Welcome to the

Java Course

Module 4 – Day 01

Content of the course

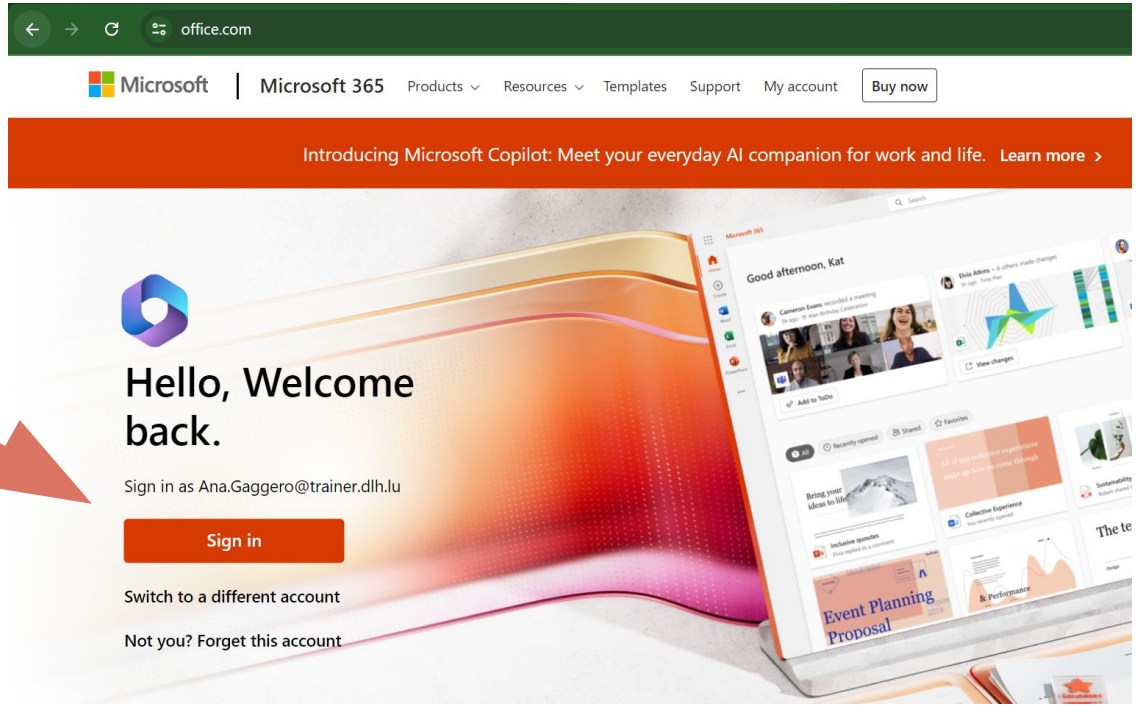
- **Introduction to Database Theory and SQL Basics**
- **Database connection with Java**
- Table Management and Relationships
- Advanced SQL Queries and Integration with Java
- Data Normalization

Log into the local PC and to Microsoft Teams

Open Google Chrome and type: office.com

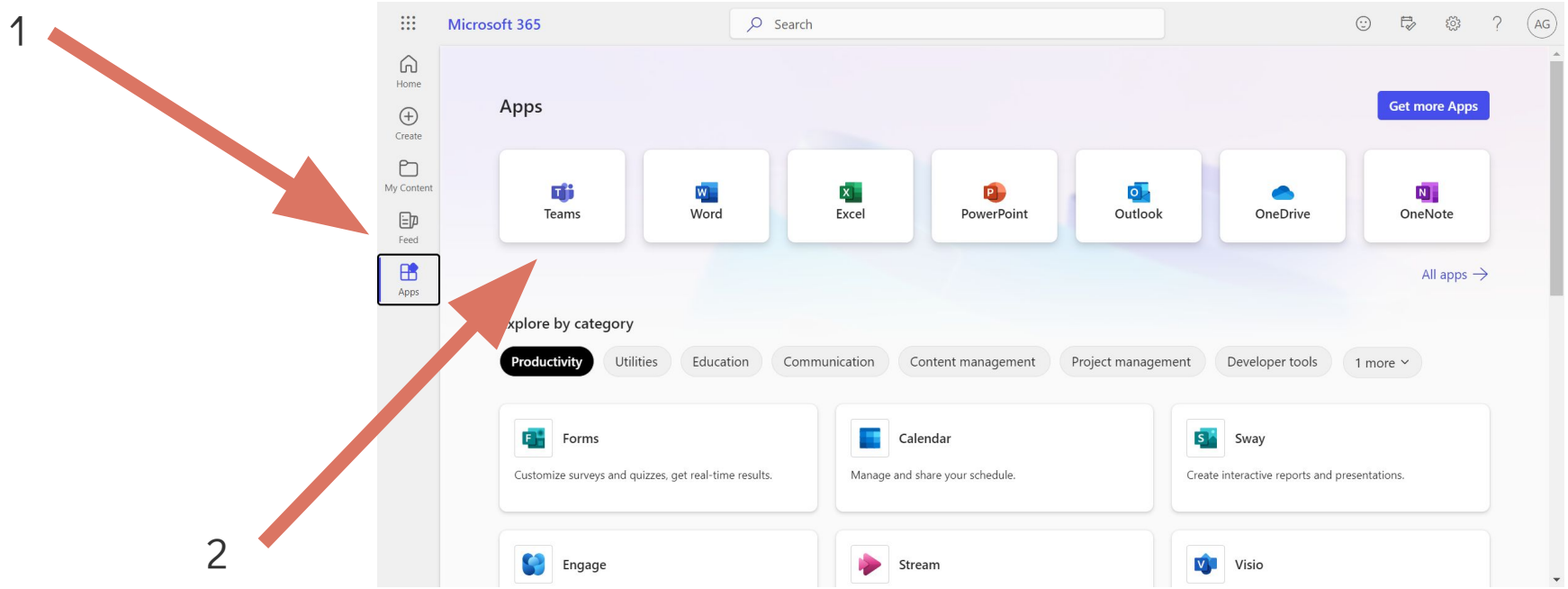
Step 1: Click on “Sign in”

You should have received your password by email.



Log into the local PC and to Microsoft Teams

Step 2: Click on “Teams”



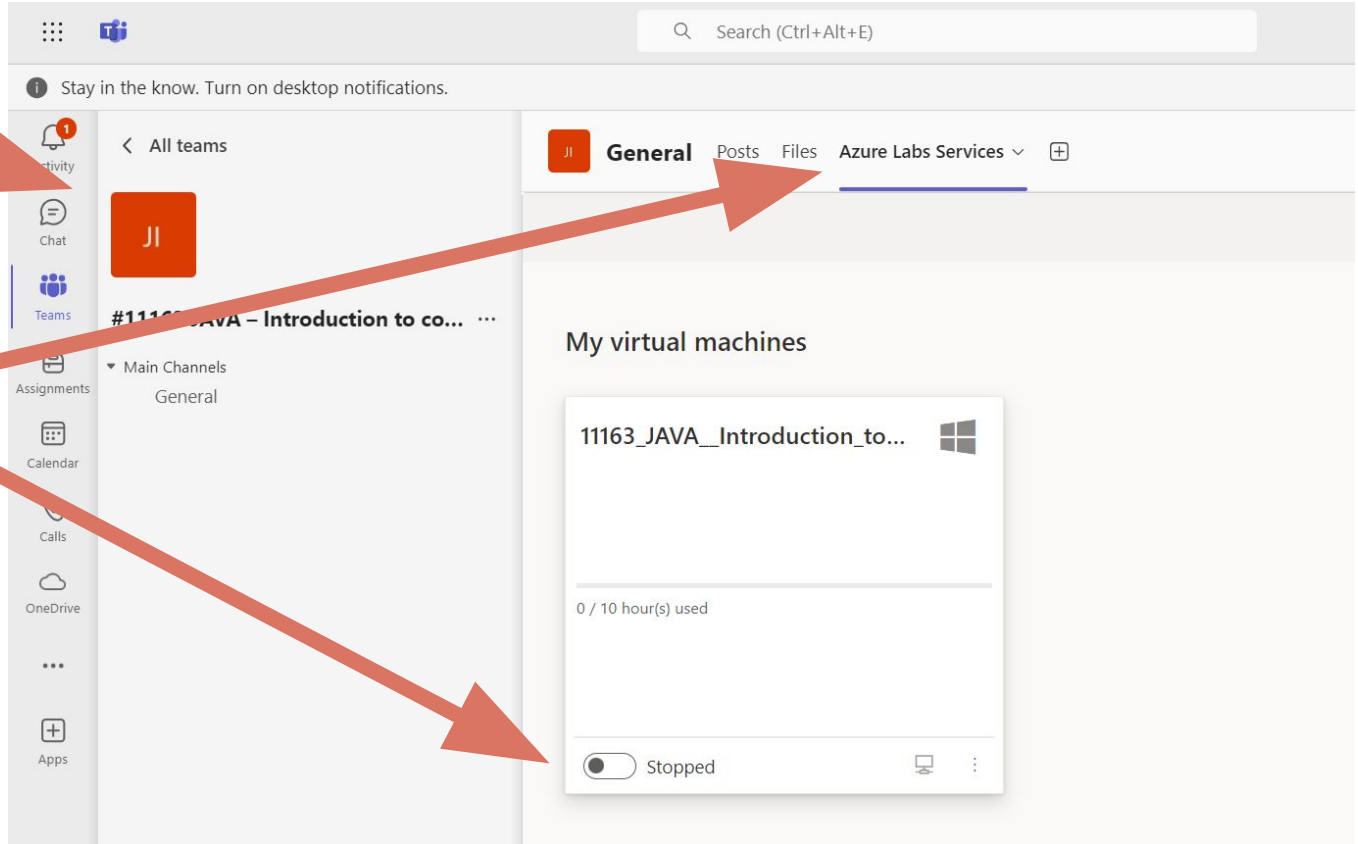
Log into the virtual machine

Step 1

Step 2

Step 3

The virtual machine will be already on. If not, turn it on, it can take up to 5 minutes to start.



Log into the virtual machine

Username: student
Password: StudentDLH2024

Event Manager Project

- **Create and manage** events.
- Manage different **types** of events (meetings, learning sessions, fun gatherings, etc...)
- **Setup** events.
- **Sell** tickets.
- Keep track of **guests**.

Databases

Database

Is an organized structure designed to **store**, **modify**, and **process** interconnected data, usually large amounts.

Importance of databases in software development

- **Data Organization and Storage**

  Databases provide a structured way to organise and store large volumes of data. Instead of saving data in unstructured files or spreadsheets, databases allow us to categorise information into tables, rows, and columns, making it easier to manage and access.

Importance of databases in software development

- **Data Retrieval and Manipulation**

  Databases offer powerful querying capabilities, allowing users to retrieve specific data based on criteria, perform complex calculations, and manipulate data as needed. This enables efficient data analysis, reporting, and decision-making.

Importance of databases in software development

- **Data Integrity and Consistency**

  Databases enforce data integrity constraints, such as unique keys, to ensure the accuracy and consistency of stored data. This helps prevent errors, duplication, and inconsistencies within the database.

Importance of databases in software development

- **Concurrency Control**

  In multi-user environments, concurrency control mechanisms ensure that transactions are executed in a controlled manner to maintain data consistency and prevent conflicts.

- **Data Scalability**

  Databases can scale to accommodate increasing storage and processing requirements. They support features like partitioning, replication, and clustering to distribute data across multiple servers and handle high workloads efficiently.

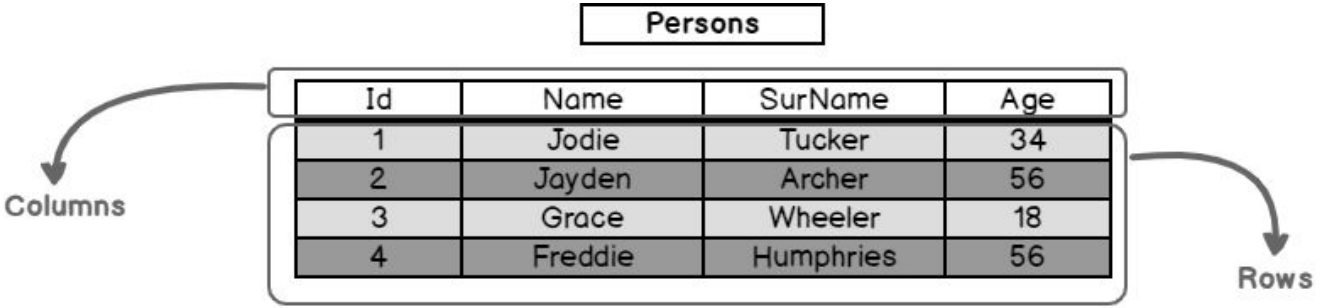
Importance of databases in software development

- **Data Security** (Access Management)
  Databases offer security features to control access to data and protect sensitive information from unauthorised users. Authentication, authorisation, and encryption mechanisms help safeguard data privacy and confidentiality.

- **Data Recovery / Backup** possibilities
  Restore data in the event of hardware failures, software errors, or disasters. Regular backups ensure data resilience and minimise the risk of data loss.

Database : Tables, Rows, Columns

- Databases are divided into **tables** which can be **linked**
- Each table is made of **columns** & **rows**



Primary & Foreign Keys

- Important columns that need to be considered while designing a database.
- **Primary Key** = Column(s) that uniquely identifies a record in a table.
- **Foreign Key** = Column(s) in one table that refers to the primary key in another table.

RDBMS for this course

- We will be using PotsgreSQL as a **Relational Database Management System** (RDBMS)

  SQL = Structured Query Language

SQL

Retrieving data

SELECT statement

SELECT <column names> **FROM** <table_name>**;**

Example :

SELECT first_name, last_name FROM actor;

Hint : You can rename columns with “AS” keyword.

Filtering data

SELECT <column names> **FROM** <table_name>
WHERE <condition>**;**

Example :

SELECT first_name, last_name FROM actor
WHERE first_name = ‘Julia’;

Sorting data

SELECT <column names> **FROM** <table_name> **WHERE** <condition> ORDER BY <column_names> **ASC/DESC;**

Example :

SELECT first_name, last_name FROM actor WHERE first_name = ‘Julia’ ORDER BY last_name ASC;

Aggregating data

SELECT <column names> **FROM** <table_name> **WHERE** <condition> **GROUP BY** <column_name>**;**

Example :

SELECT first_name, **COUNT**(first_name) **FROM** actor **GROUP BY** first_name **ORDER BY** **COUNT**(first_name) **DESC**;

Now YOUR TURN !

Let’s install PostgreSQL and pgAdmin

PostgreSQL, pgAdmin4 and dvdrental sample database

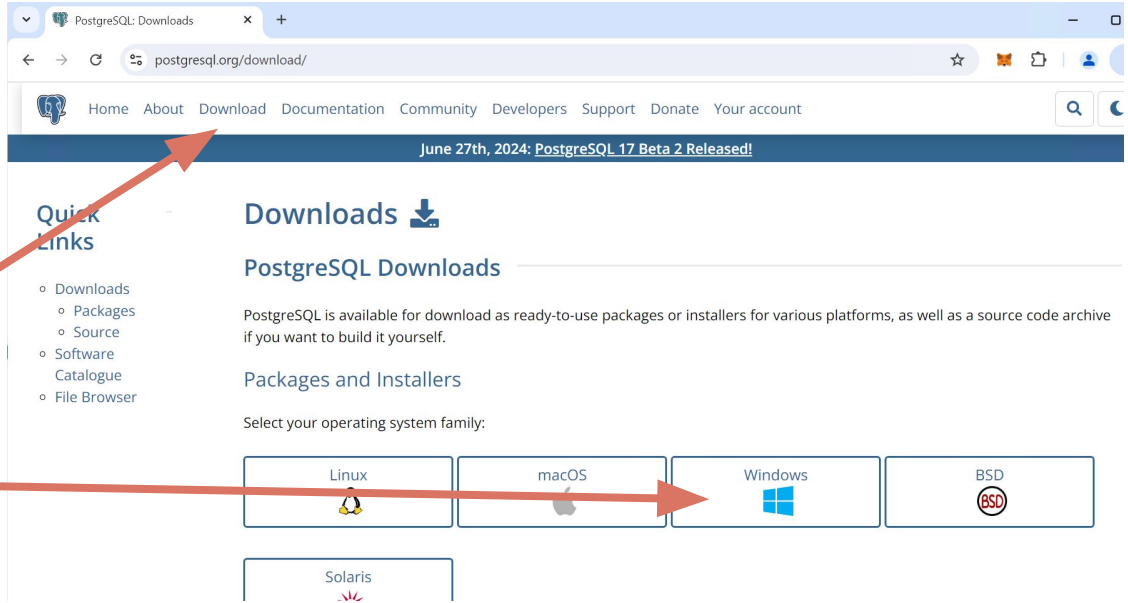
PostgreSQL and pgAdmin4 might be already installed in the virtual machine. If that is the case, skip the following installation steps.

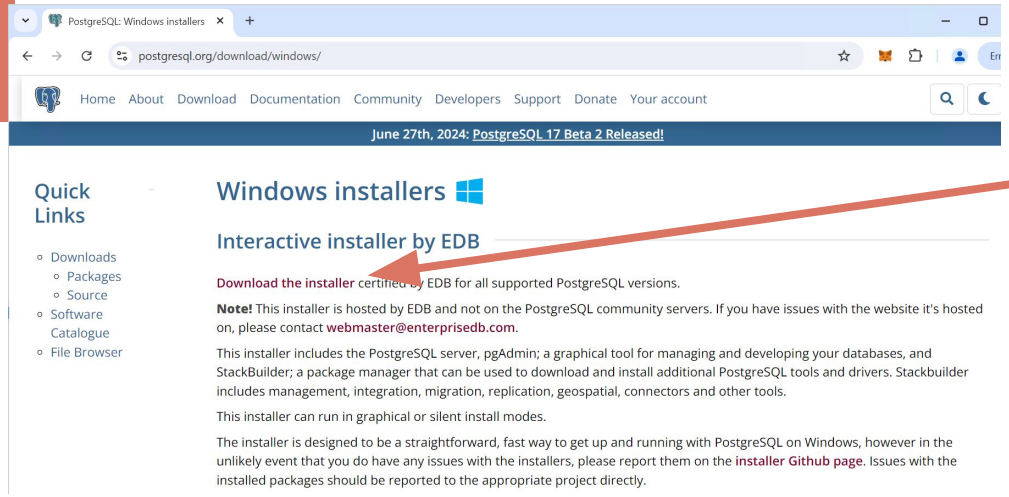
The password will be **admin**

Install PostgreSQL

Open the webpage https://www.postgresql.org

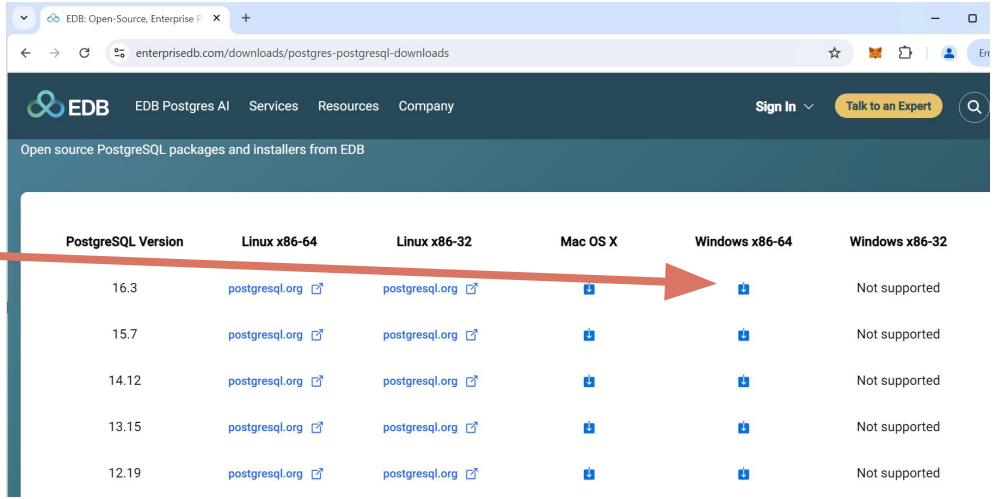
Click on “Download”, then click on “Windows”





Click on “Download the installer”

Then select the newest version for Windows x86-64



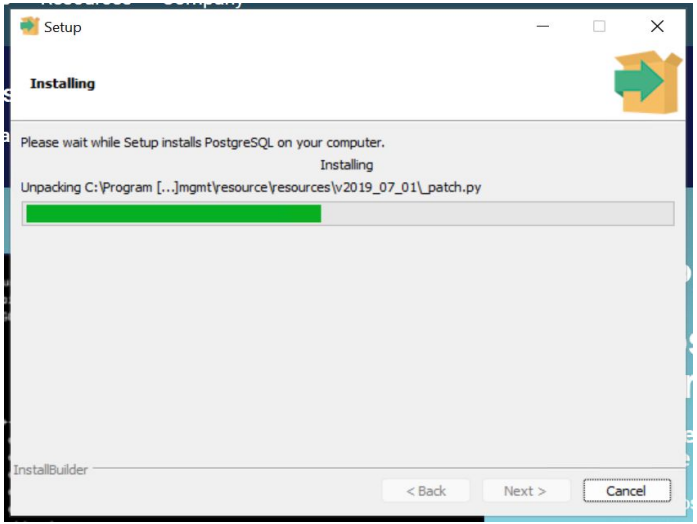
Execute the downloaded .exe file.

Go through the steps of the install wizard.

Make sure to install all components (PostgreSQL Server and pgAdmin 4)

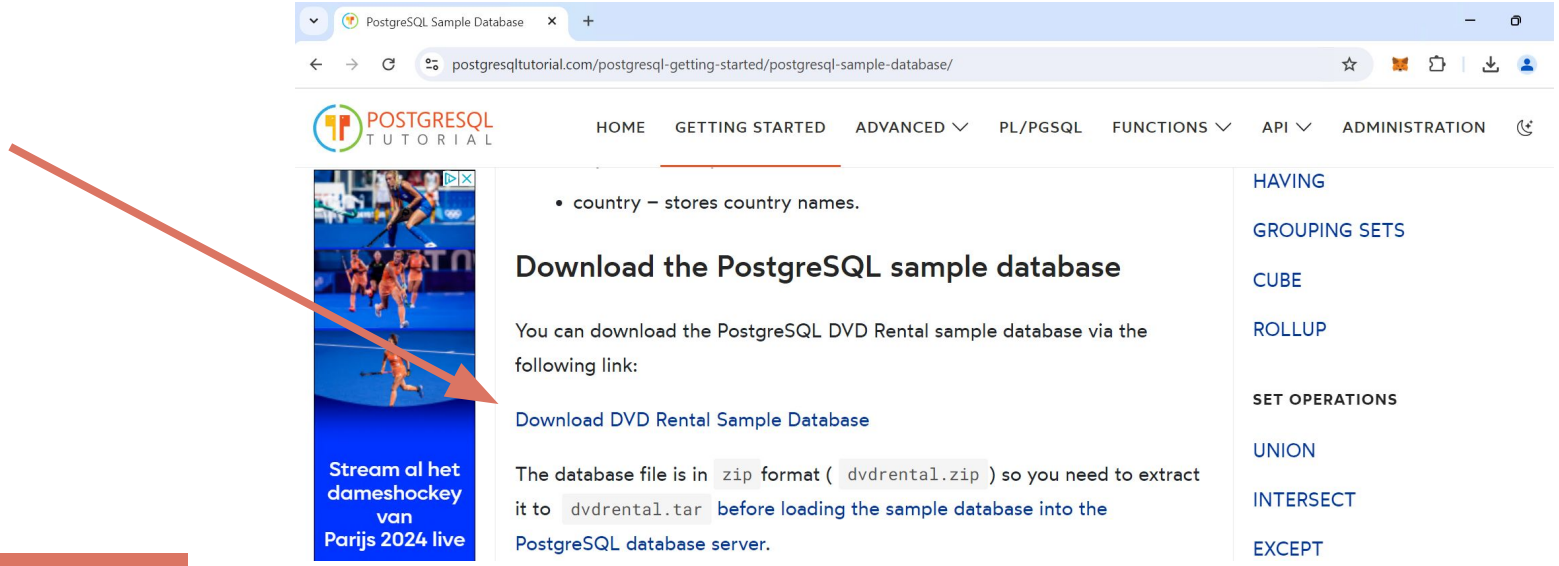
Provide a password for the superuser (postgres) and remember it!

Click on install



Open the webpage:
https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/

Search for the link to download the sample database, it will download a file called dvdrental.zip

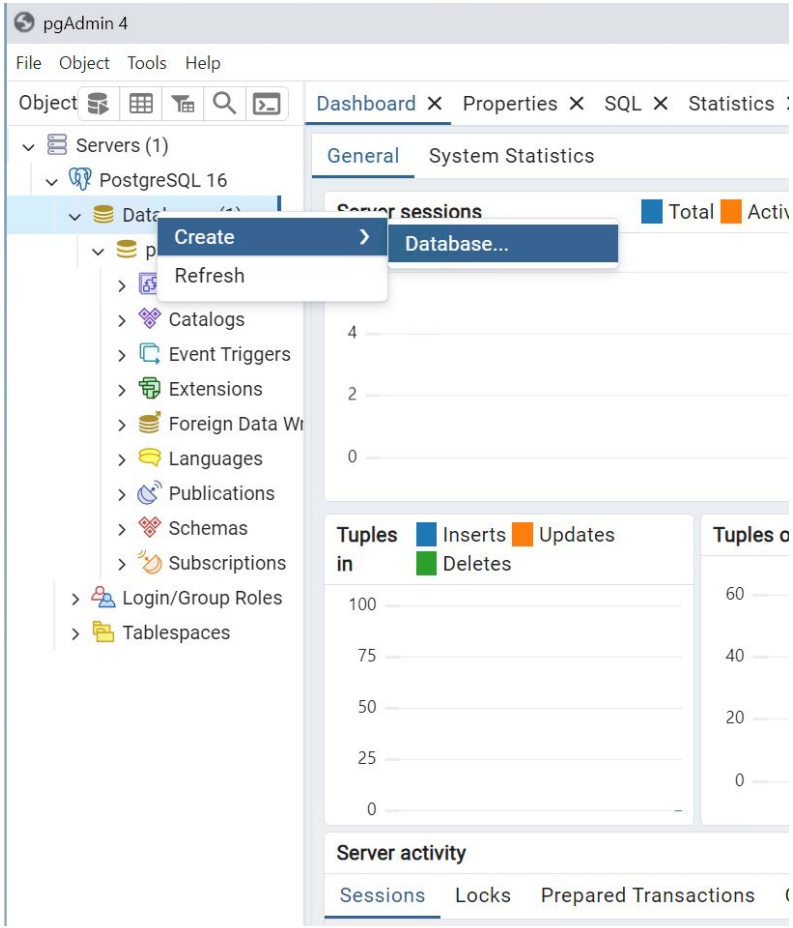


Extract the file dvdrental.zip, it will contain a file called dvdrental.tar

Open pgAdmin 4

Open “Servers”. It will request for the superuser password you created.

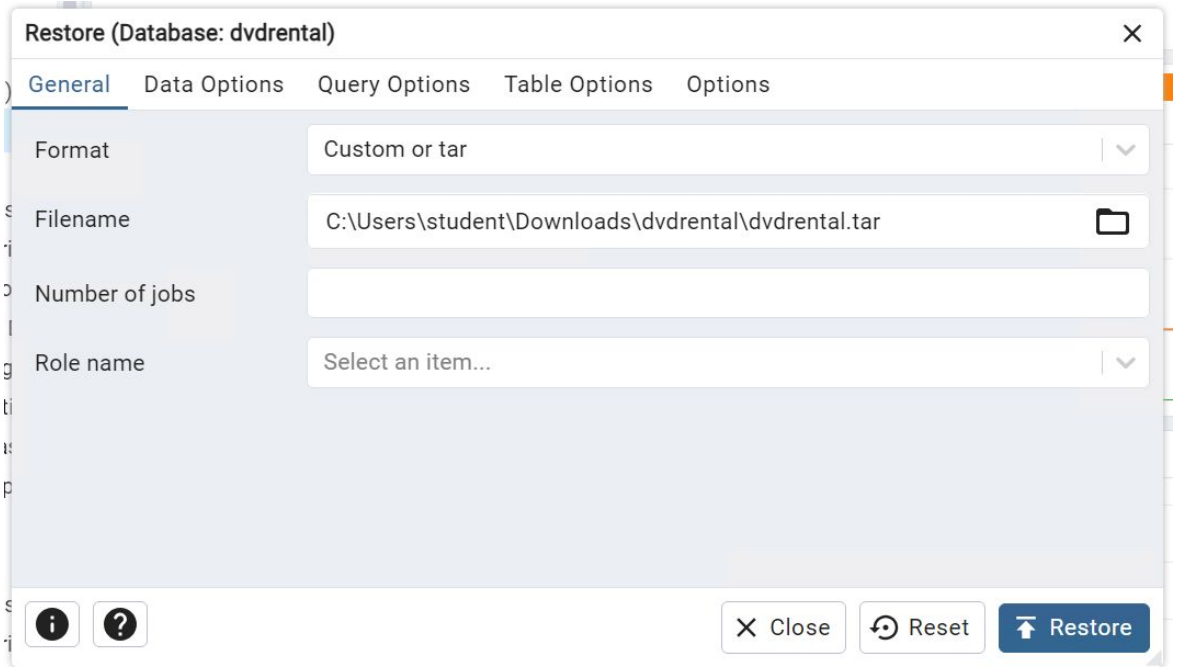
Right click on “Databases” and click on “Create” then “Database”. Name the database dvdrental.



Right click on the new database “dvdrental” and click on “Restore”.

In “Filename”, select dvdrental.tar file

Click on “Restore”



Now YOUR TURN !

Open pgAdmin and let’s do exercises 1

Java connection

JDBC Driver

PostgreSQL driver

- **Connect** to PostgreSQL database with **JDBC Driver**
- Execute **SQL queries**

JDBC Driver

JDBC Driver

```
import java.sql.*;

public class DatabaseManager {
  private String url = "jdbc:postgresql://localhost:5432/event_manager";
  private String user = "postgres";
  private String password = "admin";

  public Connection connect() {
    try {
      return DriverManager.getConnection(url, user, password);
    } catch (SQLException e) {
      System.out.println("Connection failure.");
      e.printStackTrace();
    return null;
    }
  }
}
```

JDBC Driver

```
// Example query
String query = "SELECT name, description FROM event LIMIT 10;";

// Try-with-resources statement to ensure that resources are closed
try (Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(query)) {

    while (rs.next()) {
        String title = rs.getString("name");
        String description = rs.getString("description");
        System.out.println(title + " - " + description);
    }
} catch (SQLException e) {
    System.out.println("Query execution failed: " + e.getMessage());
}
```

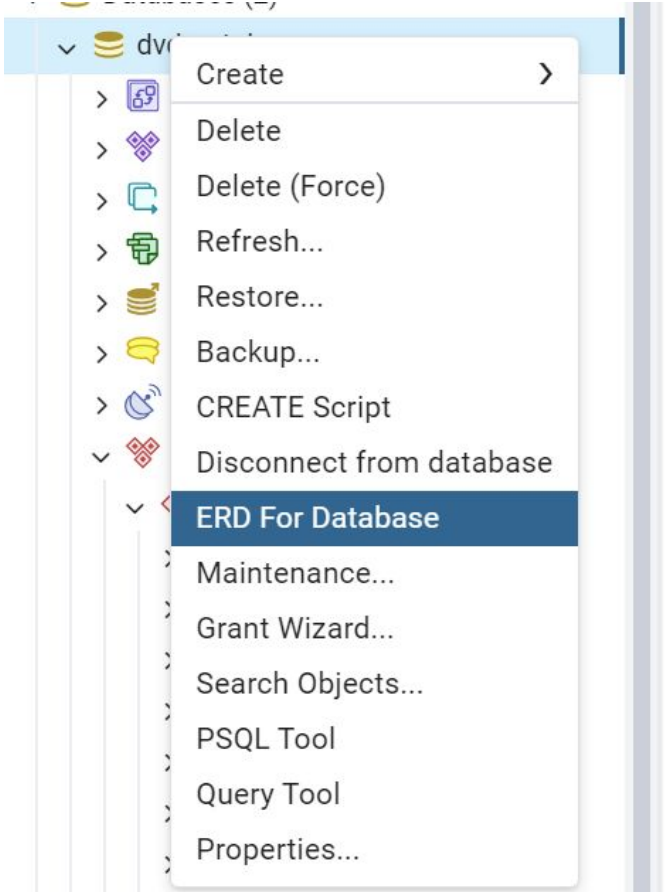
Now YOUR TURN !

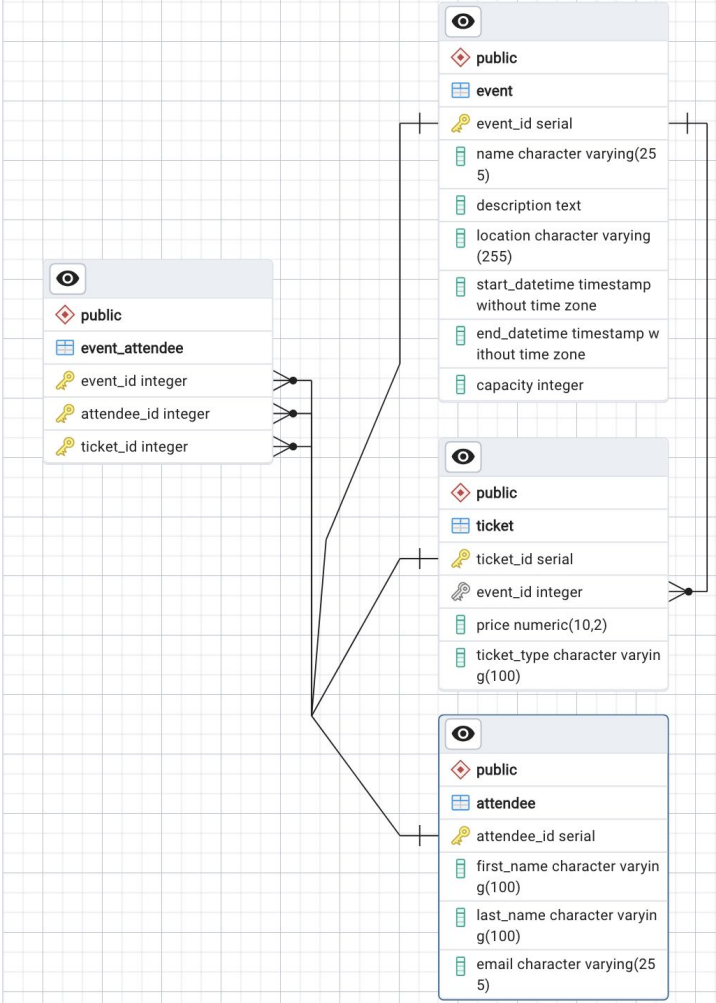
Let’s do exercises 2

ERD Schemas

- **Entity-Relationship Diagram** is a visual representation of the tables, columns, and relationships between tables in a database.

ERD Schemas


Create
Delete
Delete (Force)
Refresh...
Restore...
Backup...
CREATE Script
Disconnect from database
ERD For Database
Maintenance...
Grant Wizard...
Search Objects...
PSQL Tool
Query Tool
Properties...


public
event
event_id serial
name character varying(255)
description text
location character varying(255)
start_datetime timestamp without time zone
end_datetime timestamp without time zone
capacity integer
public
event_attendee
event_id integer
attendee_id integer
ticket_id integer
public
ticket
ticket_id serial
event_id integer
price numeric(10,2)
ticket_type character varying(100)
public
attendee
attendee_id serial
first_name character varying(100)
last_name character varying(100)
email character varying(255)

Event Manager Project - Step 1

Event Management System helps organise and run events like conferences and parties. The program should manage:

- **Types of events**: Includes things like meetings, learning sessions, and fun gatherings.
- **Set up events**: Users can make new events, change them, or remove them.
- **Sell tickets**: Helps with selling tickets, setting prices, and keeping track of how many are sold.
- **Keep track of guests**: Saves information about the people coming to the events.

Event Manager Project - Step 1

- Plan the database schema for the event management system. Make a drawing similar to the one provided in the sample database.
- For each event we should store its name, description, location, start and end times, and capacity.
- For tickets of each event, we should store their price and type.
- For the people attending the events, we should store their names and email addresses.
- Take into account that the same person might attend multiple events and we also need to know which ticket type each one bought.