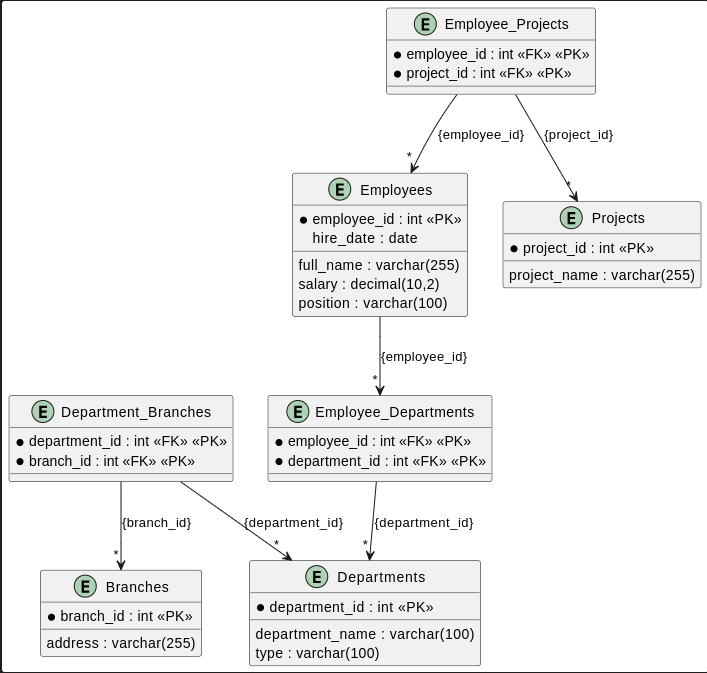
**Эталонное решение самостоятельного задания по теме «Базы данных»**

**Решение задания 1**

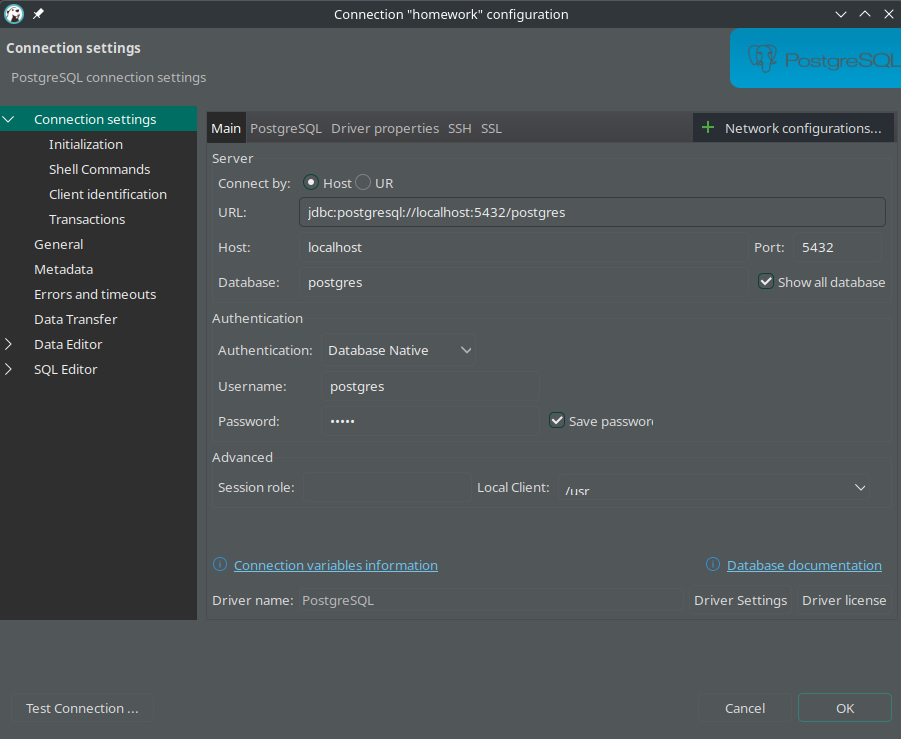


**Решение задания 2\***

Разворачиваем контейнер с postgres:

docker run --name hw\_db -p 5432:5432 -e POSTGRES\_PASSWORD=12345 -d postgres:latest

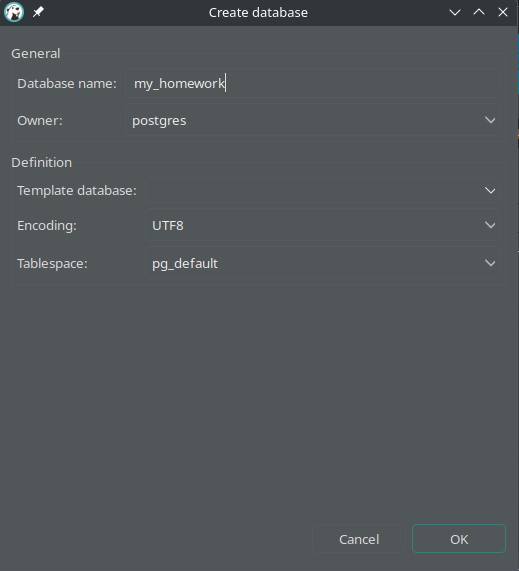
Подключаемся к созданной базе с помощью dbeaver.



Не забываем установить галочку Show all database чтобы наше подключение имело доступ ко всем базам данных на нашем узле.

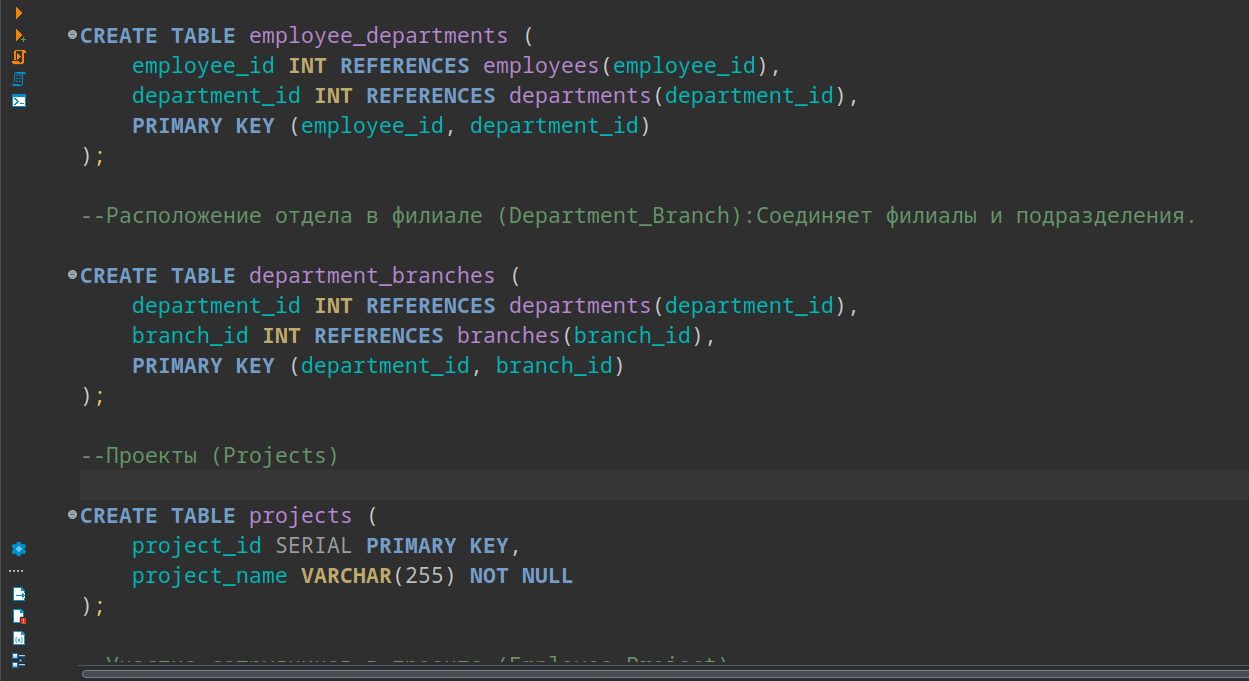
После того как подключение будет произведено в области Database Navigator, разворачиваем древовидную структуру нужного нам подключения, находим директорию Databases, щёлкаем по ней правой кнопкой и выбираем пункт меню Create New Database.

В появившемся окне



вводим название базы данных (например, my\_homework) и нажимаем ОК.

Затем разворачиваем директорию Databases в области Database Navigator и находим вновь созданную базу данных my\_homework. Щёлкаем по ней правой кнопкой мыши, выбираем меню SQL Editor, а из всплывшего подменю New SQL Script откроется текстовый редактор SQL,



где вы сможете писать и редактировать SQL запросы.

Если схема вашей модели данных такая же, как в эталонном решении к заданию 1, то SQL для описания этой схемы:

--Сотрудники (Employees)

**CREATE** **TABLE** employees (

employee\_id SERIAL **PRIMARY** **KEY**,

full\_name **VARCHAR**(255) **NOT** **NULL**,

salary **DECIMAL**(10, 2),

position **VARCHAR**(100),

hire\_date **DATE**

);

--Отделения (Departments)

**CREATE** **TABLE** departments (

department\_id SERIAL **PRIMARY** **KEY**,

department\_name **VARCHAR**(100),

type **VARCHAR**(100)

);

--Филиалы (Branches)

**CREATE** **TABLE** branches (

branch\_id SERIAL **PRIMARY** **KEY**,

address **VARCHAR**(255)

);

--Назначение сотрудника в отдел (Employee\_Department):Эта таблица соединяет сотрудников с подразделениями.

**CREATE** **TABLE** employee\_departments (

employee\_id **INT** **REFERENCES** employees(employee\_id),

department\_id **INT** **REFERENCES** departments(department\_id),

**PRIMARY** **KEY** (employee\_id, department\_id)

);

--Расположение отдела в филиале (Department\_Branch):Соединяет филиалы и подразделения.

**CREATE** **TABLE** department\_branches (

department\_id **INT** **REFERENCES** departments(department\_id),

branch\_id **INT** **REFERENCES** branches(branch\_id),

**PRIMARY** **KEY** (department\_id, branch\_id)

);

--Проекты (Projects)

**CREATE** **TABLE** projects (

project\_id SERIAL **PRIMARY** **KEY**,

project\_name **VARCHAR**(255) **NOT** **NULL**

);

--Участие сотрудников в проекте (Employee\_Project)

**CREATE** **TABLE** employee\_projects (

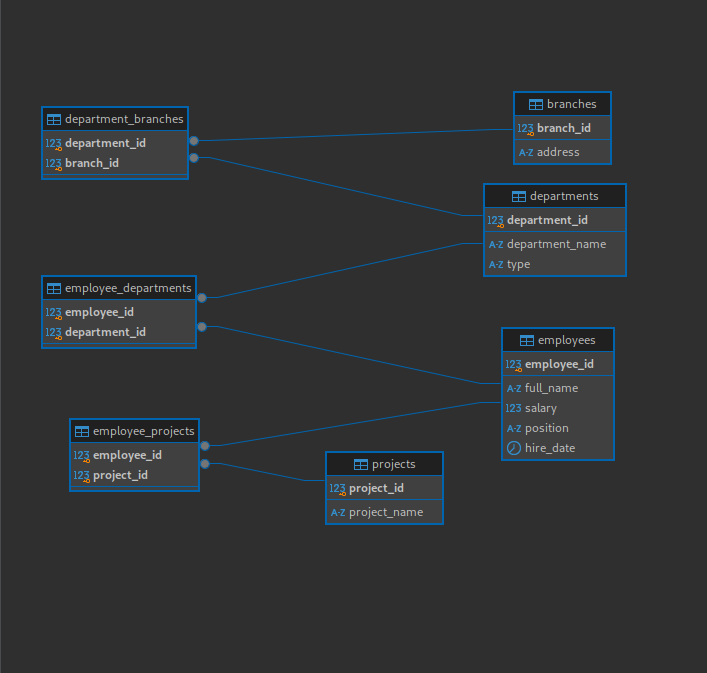
employee\_id **INT** **REFERENCES** employees(employee\_id),

project\_id **INT** **REFERENCES** projects(project\_id),

**PRIMARY** **KEY** (employee\_id, project\_id)

);

Диаграмма схемы БД из dbeaver:



Databases -> ваша\_база\_данных -> Schemas -> ваша\_схема

Во всплывающем меню выбрать View Diagram или использовать горячие клавиши, нажав ctrl+shift+enter

