Week 1

Database Crash Course

Table of content

- Overview
- About the instructor
- Our Communication Channels
- Install required tools
- The History of Database
- SQL Vs NoSQL
- Why MySQL (MySQL vs PostgreSQL)
- Who Uses MySQL?
- What is Database Design
- Setup MySQL
- MySQL Basic Commands
- Practise SQL
- Advanced Topics (To be Continue in Advanced Course)
    - Database Design
    - MySQL Complex Queries
    - Database indexing
    - Database performance and optimizations
- Questions
- Thank you

About the instructor

My name is ahmed salah . I graduated from FCI Menofia.I have participated in many Programming Competitions (ECPC 2017 , 18 , 19) , Snackdown 2019.from Nov 2017 to Nov 2019 , I was a Coach for ICPC Menofia Community. I have also applied for many internships (Google North America , Microsoft ATL , Samsung , Quora , Databrick ....and Dolby Australia).

Worked as a Java Software engineer for about +4 years . within this period , I worked as a Senior Backend Engineer in 3 multinationals (Mondia , CME and Solutions By STC[Current]).

Feel free to contact me for any question:
Facebook Mail
Linkedin

Install required tools

- jdk8
- Intellij
- VSCode
- Git
- Github
- Xampp
- DBeaver

Our Communication Channels

- Whatsapp Group
- Google Calendar

We will have 2~3 Sessions per week.our weekly sessions will be on Tuesday at 8 PM , Friday 9 AM

The History of Database

- **The 1960s – beginnings**

The history of databases begins with the two earliest computerised examples. Charles Bachman designed the first computerised database in the early 1960s. This first database was known as the Integrated Data Store, or IDS. This was shortly followed by the Information Management System, a database created by IBM.

- **The 1970s – relational databases**
- **The 1980s – growth and standardisation**
  - The 1980s also saw SQL become the standard language used for databases
- **The 1990s – the internet**
- **The 2000s – NoSQL**
  - It refers to databases that use query language other than SQL to store and retrieve data. NoSQL databases are useful for unstructured data, and they saw a growth in the 2000s
- **The 2010s – distributed databases and cybersecurity.**

Source: the history of db

SQL Vs NoSQL

SQL is the programming language used to interface with relational databases. (Relational databases model data as records in rows and tables with logical links between them). NoSQL is a class of DBMs that are non-relational and generally do not use SQL.

There are five practical differences between SQL and NoSQL:

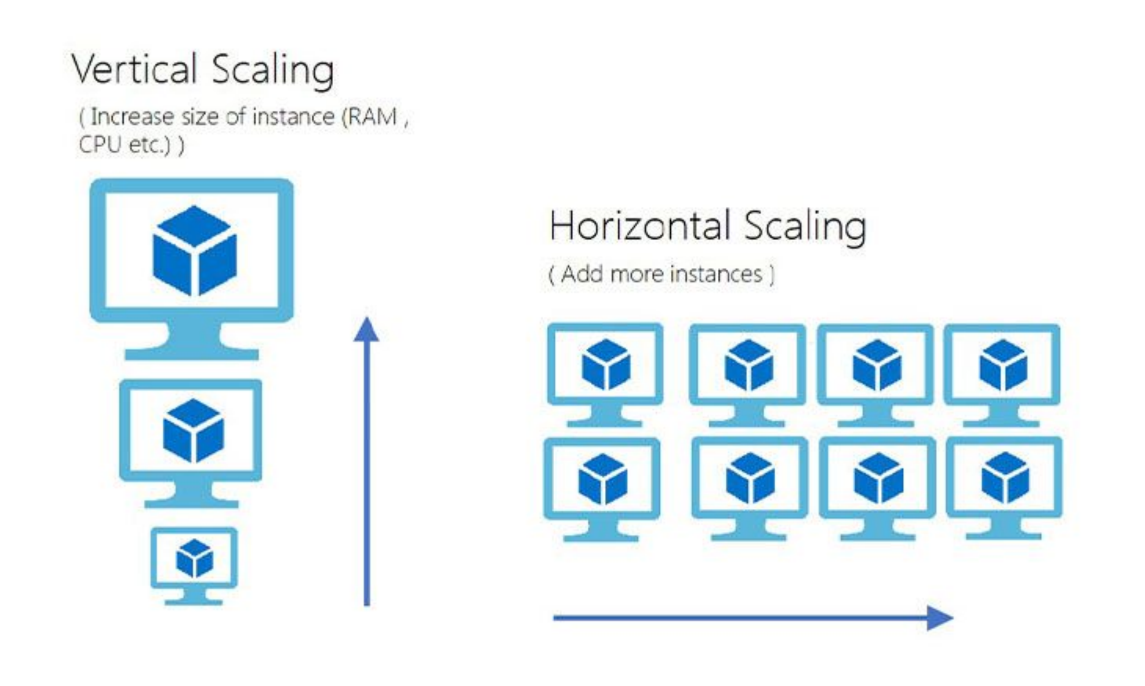
- Language
- Scalability
- Structure
- Properties
- Usage

SQL Vs NoSQL (**Language**)

SQL	NoSQL
SQL has been around for over 40 years, so it is ● recognizable ● documented ● Widely used. ● Safe and versatile it’s particularly well suited for complex queries	The dynamic schemata of NoSQL databases ● allow representation of alternative structures, often alongside each other ● greater flexibility. ● less emphasis on planning ● greater freedom when adding new attributes or fields however, NoSQL languages lack the standard interface which SQL provides, so more complex queries can be difficult to execute.

SQL Vs NoSQL (**Scalability**)

SQL	NoSQL
Vertical Scalability Most SQL increasin hardware	Horizontal Scalability r-slave ers or



SQL Vs NoSQL (**Structure**)

SQL	
SQL database s	s format, but

Document 1

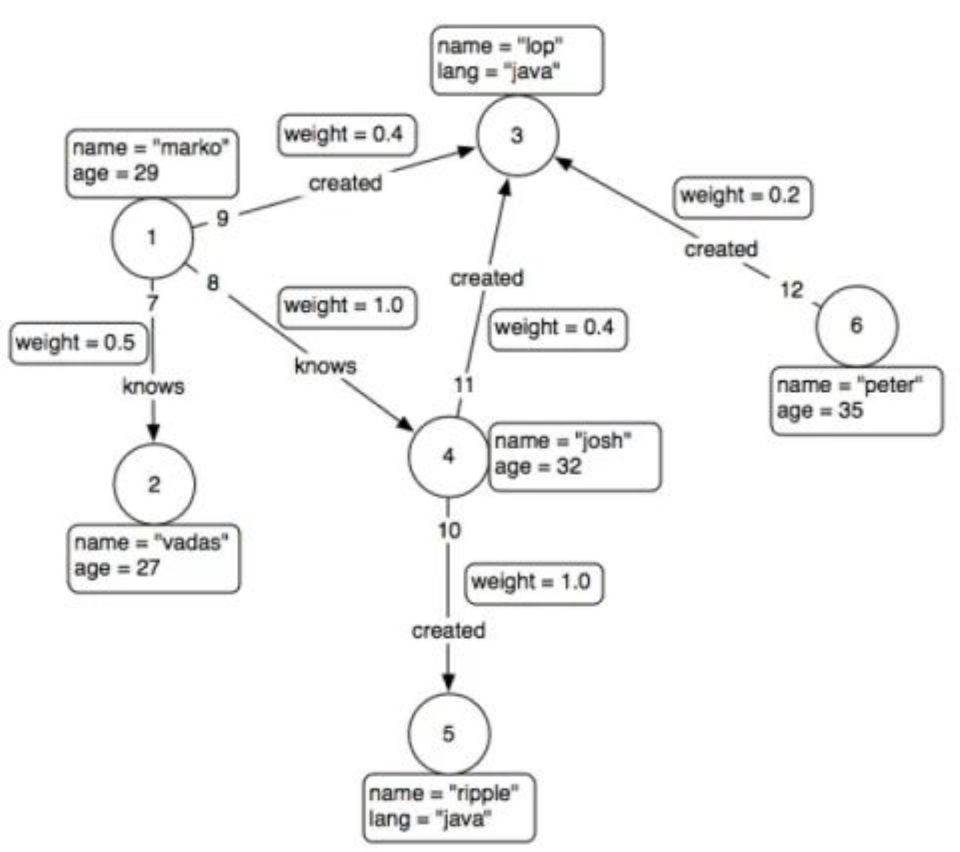
```
{
 "id": "1",
 "name": "Johr
 "isActive": tru
 "dob": "1964-
}
```

t 3

```
":

dam",
ark"

: true,
15-04-19"
```



SQL Vs NoSQL (**Properties**)

At a high level, SQL and NoSQL comply with separate rules for resolving transactions.

SQL	NoSQL
RDBMSs must exhibit four “ACID” properties: ● **Atomicity** ● **Consistency** ● Isolation ● **Durability**	NoSQL technologies adhere to the “CAP” theorem, which says that in any distributed database, only two of the following properties can be guaranteed at once: ● **Consistency**: Every request receives the most recent result, or an error. (Note this is different than in ACID) ● **Availability**: Every request has a non-error result, regardless of how recent that result is. ● **Partition tolerance**: Any delays or losses between nodes will not interrupt the system’s operation.

SQL Properties - Atomicity

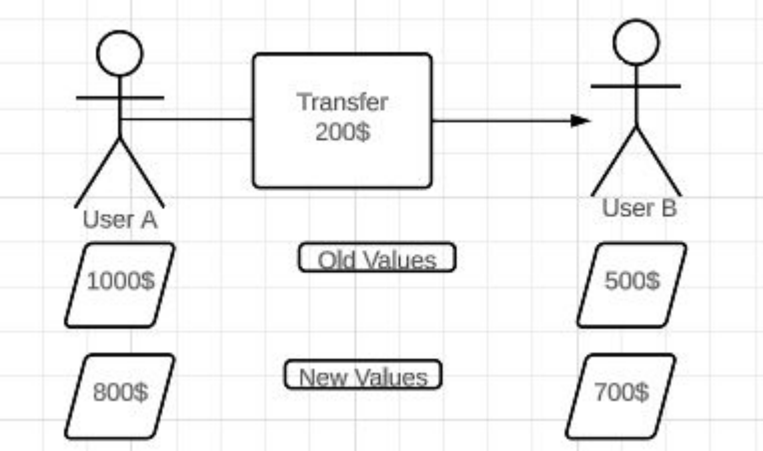
The entire transaction takes place at once or does not happen at all.

- (read , update , delete)
    - If all success -> commit
    - If one fail -> rollback

SQL Properties - Consistency

Consistency Requiren

- The total balanc efore and after the transaction.
- The sum of the b t.
- No money shoul



Consistent Outcome:

- The database sh count A and adding $200 to Account B occur ystem failure, error), one part of the trans led back, and the database should remain in a consistent state.
-

Example Violation of Consistency:

- Without consistency, if the deduction from Account A succeeds, but the addition to Account B fails (e.g., due to a system error), the system could end up in an inconsistent state where money has been deducted but not deposited, leading to incorrect balances.

●

SQL Properties - Isolation

Transactions happens independently

Even though multiple transactions may be executing concurrently, the final result should be as if they were executed in some sequential order.

SQL Properties - Durability

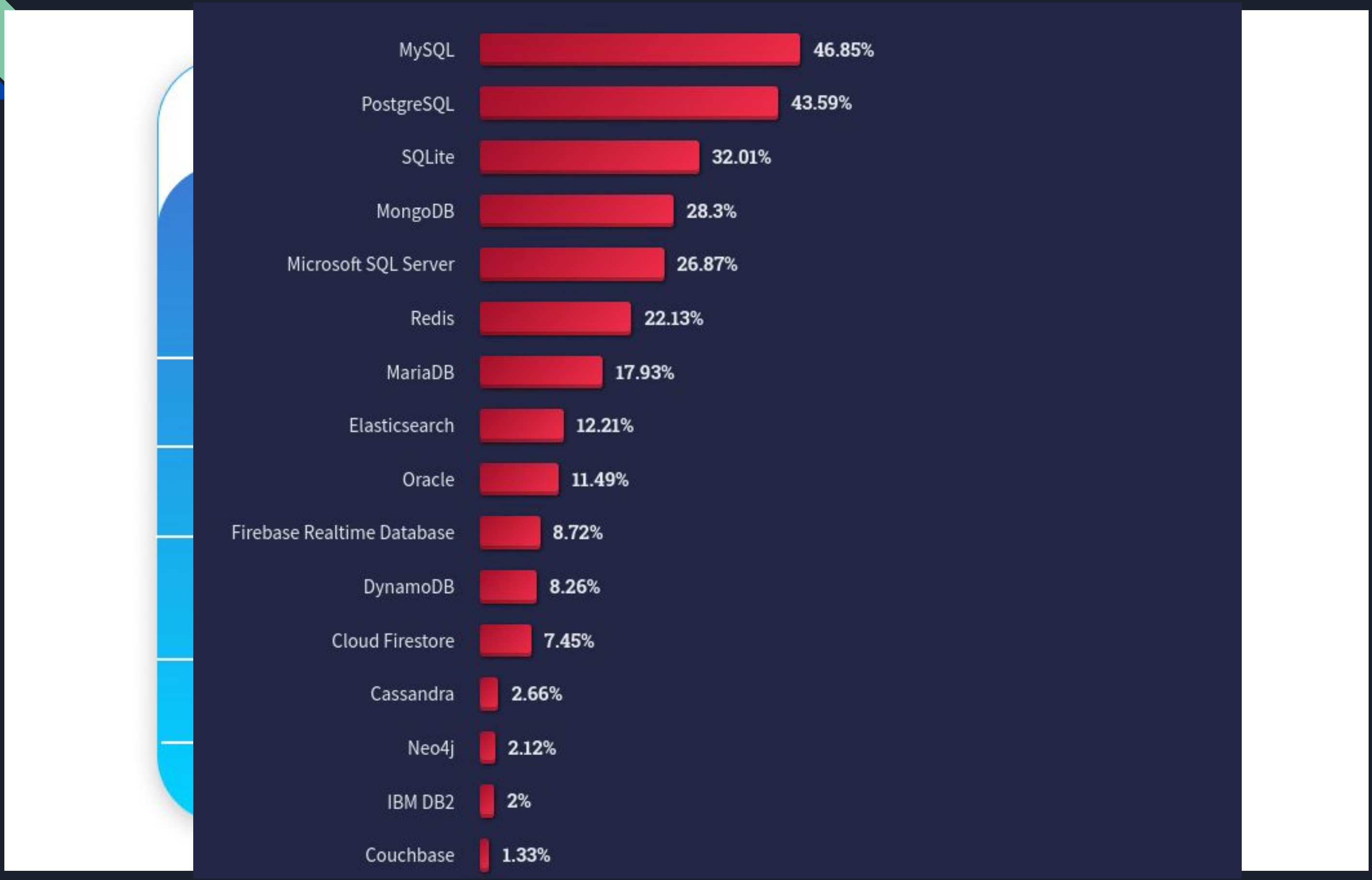
guarantees that once a transaction is committed, its effects are permanent, such as power outages or crashes. The committed changes must be stored in a non-volatile memory (e.g., disk) to ensure durability.

Example: If a user is notified that a transaction is successfully completed, the changes made by that transaction should persist even in the event of a system failure.

SQL Vs NoSQL (**usage**)

SQL	NoSQL
SQL is more appropriate when the data is: ● Small ● Conceptually modeled as tabular ● In systems where consistency is critical. In terms of use cases, this might translate to small business' accounting systems, sales databases, or transactional systems like payment processing in e-commerce. When in doubt, SQL is also more appropriate, as RDBMSs are better supported and fault-tolerant.	Generally, NoSQL is preferred for: ● Graph or hierarchical data ● Data sets which are both large and mutate significantly, ● Businesses growing extremely fast but lacking data schemata. In terms of use cases, this might translate to social networks, online content management, streaming analytics, or mobile applications.

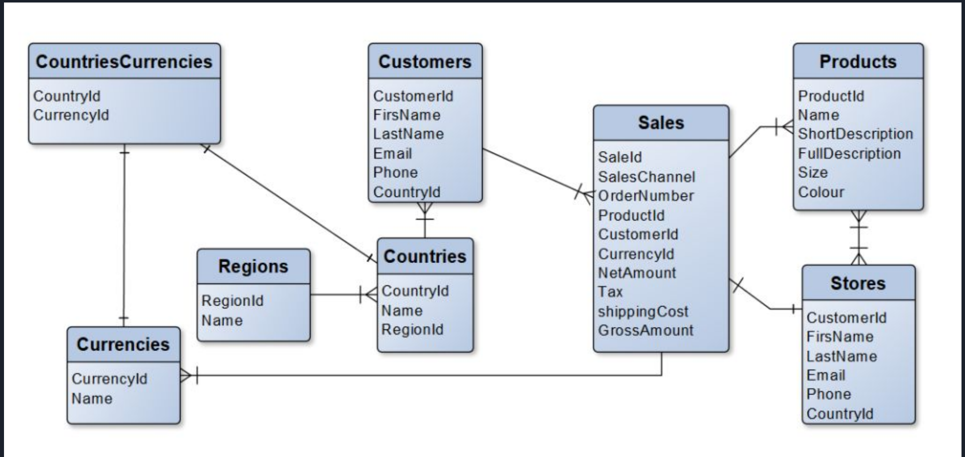
Why MySQL (**MySQL vs PostgreSQL**)



SRC

What is Database Design

Database design is the organisation of data according to a database model. The designer determines what data must be stored and how the data elements interrelate



Installing MySQL

https://www.mysqltutorial.org/install-mysql/

You can also use XAMPP

Install MySQL using Docker

MySQL Basic Commands

https://www.w3schools.com/mySQl/default.asp

Practise SQL

https://www.hackerrank.com/domains/sql?filters%5Bdifficulty%5D%5B%5D=easy

Thanks