Intro to Databases (1)

what are databases?

- when you write programs you typically want some sort of data to be generated, if we dont save this generated data then it is lost

SQL Databases provide a sensible choice of where to save your data.

SQL provides:

- Highly optimiseed storage of **tabular** data
  - data presented in columns or tables
- SQL has a fast, well understood, query langauge
- there are fault tolerant protocols

While you could write it simply to a file, databases come with a host of advantages meaning they are the better option for data storage.

So what is a database?

- basically a super fancy spreadsheet
- each database contains **tables** which store data
- data in tables can be **queried** using using a language called **SQL**
- you can **join** data in one table with data in other tables to answer more questions
- designing them is tricky
  - olden days used to have data engineers

Database Variants

Traditionally, the database would reside on a seperate machine

- storage restrictions - space used to be expensive
- you used to have to connect to it if you wanted to access it

Now, space is cheap

- local app databases are very common

if you need remote database access:

- **server-style** database e,g, **MariaDB or MySQL**

Else you can use a file-style database:

- **SQlite**
  - database is just a file on your computer
  - very popular

Using SQlite

- install the required packages on however your operating system does it
  - or connect to SQlite via programming langauge

MySQL and MariaDB

Sometimes you need a server style database

- Joe recommends MariaDB

MariaDB History

- used to be called MySQL
  - named after the developers kid (My) and the lanaguge used to query it (SQL)
- it was brought by Oracle, however many developers dont like Oracle

- the original devleoper forked the open source one to make MariaDB (named after his other kid)

the command is **mysql** for both

Using MariaDB

- you need to first start the database server and tell it where to connect to
- on most LINUX machines it will be via **SystemD:**

```
systemctl start mariadb
systemctl enable mariadb
```

On Alpine Linux itll be via **OpenRC:**

```
rc-service mariadb start
```

Security

Once MariaDB is up and running itll have some tets databases and a **root** user with no password, **it is up to you to secure your database server!**

You can automate most of it with this command:

```
mysql_secure_installation
```

However if someone is paying you to do it:

- manually set usernames and passwords
  - dont use root user for everything, have different users with different levels or permissiosn
- firewall off ports
  - so you can only connect to the server on devices you are expecting etc
- add intrusion detection

- Backup the database
- Secure those backups
    - it is often times easier to steal backups as they are less protected that the current database

Conlclusion/when to use a database

Does the data need to be assessed remotely?

- yes → use server style database (MySQL/MariaDB)
- no → use a file-style database (SQLite)

There are situations where SQL databases arent the best option:

- when data contains recursive data structures
    - SQL isnt a Turing Complete langauge, therefore there are calculations it is unable to perform
    - use **Prolog or Datalog instead**