EUROPEAN UNIVERSITY OF LEFKE

Faculty of Engineering
Department of Software Engineering

COMP337

DATABASE MANAGEMENT SYSTEMS

Lab Work No. 1

Prepared by Abdelrahman Mohamed Radwan Mostafa

Student Number : 21140036

Submitted to Dr. Ferhun Yorgancıoğlu

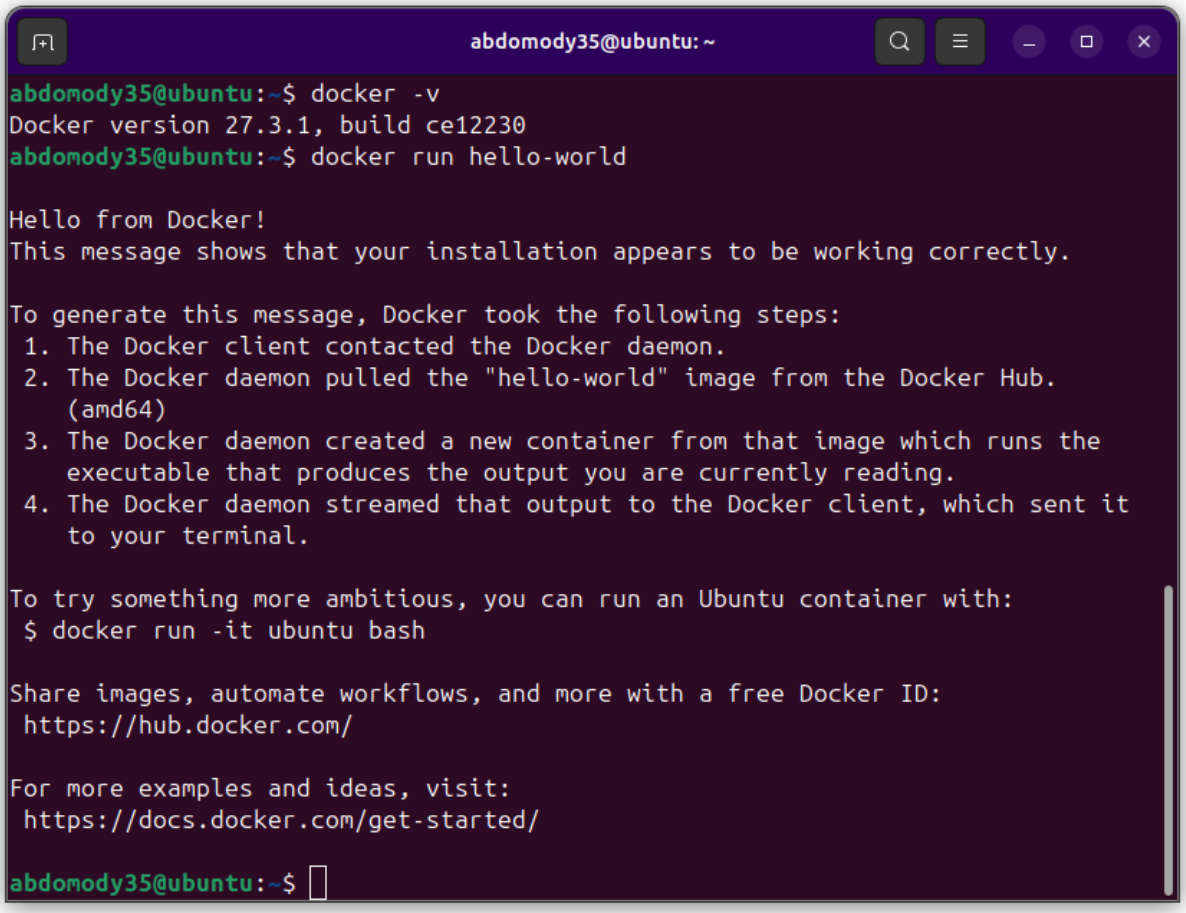
And his assistant Mr. Salman Khan

I have already used multiple RDBMS like MySQL, Microsoft SQL Server, and sql lite so during this course I want to try something new.

I chose PostgreSQL because it's widely used in the tech industry, so it's a great skill to have for real-world projects. It's also open-source, which means it's constantly improving thanks to its active community. I like how reliable and flexible it is, especially for handling complex data.

I will set it up using docker (an industry standard containerizing software). That way it can run on any device regardless of the hardware differences.

The first step would be to install docker. Since I already have docker installed, I will run some commands to ensure that it runs with no issues.



As we can see in the screenshot above, docker is installed and works correctly.

Secondly, we are going to write a yaml file for and use docker compose to run the database server.

```yaml
! db.yml
services:
  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_PASSWORD: password
      POSTGRES_USER: admin
      POSTGRES_DB: university
    volumes:
      - pgdata:/var/lib/postgresql/data
    ports:
      - "5433:5432"

volumes:
  pgdata:
```

The screenshot of the code above includes the configuration for creating the database server specifying the user, password and default database to be used.

Next we are going to execute a command that will use docker compose to create the database server and run it then expose it to the localhost server on port 5433.

The command is :

```
docker compose -f db.yml up
```

The result of such command is going to be very huge so here is the part that we need:

```
✔ Network comp337_default   Created
✔ Volume "comp337_pgdata"   Created
✔ Container comp337-db-1    Created
```

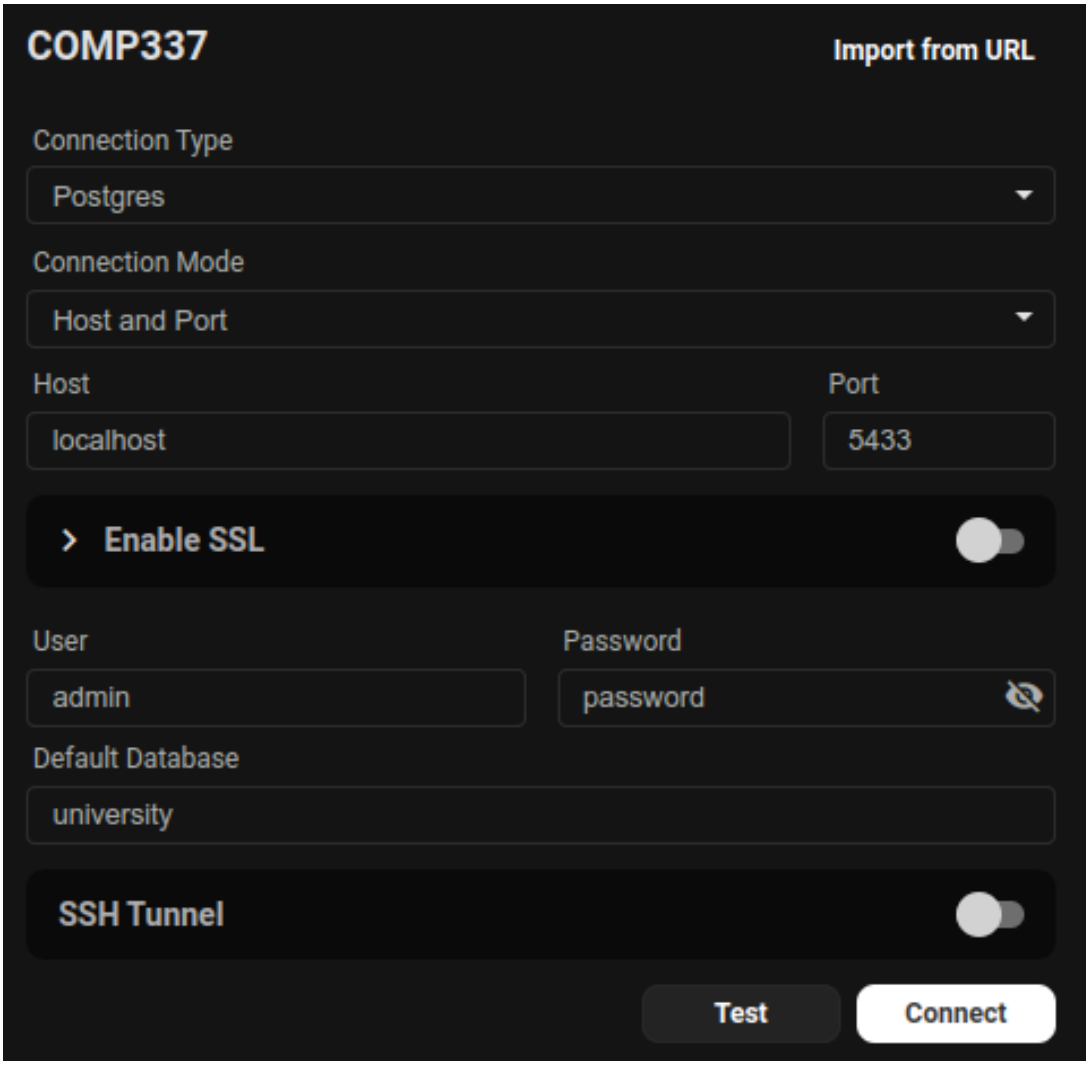
```
db-1  | 2024-10-21 10:17:15.031 UTC [1] LOG:  database system is ready to accept connections
```

And here is the full output:

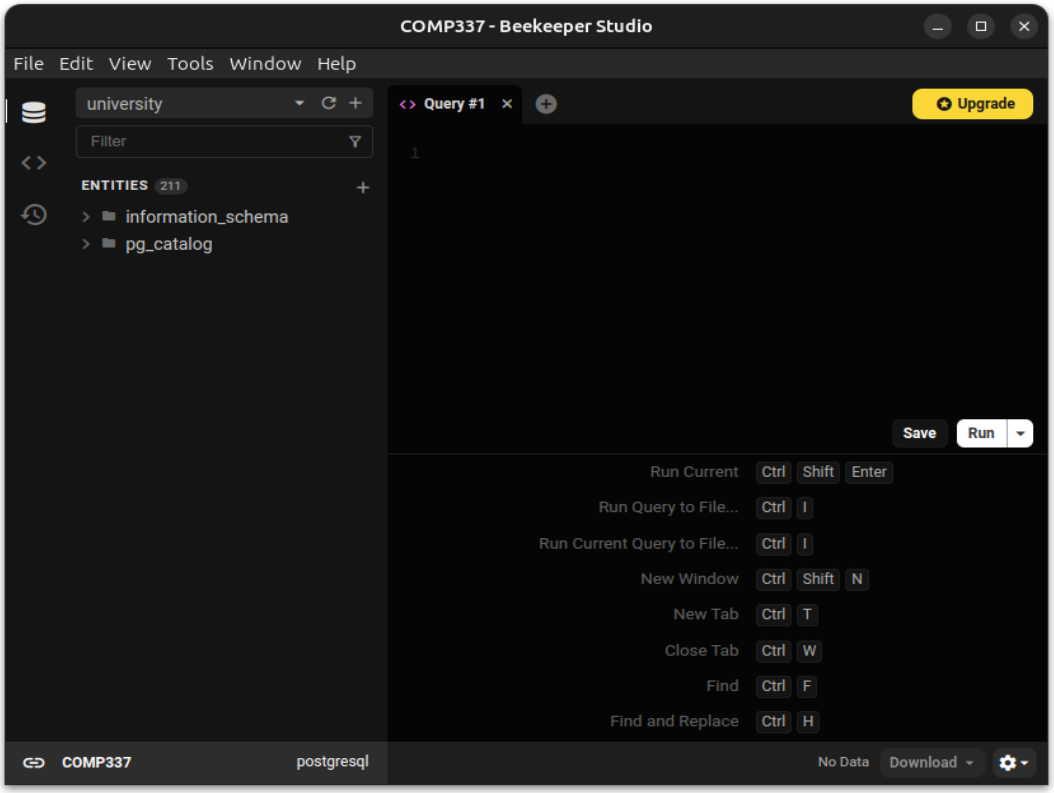
```
 ✔ Network comp337_default  Created                                                                                    0.1s
 ✔ Volume "comp337_pgdata"  Created                                                                                    0.0s
 ✔ Container comp337-db-1   Created                                                                                    0.0s
Attaching to db-1
db-1  | The files belonging to this database system will be owned by user "postgres".
db-1  | This user must also own the server process.
db-1  |
db-1  | The database cluster will be initialized with locale "en_US.utf8".
db-1  | The default database encoding has accordingly been set to "UTF8".
db-1  | The default text search configuration will be set to "english".
db-1  |
db-1  | Data page checksums are disabled.
db-1  |
db-1  | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db-1  | creating subdirectories ... ok
db-1  | selecting dynamic shared memory implementation ... posix
db-1  | selecting default "max_connections" ... 100
db-1  | selecting default "shared_buffers" ... 128MB
db-1  | selecting default time zone ... Etc/UTC
db-1  | creating configuration files ... ok
db-1  | running bootstrap script ... ok
db-1  | performing post-bootstrap initialization ... ok
db-1  | initdb: warning: enabling "trust" authentication for local connections
db-1  | initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.
db-1  | syncing data to disk ... ok
db-1  |
db-1  |
db-1  | Success. You can now start the database server using:
db-1  |
db-1  |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db-1  |
db-1  | waiting for server to start....2024-10-21 10:17:14.658 UTC [48] LOG:  starting PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db-1  | 2024-10-21 10:17:14.658 UTC [48] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db-1  | 2024-10-21 10:17:14.662 UTC [51] LOG:  database system was shut down at 2024-10-21 10:17:14 UTC
db-1  | 2024-10-21 10:17:14.666 UTC [48] LOG:  database system is ready to accept connections
db-1  | done
db-1  | server started
db-1  | CREATE DATABASE
db-1  |
db-1  |
db-1  | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
db-1  |
db-1  | waiting for server to shut down...2024-10-21 10:17:14.884 UTC [48] LOG:  received fast shutdown request
db-1  | .2024-10-21 10:17:14.885 UTC [48] LOG:  aborting any active transactions
db-1  | 2024-10-21 10:17:14.886 UTC [48] LOG:  background worker "logical replication launcher" (PID 54) exited with exit code 1
db-1  | 2024-10-21 10:17:14.886 UTC [49] LOG:  shutting down
db-1  | 2024-10-21 10:17:14.887 UTC [49] LOG:  checkpoint starting: shutdown immediate
db-1  | 2024-10-21 10:17:14.920 UTC [49] LOG:  checkpoint complete: wrote 921 buffers (5.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.012 s, sync=0.020 s, total=0.035 s; sync files=301, longest=0.001 s, average=0.001 s; c
istance=4238 kB, estimate=4238 kB; lsn=0/1908978, redo lsn=0/1908978
db-1  | 2024-10-21 10:17:14.925 UTC [48] LOG:  database system is shut down
db-1  | done
db-1  | server stopped
db-1  |
db-1  | PostgreSQL init process complete; ready for start up.
db-1  |
db-1  | 2024-10-21 10:17:15.023 UTC [1] LOG:  starting PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db-1  | 2024-10-21 10:17:15.023 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db-1  | 2024-10-21 10:17:15.023 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db-1  | 2024-10-21 10:17:15.025 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db-1  | 2024-10-21 10:17:15.027 UTC [64] LOG:  database system was shut down at 2024-10-21 10:17:14 UTC
db-1  | 2024-10-21 10:17:15.031 UTC [1] LOG:  database system is ready to accept connections
```

Now that we know that the database is running, Let's try to connect to it. I will be using an app called beekeeper that allows you to connect to a database server.

Here is a screenshot of the connection configuration:



After we have connected, it will take us to a page where we can select the databases and run queries on them as seen in the screenshot below:



To ensure that the connection has been established let's run a query.