

DATABASES

Databases give you **persistance** of records

- Survive server
restart

Databases allow you to

- store data
- retrieve data
- search for
data

A file can be a database

TYPES

Many basic KINDS of database.

Two big concepts (but others exist too)

- Relational
Databases
- Document
Databases

commonly: SQL vs NoSQL

SQL

- S - Structured
- Q - Query
- L - Language

SQL isn't a database, it is a language to send/search/receive data.

Commonly supported by SEVERAL database vendors (mostly the same)

example:

```
SELECT name, age FROM people;  
SELECT age FROM people WHERE name='Amit';  
UPDATE people SET age = 28 WHERE name='Bao';
```

Assumes **RELATIONAL** data

RELATIONAL

"Relational" means data is "related"

Think of spreadsheets:

- rows = records
- columns = different data values

All records have all columns, even if they aren't filled in

NORMALIZATION

Students in a class:

- Name
- Age
- Home address
- Name of class1 (also room. also time)
- Name of class2 (also room. also time)

Break out into different collections

- Table of students (student id, names, ages, home addresses)
- Table of classes (class id, name, room, time)
- Table of enrollment (student id, class id)

You can pull data from all three - use the enrollment table and pull up the RELATED data from the other tables

NOSQL

NoSQL is a term often used to generalize "document"-based databases.

Not like a spreadsheet - not every record is the same (though they often are)

Still has a query language

- Often looks a lot like JS
- Not very standardized yet

SERVERS

Databases are often run as separate servers

- Use non-web connections
- Often have login systems

Login and client/server isn't required, but is very common

- Ex: Oracle, SQL Server, MySQL, PostgreSQL, MongoDB

Some systems DO have web interfaces to the DB

- these are just wrappers around the real DB

This means most DB interactions will be backend only

- Unless going through a service to talk to the DB
 - Because the service will talk to the DB on **its** backend

SIMPLE DATABASES

SQL: SQLite (<https://sqlite.org/>)

NoSQL: LokiJS (<http://lokijs.org>)

- No separate client/server, just local DB files
- Remarkably fast
- Limited handling of multiple users

ASYNC

Most DB operations are async!

- Not sure how much delay there will be
- So you give a callback (maybe via a promise) for when it is done
- Remember error handling!