Databases

Installing a DBMS (PostgreSQL)

Marco Vieira

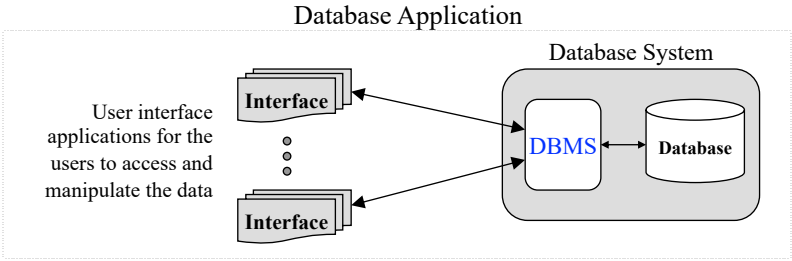
Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022

Outline

- Database Management Systems (DBMSs)
- Installation of PostgreSQL
- Install *psql* in the localhost
- Setup the database for the exercises in the following lessons
- Install *pgadmin*

Database Management System (DBMS)

- A Database Management System (DBMS) is software package to define, manipulate, retrieve and manage data in a database



Some Alternatives Available

- Commercial

ORACLE DATABASE, Microsoft SQL Server, IBM DB2, SAP MaxDB The SAP Database

- Open source

PostgreSQL, MySQL, SQLite, monetdb, cassandra

Lightweight (not a server)

NoSQL column-oriented

Oracle Database

- Commonly referred to as Oracle DBMS or simply as Oracle
  - Oracle Corporation
  - First version in 1979, Oracle v2
  - Current version: Oracle Database 19c
- Multi-model database management system
- Commonly used for running Online Transaction Processing (OLTP), Data Warehousing (DW) and mixed (OLTP & DW) workloads
- Available by several service providers on-prem, on-cloud, or as hybrid cloud installation
- One of the the most complete and most used commercial DBMSs
  - It has been a market driver for many years!

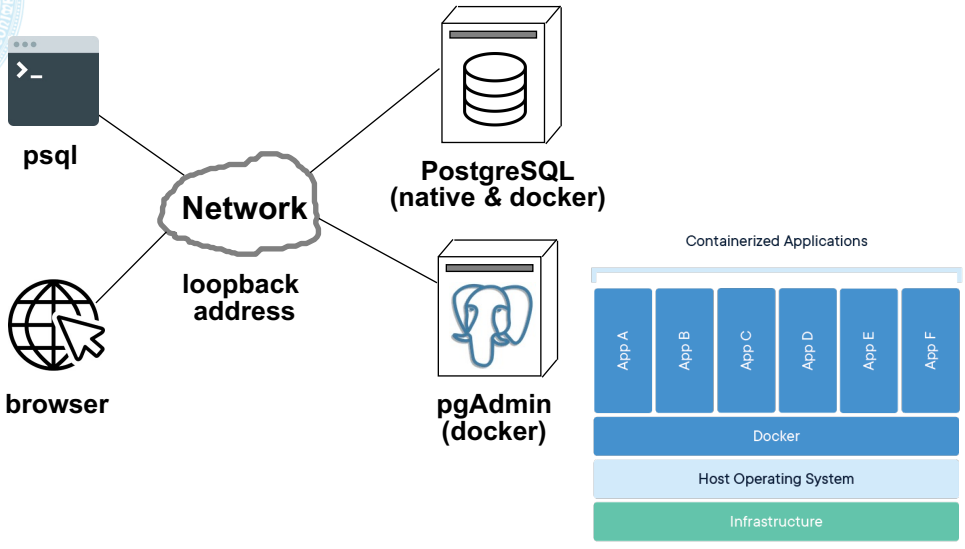
PostgreSQL

- Free and open-source, commonly referred to as Postgres
  - University of California, Berkeley
  - First version in 1997, release 6.0 (successor to the Ingres database)
  - Current version: 13
- Relational database management system (RDBMS), emphasizing extensibility and SQL compliance
- Default database for macOS Server, and is available for Linux, FreeBSD, OpenBSD, and Windows
- Many organizations use PostgreSQL as the primary database, e.g.:
  - Instagram, TripAdvisor, WhitePages.com, MusicBrainz, BASF, Sony Online

What are we going to do next?

- Two different installations of PostgreSQL (both in local machine):
  - Native installation
  - Using Docker
- Install *psql* in the localhost (without a full installation of PostgreSQL)
- Setup the database for the exercises in the following lessons
  - Create user, database and schema
  - Create tables
- Install *pgadmin* using Docker
- Note: we are assuming Mac OS / Linux, but installation concepts will be similar for other operating systems

The Outcome Will Be…



Native Installation of PostgreSQL

- Download the certified installer for the latest version from:
  - https://www.postgresql.org/download/
- Run the installer
  - Select what you want to install
    - *pgadmin* can be installed together with the DBMS
    - *pqsl* can also be installed (select Command Line Tools)
  - Set the directory where the database files will be stored
  - Define the password (*postgres*) for the *postgres* user (default superuser)
  - Select the port number to be used to connect to the server (default is 5432)
  - Note: **remove** Stack Builder Check option **after** installation
- Access the database via *psql*
  - **Windows**: add C:\Program Files\PostgreSQL\13\bin to PATH environment variable

```
psql -d postgres -U postgres
```


Install PostgreSQL using Docker

- Install *Docker Desktop* (if not yet installed):
  - https://www.docker.com/products/docker-desktop (may require restarting)
- Take a look at: https://hub.docker.com/_/postgres
- Pull *postgres* image from *dockerhub*:

```
docker pull postgres
```

- Create and start a *postgres* instance:

```
docker run -d -p 5432:5432 --name postgres -e
POSTGRES_PASSWORD=postgres postgres
```

- Open a terminal to the *postgres* instance and try *psql*:

```
docker exec -it postgres bash
psql -d postgres -U postgres
```

Install *psql* in the Localhost

- Note: the native installation of PostgreSQL (as done before), includes the installation of *psql*
- Without a full installation of PostgreSQL:
  - https://blog.timescale.com/tutorials/how-to-install-psql-on-mac-ubuntu-debian-windows/
- Windows
  - Use native installer but select only Command Line Tools
  - Add C:\Program Files\PostgreSQL\13\bin to PATH environment variable
  - Or use "SQL shell" app

Install *psql* in the Localhost

- MacOS/Linux
  - Install the Homebrew Package Manager (if not yet installed):

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

  - Update brew, and install *libpq*, from the command line:

```
brew doctor
brew update
brew install libpq
```

  - Symlink *psql* (and other libpq tools) into */usr/local/bin*:

```
brew link --force libpq
```

- Connect to the PostgreSQL instance running on Docker:

```
psql -h localhost -p 5432 -d postgres -U postgres -W
```

  - This works on *localhost* because we have forwarded port 5432 to the container

Install *pgadmin* using Docker

- Note: the native installation of PostgreSQL already includes *pgadmin*
- Take a look at: https://hub.docker.com/r/dpage/pgadmin4/
- Pull *pgadmin4* image from *dockerhub*:

```
docker pull dpage/pgadmin4
```

- Create and start a *pgadmin* instance:

```
docker run -d -p 80:80 --name pgadmin \
       -e PGADMIN_DEFAULT_EMAIL=someone@dei.uc.pt \
       -e PGADMIN_DEFAULT_PASSWORD=pass12345 dpage/pgadmin4
```

- Access *pgadmin* using a browser:

```
http://127.0.0.1
```

  - Username: someone@dei.uc.pt ; Password: pass12345

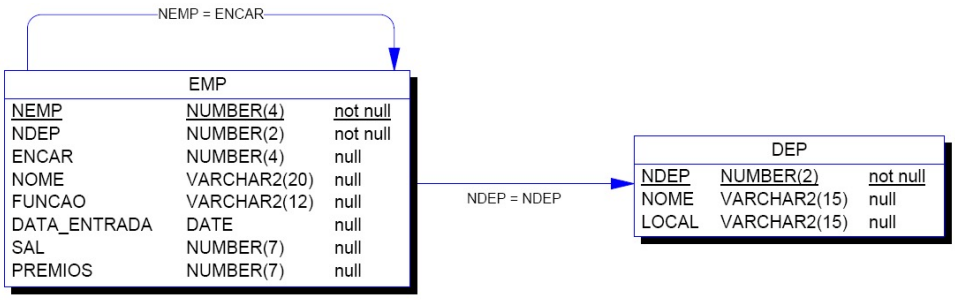
Install *pgadmin* using Docker

- Add a server in *pgadmin* GUI to connect to postgres container:
  - Either use the container IP directly:

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
postgres
```

  - Or use the special DNS name *host.docker.internal* (will connect to localhost, which will forward the request to the container)
  - Username: postgres ; Password: postgres

Database to be used in the SQL Lessons



Setup the Database

- Create *dbfichas* database:

```
psql -h localhost -p 5432 -d postgres -U postgres
create database dbfichas;
```

- Create *aulaspl* user:

```
create user aulaspl password 'aulaspl';
exit
```

- Create the tables (code in file *criaTabelas.sql*):

```
psql -h localhost -p 5432 -d dbfichas -U aulaspl
\i criaTabelas.sql
```

- Fill the tables (code in file *insereDados.sql*):

```
\i insereDados.sql
select * from emp;
```

Practical Exercises

Solve the practical exercises made available for this class

Q&A

Databases

Installing a DBMS (PostgreSQL)

Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022