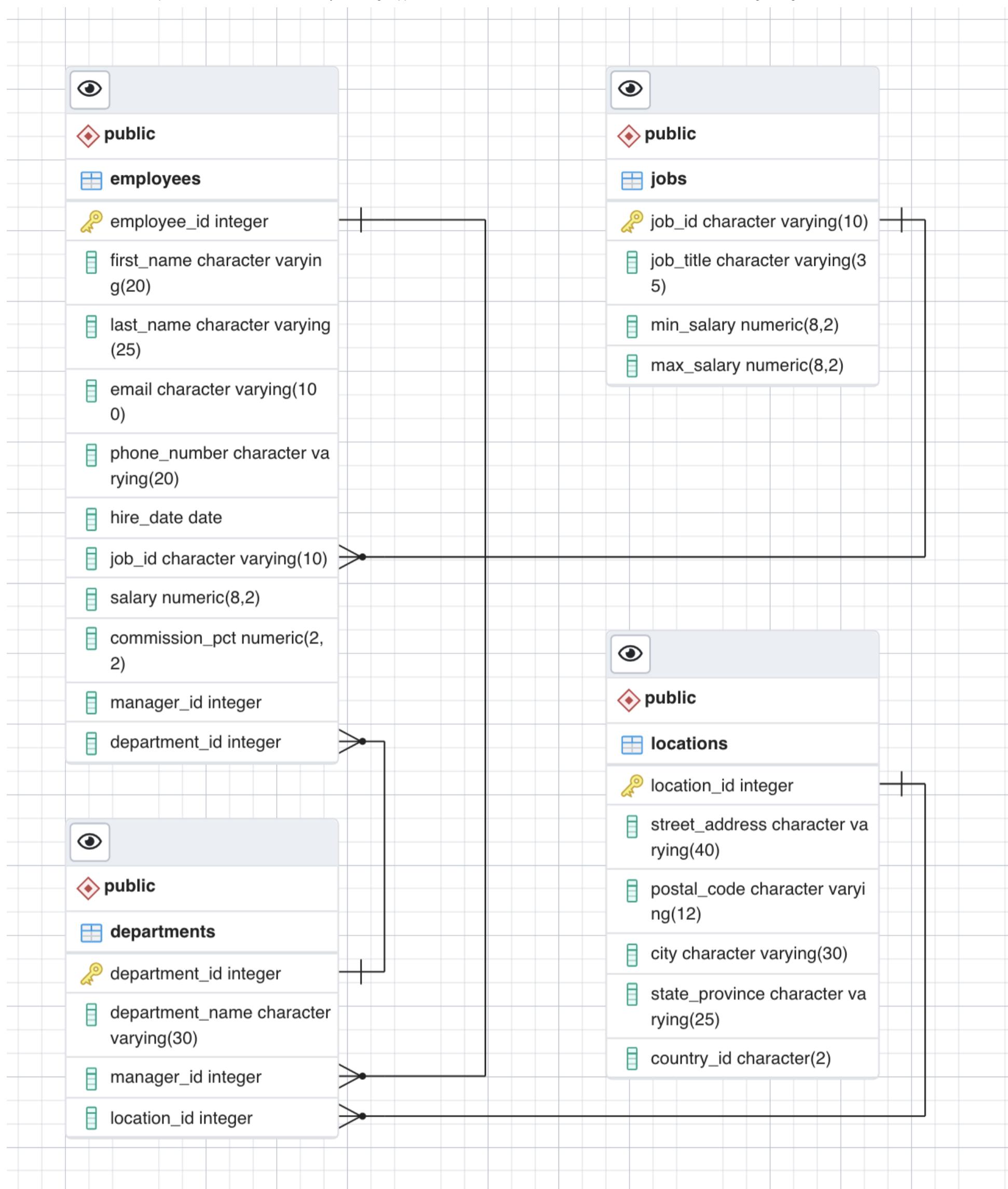
One to many relation

General

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

Cancel **OK**

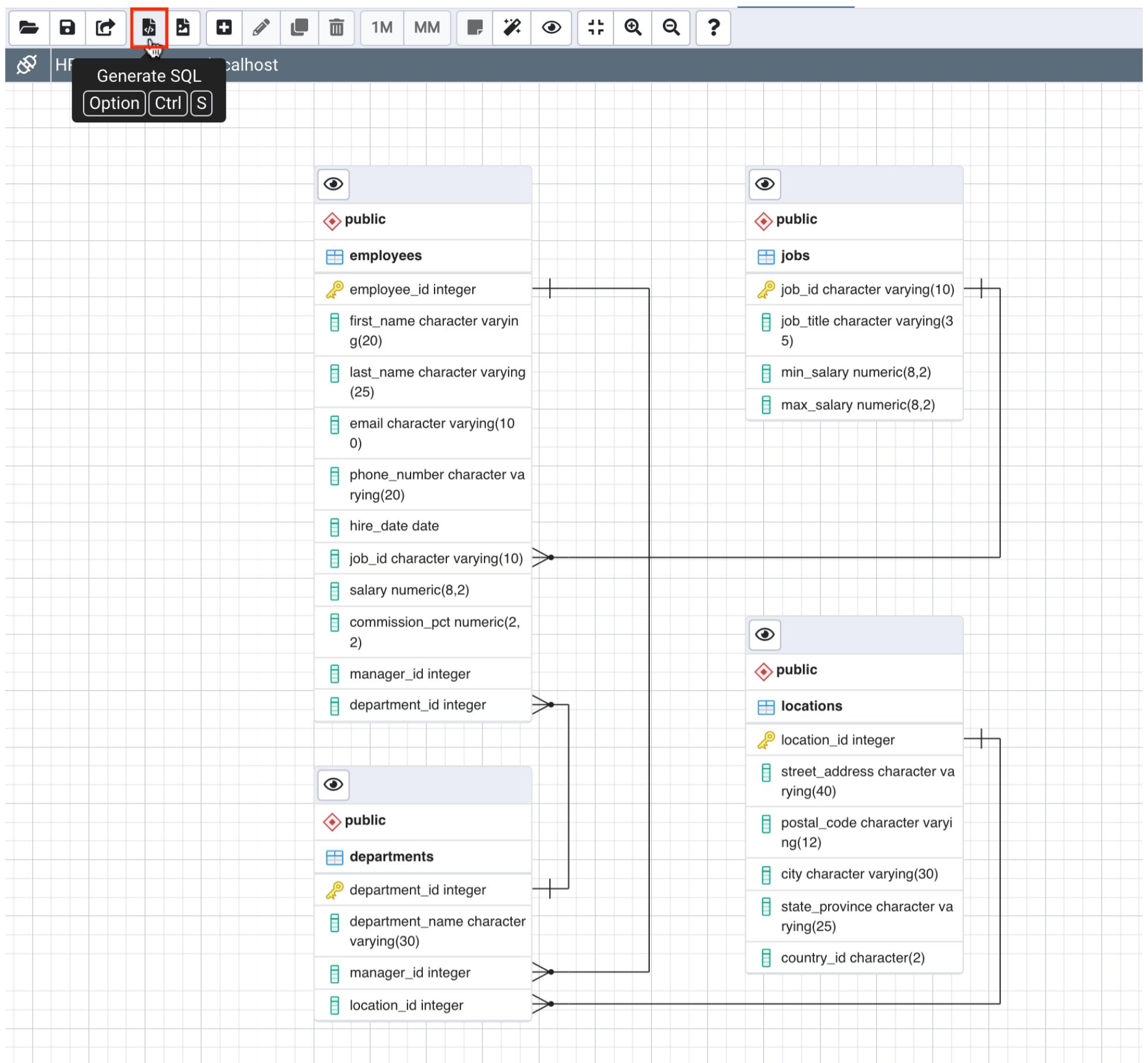
13. After creating all four relationships, you have completed the ERD for this exercise. Proceed to Task B.



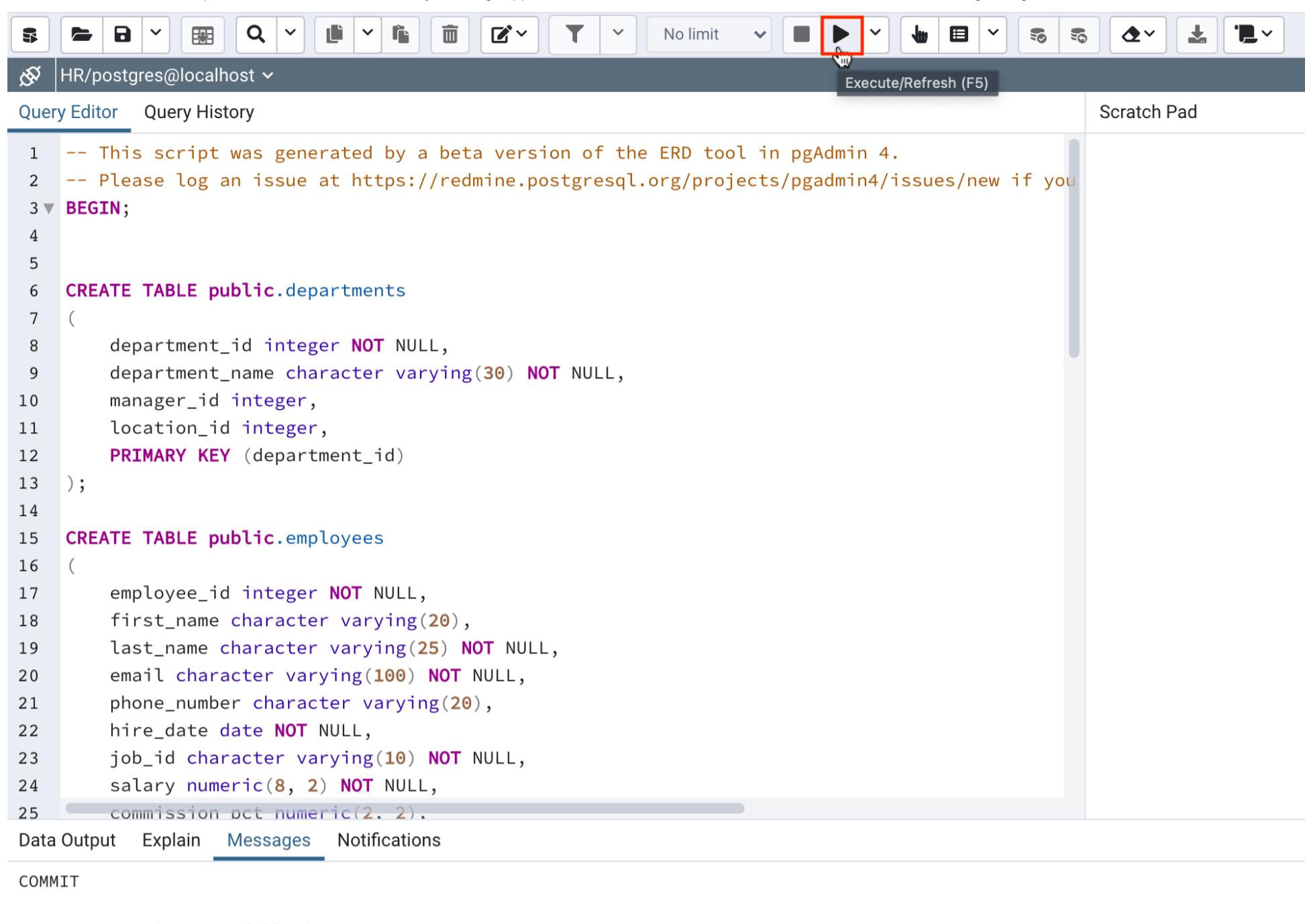
Task B: Generate and execute SQL script from ERD to create schema

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

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



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



```

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

Query Editor Query History Scratch Pad

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 storage.
 - o [HR_pgsql_dump_data_for_example-exercise.tar](#)
2. Follow the instructions below to import/restore the data:
 - o 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

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...

The screenshot shows the pgAdmin interface. The left pane displays a tree view of database objects under 'Servers (1) > postres > Databases (2) > HR'. A red box highlights the 'HR' database node. A context menu is open over the 'HR' node, listing various database management options. The 'Restore...' option is highlighted with a blue background and a hand cursor icon, indicating it is the current selection.

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

Restore (Database: HR)

General Restore options

Format	Custom or tar
Filename	<input type="text"/> ...
Number of jobs	<input type="text"/>
Role name	Select an item...

Cancel **Restore**

- Click the **Upload File** button.

Select file

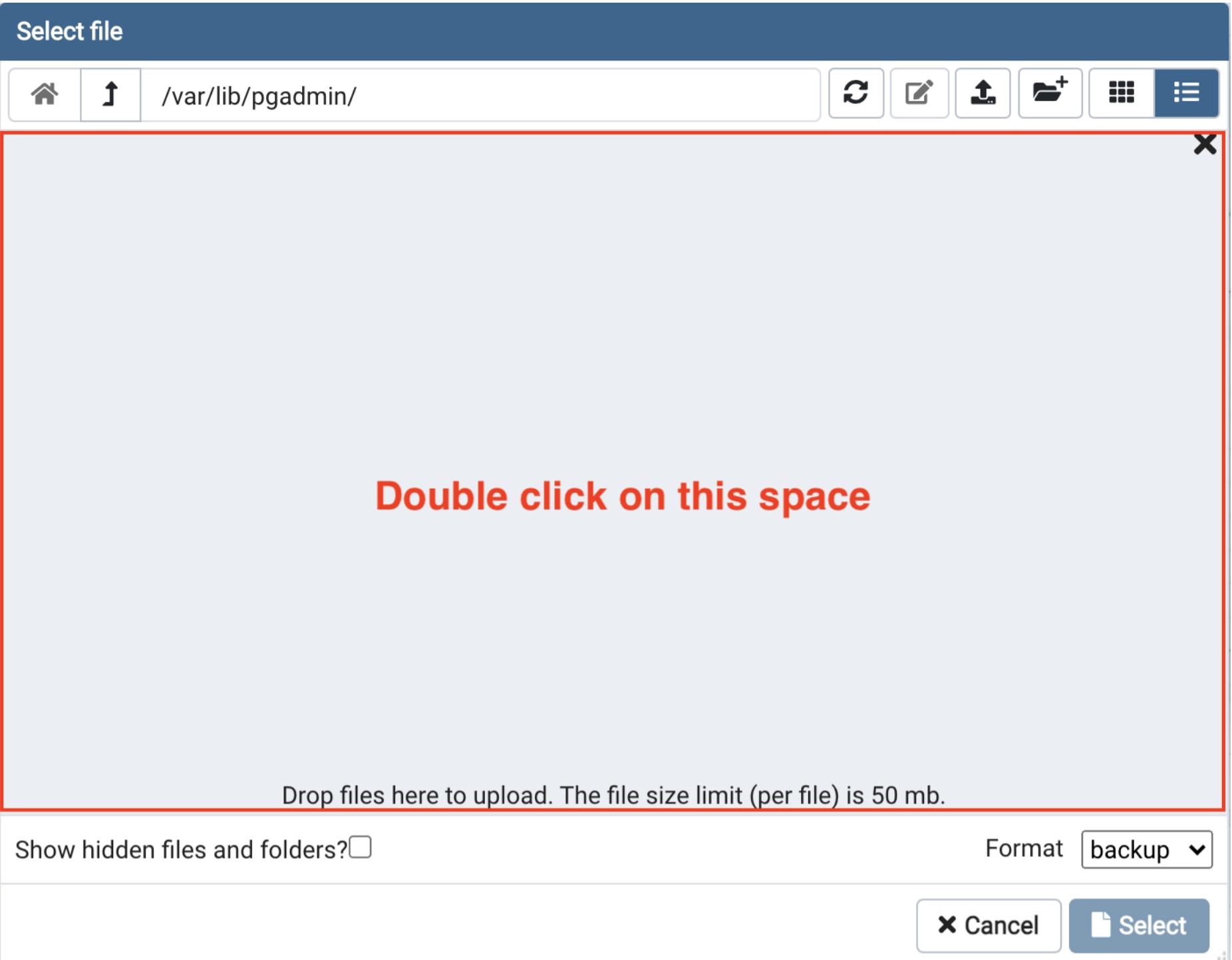
/var/lib/pgadmin/ Upload

Name	Size	Modified
sessions	4.0 kB	Mon Mar 29 10:20:20 2021
storage	4.0 kB	Mon Mar 29 10:04:10 2021

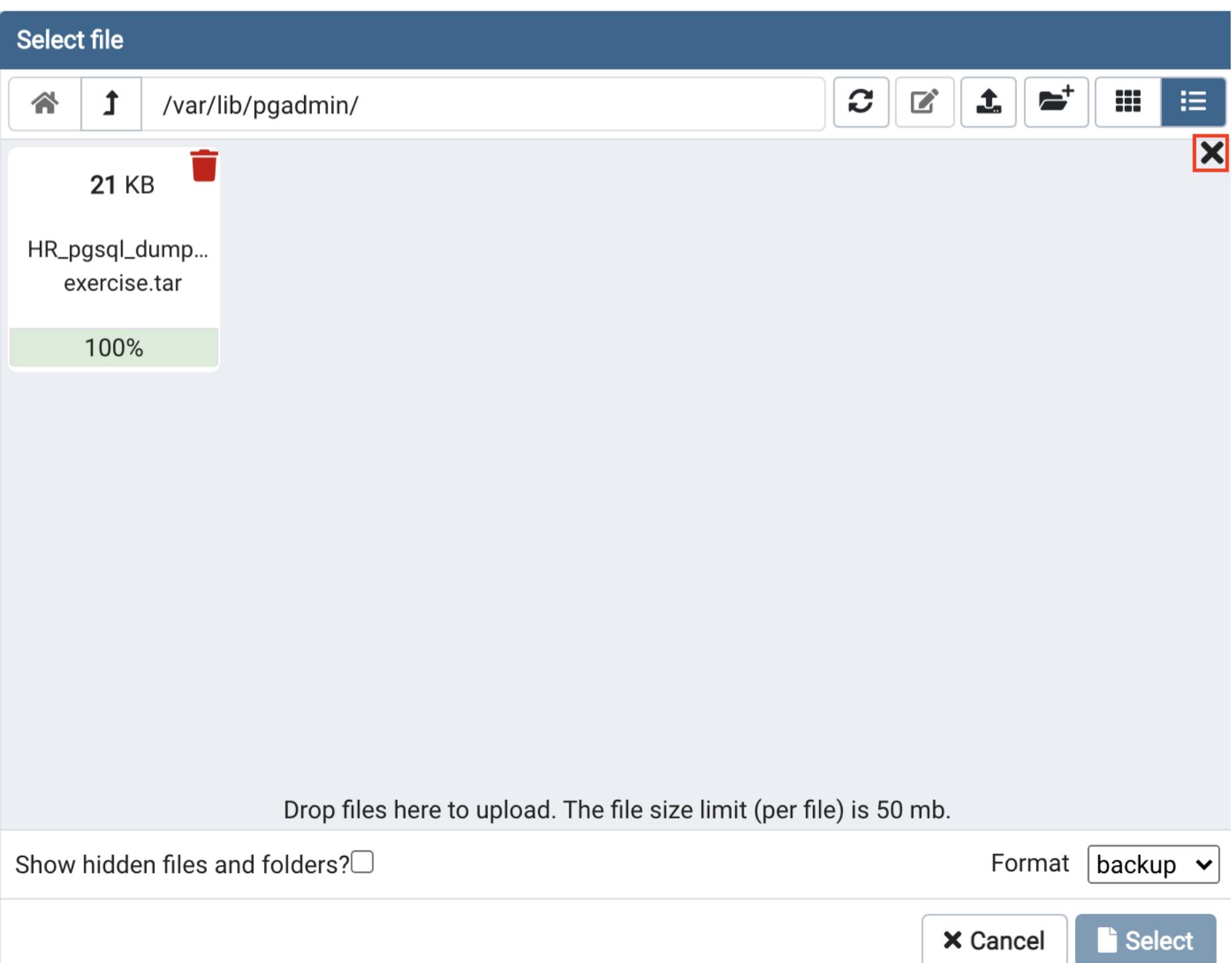
Show hidden files and folders? Format **backup**

Cancel **Select**

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



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



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

Select file

File operations:

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**

Cancel Select

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

Restore (Database: HR)

General

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** button.

[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

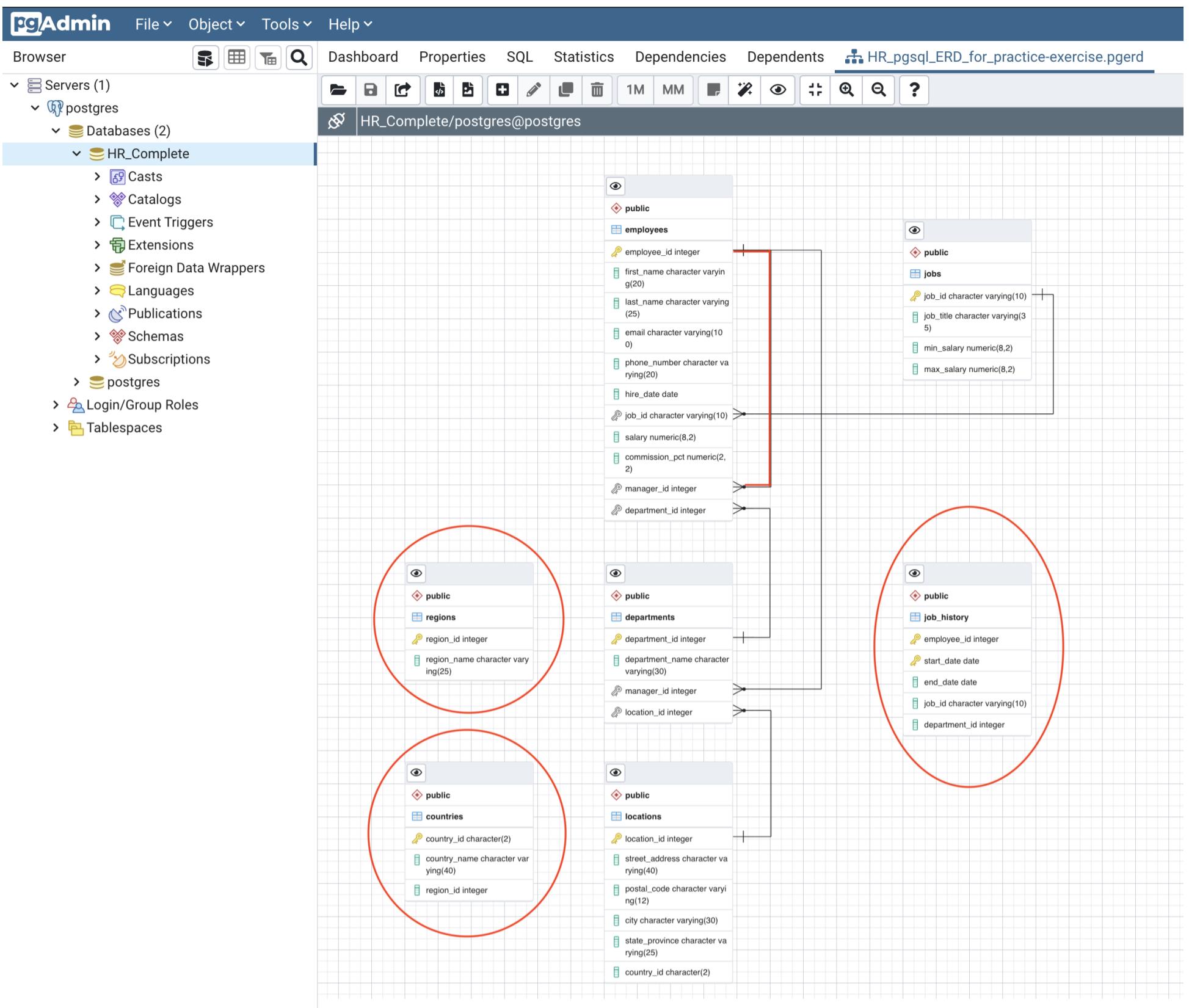
In this practice exercise, first you will finish creating a partially complete ERD for the HR database. Next, you will generate and execute an SQL script to build the complete schema of the HR database from its ERD. Finally, you will load the complete database schema with data by using restore feature.

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 Example Exercise) below to your local computer storage.
 - 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 the **Load from file** button to load the **HR_pgsql_ERD_for_practice-exercise.pgerd** file.

The screenshot shows the pgAdmin interface with the following details:

- Toolbar:** Includes File, Object, Tools, Help, and various icons for browser, dashboard, properties, SQL, statistics, dependencies, dependents, and an untitled tab.
- Servers:** A tree view under 'Servers (1)' shows a single entry for 'postgres' which contains 'Databases (2)' with 'HR_Complete' selected.
- Tool Buttons:** A toolbar at the top has a folder icon highlighted with a red box, followed by other icons for file operations like copy, paste, refresh, etc.
- Context Menu:** A context menu is open over the 'HR_Complete' database entry, with the 'Load from file' option highlighted and a keyboard shortcut 'Ctrl+O' displayed.
- Tip Box:** A callout box labeled 'Tip:' contains the text: '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 as well as one new relationship within the entity diagram of the **employees** table between *manager_id* as local column and *employee_id* as referenced column.



5. Create the remaining relationships between the tables:

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

Select the entity diagram **countries** and click the **One-to-Many link** button. 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 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	employee_id	✖ ▼
Referenced Table	(public) employees	✖ ▼
Referenced Column	employee_id	✖ ▼

X Cancel **OK**

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

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

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 the **One-to-Many link** button. 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

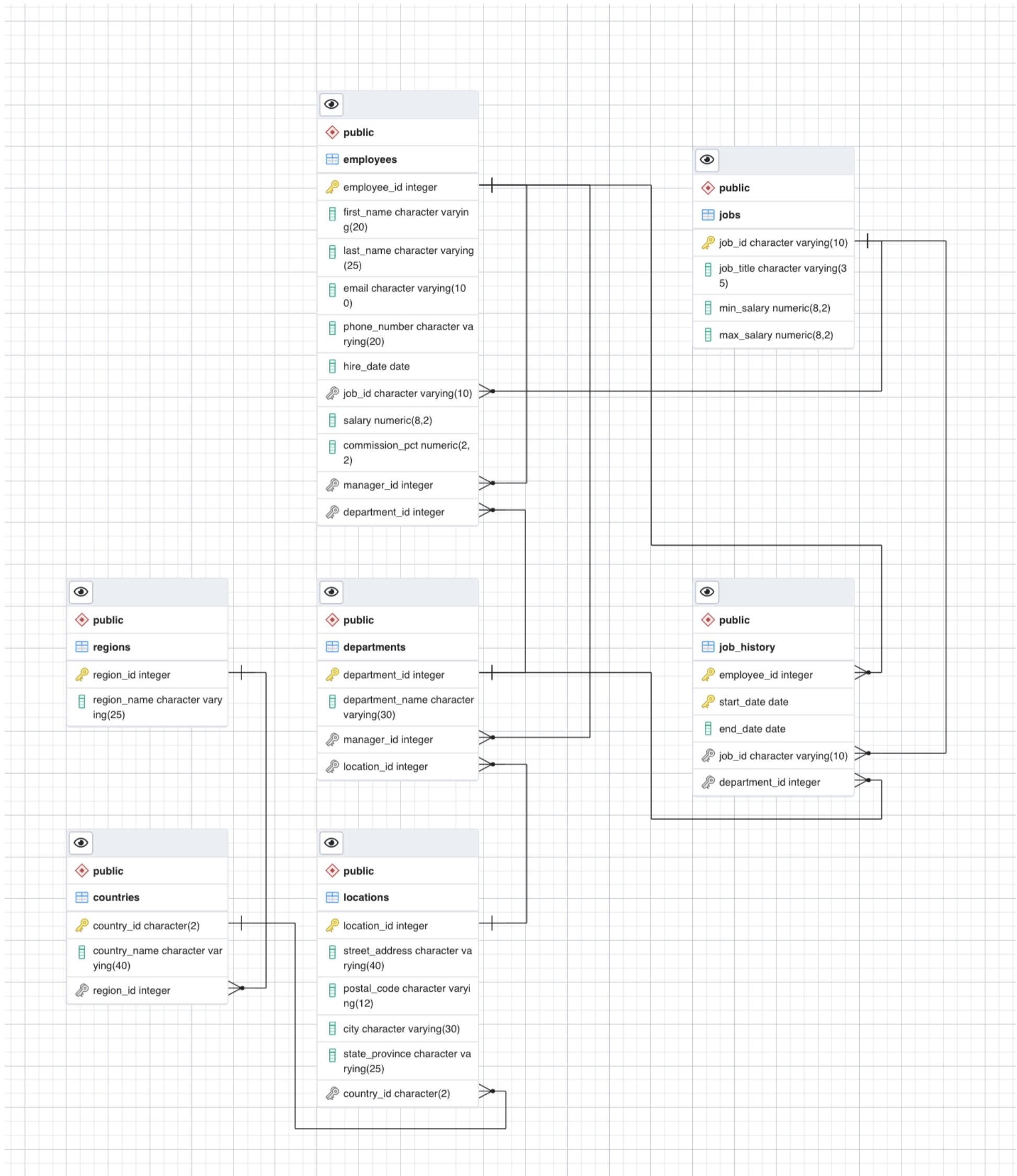


Cancel

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 storage. 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.

Congratulations! You have completed this lab, and you are ready for the next topic.

Author(s)

- [Sandip Saha Joy](#)

Other Contributor(s)

-

Changelog

Date	Version	Changed by	Change Description
2021-03-31	1.0	Sandip Saha Joy	Created initial version

© IBM Corporation 2021. All rights reserved.