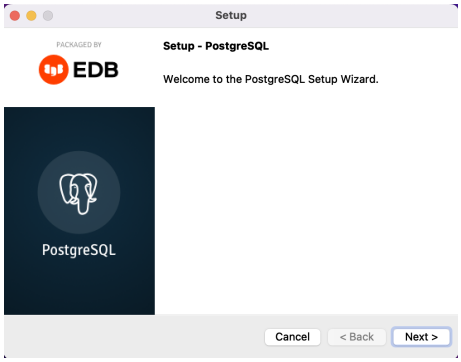
DBeaver and PostgreSQL test

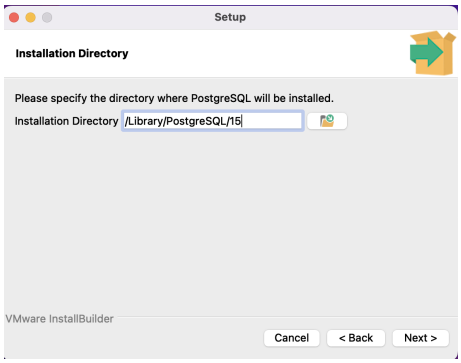
This part of the introduction is to help you download PostgreSQL (our chosen language for the course), and DBeaver (a helpful application that provides intuitive GUI to PostgreSQL); set up the database and run tests with mock data.

Download DBeaver and PostgreSQL

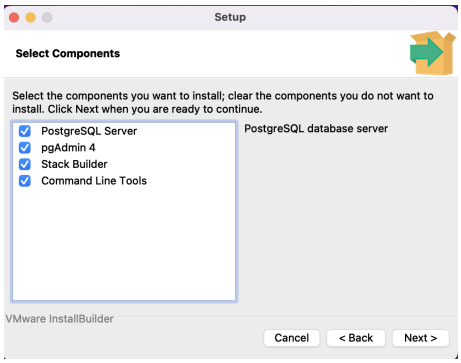
1. Go to dbeaver.io and find the suitable version of DBeaver suitable to your machine to download.
2. Go to PostgreSQL download page to find the suitable PostgreSQL.
3. Once you have downloaded the installer, run the setup file and you will be greeted with this window. Click on **Next** button to proceed.



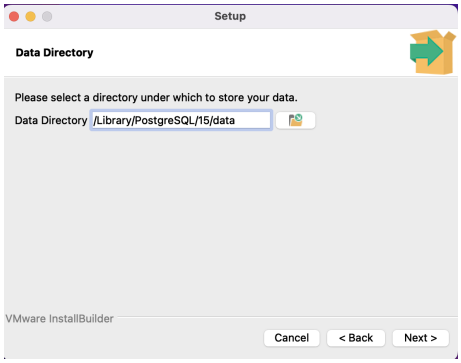
4. First, choose the directory that PostgreSQL will be installed into. Unless you have a specific place in mind, it is generally good to leave it to default location.



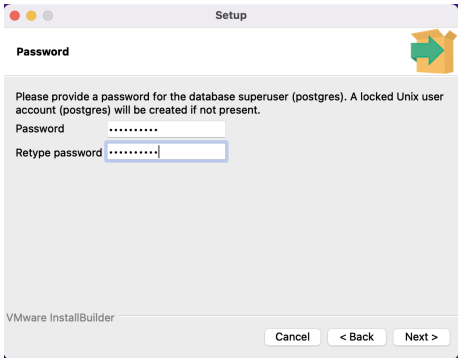
5. This allows you to select the components to install. Installing components other than PostgreSQL is recommended for possible usage in the future.



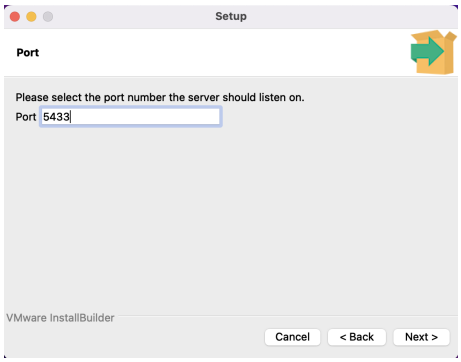
6. Specify the data folder. Again, be cautious of where you keep the data.



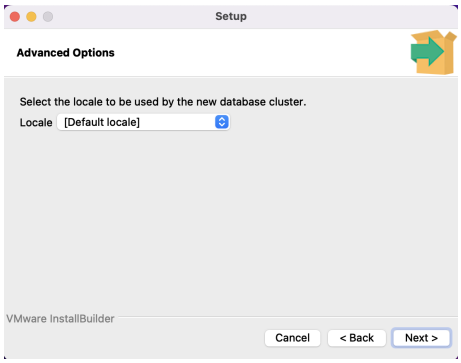
7. Next, create a password for the database superuser (which is named by default "postgres"). This is important to remember as you will need it when using PostgreSQL.



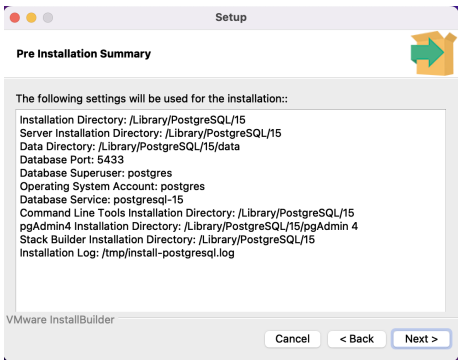
8. Specify the port number of the server. In PostgreSQL, it's typical to use port 5432 if it is available. If it isn't, most installers will choose the next free port, usually 5433. You can generally leave it as it is.



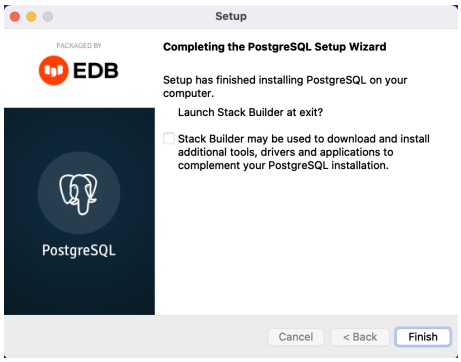
9. This is an advanced options to set the location for the database cluster, which should be left by default unless you know what you are doing.



10. Confirm the details of your installation



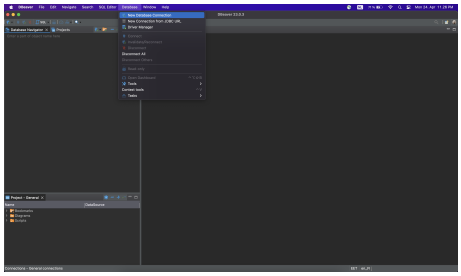
11. After everything has been installed, you may want to download additional tools with StackBuilder. However, this is not needed so you can just uncheck the box and **Finish** the installation.



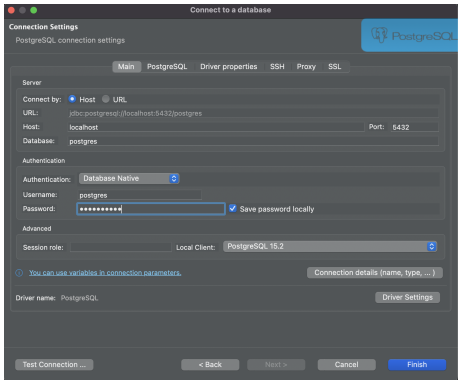
Create database from .sql file template

The example file is available at MyCourses (exercise sessions > week 1 section)

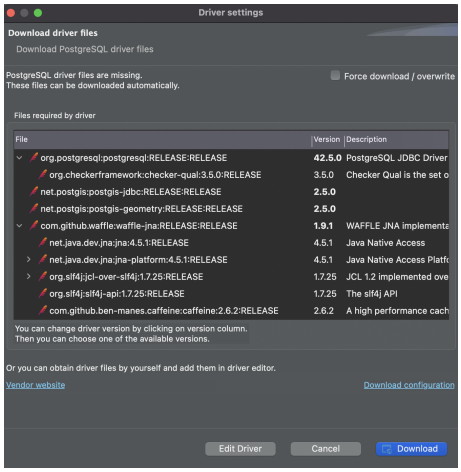
1. When opening DBeaver for the first time, you will be asked whether to create a sample database. You do not need to for this course, but it would not hurt to explore on your own. TAs will however not offer help for matters relating to this.
2. In DBeaver, in **Database**, choose **New Database Connection**



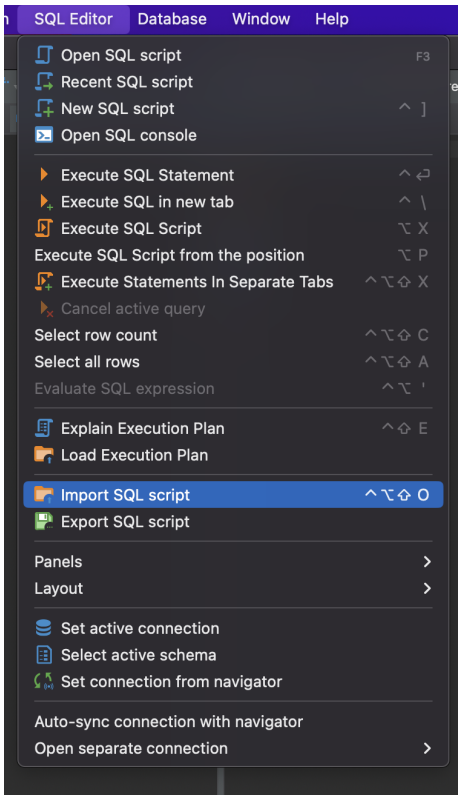
3. Select **PostgreSQL**
4. In connection settings, select **Connection by host** (which should be the default) and input your password. Other credentials can be left as they are.



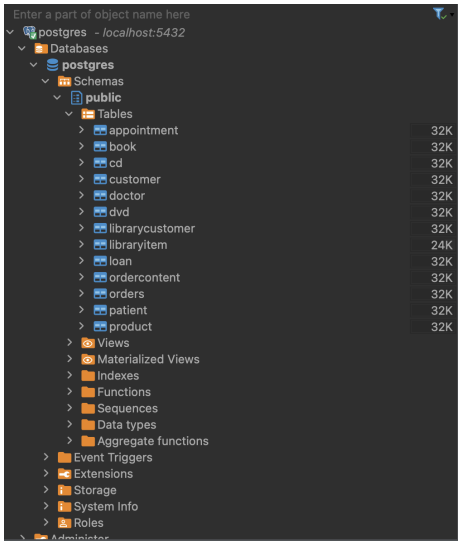
5. If a window of needed drivers is prompted, download them since they are crucial for running the database.



6. To populate your database, you can import them directly by using **SQL Editor**, **Import SQL script** and navigate to your saved .sql file.



7. After connecting to the database and populate it with data, 13 tables should be visible. You can verify it by clicking on the database, then **Schemas** → **public** → **Tables**. All tables should be listed there.

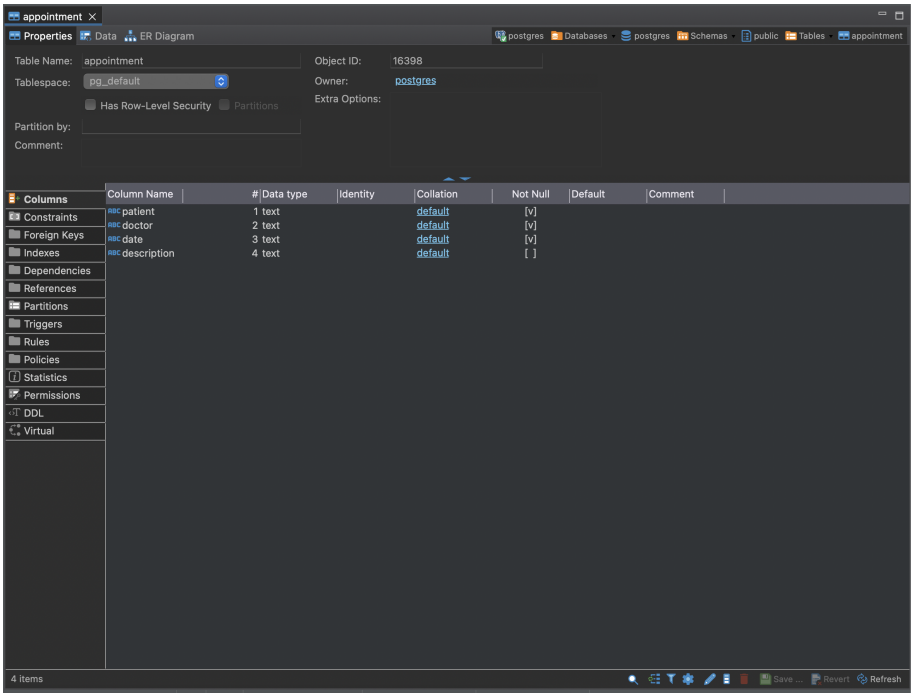


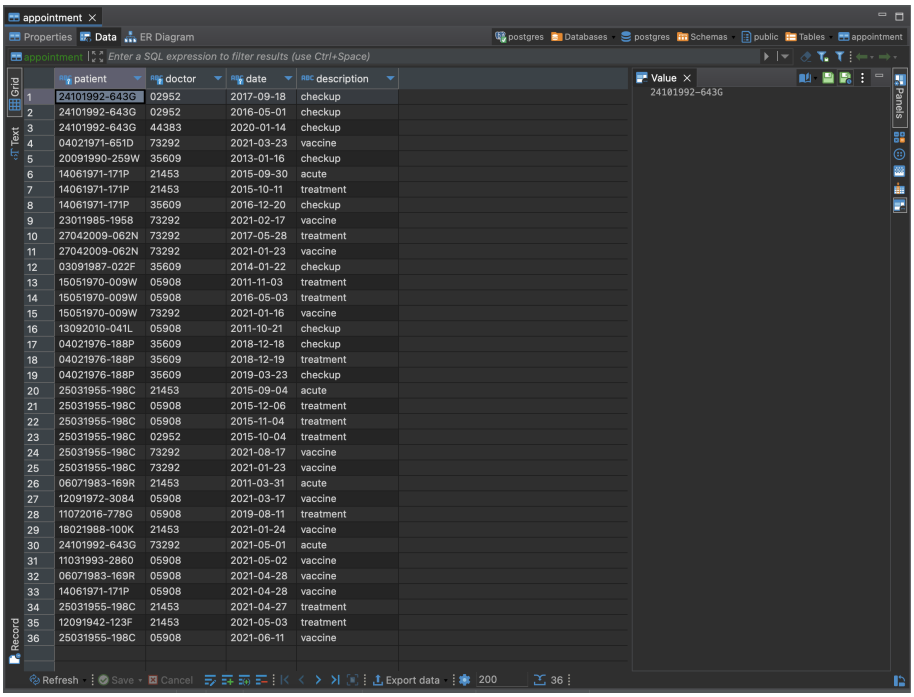
Information about a table

By double-clicking a table, you should be able to see the tab Properties which contains information about the table. The most relevant tabs this session are "Columns" and "Constraints". In the "Columns" you can see information about the attributes. In the "Constraints", you can see the primary key (or keys) for each table. For an example, can you tell what the primary keys of the table "loan" are?

In the tab "Data" you can review the rows of the table.

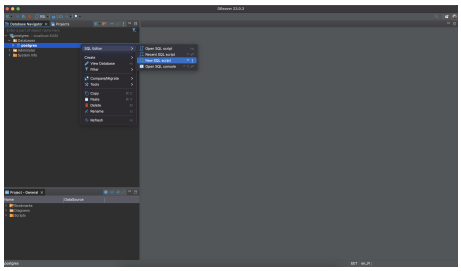
You can exit the Table Info view by clicking the small "x" in the upper-right corner of the table tab.



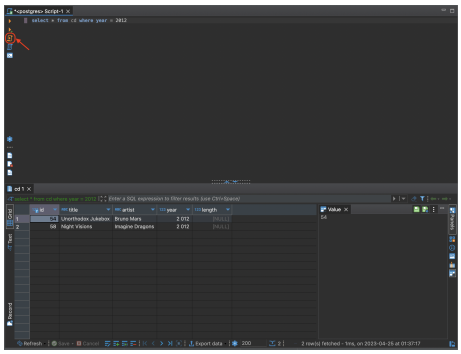


Testing your queries in PostgreSQL

1. Create a new script to run queries on your database. On the left panel, right click on your database, choose **SQL Editor**, **New SQL script**.



2. You can now write the queries to the script. To run the queries you can run the whole script at once by pressing **Execute SQL Script**. You can also then click the small blue orange button to run the code. If you have multiple SQL statements in the same code, you should highlight all the statements you want to run. The results of a successful query should appear below.



Example – Building a large SQL query

Find the details (name, phone number, and email) of customers who have currently loaned books from authors that have published books in English before the year 2000. Order the customers alphabetically and list each customer only once.

- Find authors who have published English books before 2000.

```
SELECT author FROM Book WHERE year < 2000 AND language = 'English';
```

- Find books that are currently loaned.

```
SELECT * FROM Loan JOIN Book ON itemID = ID WHERE returned = 0;
```

- Find currently loaned books from authors who have published English books before 2000.

```
SELECT *
FROM Loan JOIN Book ON itemID = ID
WHERE returned = 0 AND author IN (
    SELECT author FROM Book WHERE year < 2000 AND language = 'English'
);
```

- Find customers that have currently loaned books from authors who have published English books before 2000.

```
SELECT name, phone, email
FROM LibraryCustomer NATURAL JOIN Loan JOIN Book ON itemID = ID
WHERE returned = 0 AND author IN (
    SELECT author FROM Book WHERE year < 2000 AND language = 'English'
);
```

- Order the customers alphabetically and list each customer only once.

```
SELECT DISTINCT name, phone, email
FROM LibraryCustomer NATURAL JOIN Loan JOIN Book ON itemID = ID
WHERE returned = 0 AND author IN (
    SELECT author FROM Book WHERE year < 2000 AND language = 'English'
)
ORDER BY name;
```