CSC 460: SmallData

POD 2 Databases

William Ng, Zakaria Chowdhury, Jawad Kabir, Jawad Chowdhury, Timson Tan, Darren Ling

Introduction

Agenda Overview:

- Quick preview of each module: SQL, NoSQL, Cloud DB, and Docker
- History and evolution of each database type.
- Overview of tabular comparison on features, use cases, and pros/cons.
- Discussion on industry relevance and recommendations.

Goal:

To explore how different databases serve varying needs and discuss hosting solutions for collaboration.

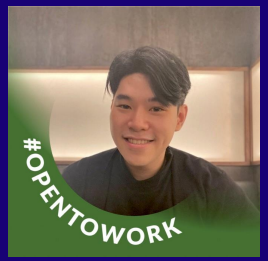
Table of contents

Table of contents

Small Data

This presentation is brought to you by:

Timson T



William N

Zakaria C

Darren L

Jawad K

Jawad C

What is a Database?

A database is a system designed for storing, managing, and retrieving data efficiently. First Introduced in 1970, Edgar F. Codd, a computer scientist at IBM, introduces the relational database model in his paper "A Relational Model of Data for Large Shared Data Banks."

It allows users to organize data in a structured format, ensuring easy access and manipulation. There are two types of databases that are relevant to current Industry, SQL and NoSQL databases.

What is SQL and NoSQL?

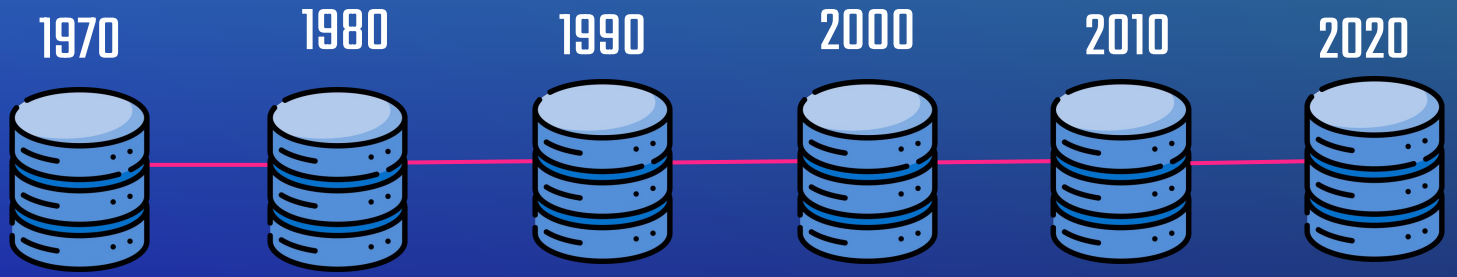
SQL (Structured Query Language):

- Relational databases that use structured schemas.
- Ideal for structured data with predefined relationships (e.g., tables).
- ACID Transaction (Atomicity, Consistency, Isolation, Durability)

NoSQL (Not Only SQL):

- Non Relational databases that allow for flexible schemas.
- Suitable for unstructured or semistructured data, accommodating rapid changes.
- BASE (Basically Available, Soft State, Eventually Consistent)

History


1970
1980
1990
2000
2010
2020
Birth of SQL
Rise of SQL
Growth of Relational Database
Emergence of NoSQL
NoSQL popularity Surge
Continued Evolution

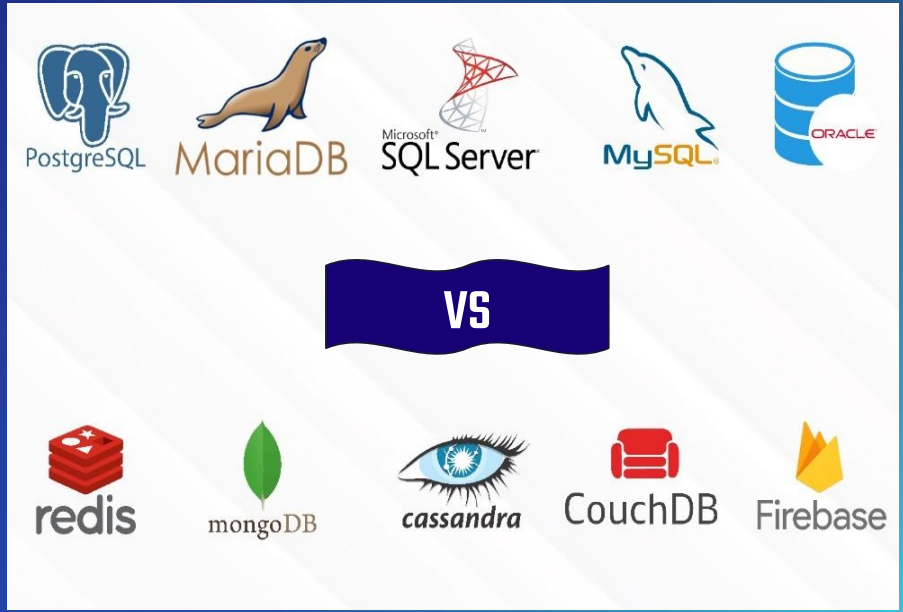
Examples of SQL and NoSQL

SQL Database

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle DB
- Azure DB

NoSQL Database

- MongoDB
- Redis
- Cassandra
- DynamoDB
- Firebase



Cloud DBs

Cloud Hosting allows users migrate databases remotely for better scalability .

Amazon RDS

Pros:

- Flexible ay-as-you-go model.
- Automatic backups of DB

Cons:

- Costly for large-scale deployment
- Scaling downtime

Microsoft Azure

Pros:

- Reduced latency - global distribution.
- Automated scaling by demand

Cons:

- Less budget-friendly than Amazon RDS.
- Query complexity increased due to consistency models.

Google Cloud

Pros:

- Easy integration with Google products
- Flat rate pricing for BigQuery, a service for data analytics.

Cons:

- Limited RDBMS offered
- Higher costs to transfer data out of ecosystem.

Hosting DBs on the Cloud vs Locally

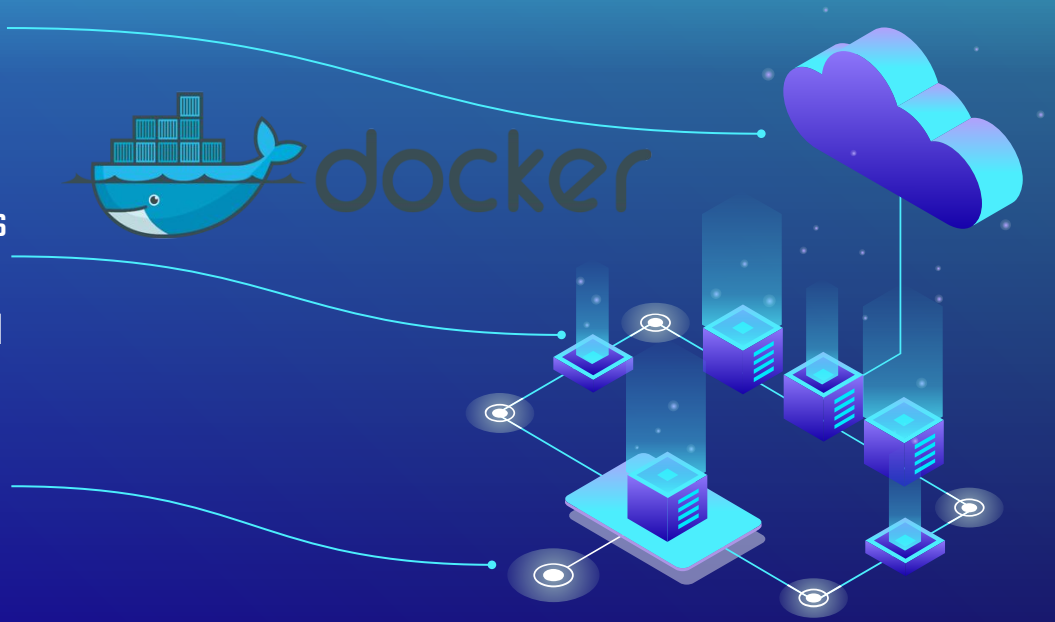
- Cloud Hosting :
  - Benefits include scalability, reduced maintenance, improved accessibility, and monitoring.
  - Can be more expensive in the long run, and you have less direct control over your data.
- On Premises Hosting :
  - Direct control over your data, can be more cost effective.
  - You're responsible for all maintenance, and scaling.

What is Docker?

Docker

Containerization :

- Ensures all users can access a service regardless of working environment
- Your project is portable and scalable
- Run your projects in isolation
- Lightweight solution as opposed to virtualization



Pros & Cons (SQL DBs)

MySQL

- Easily Managed
- Faster for Simple Queries

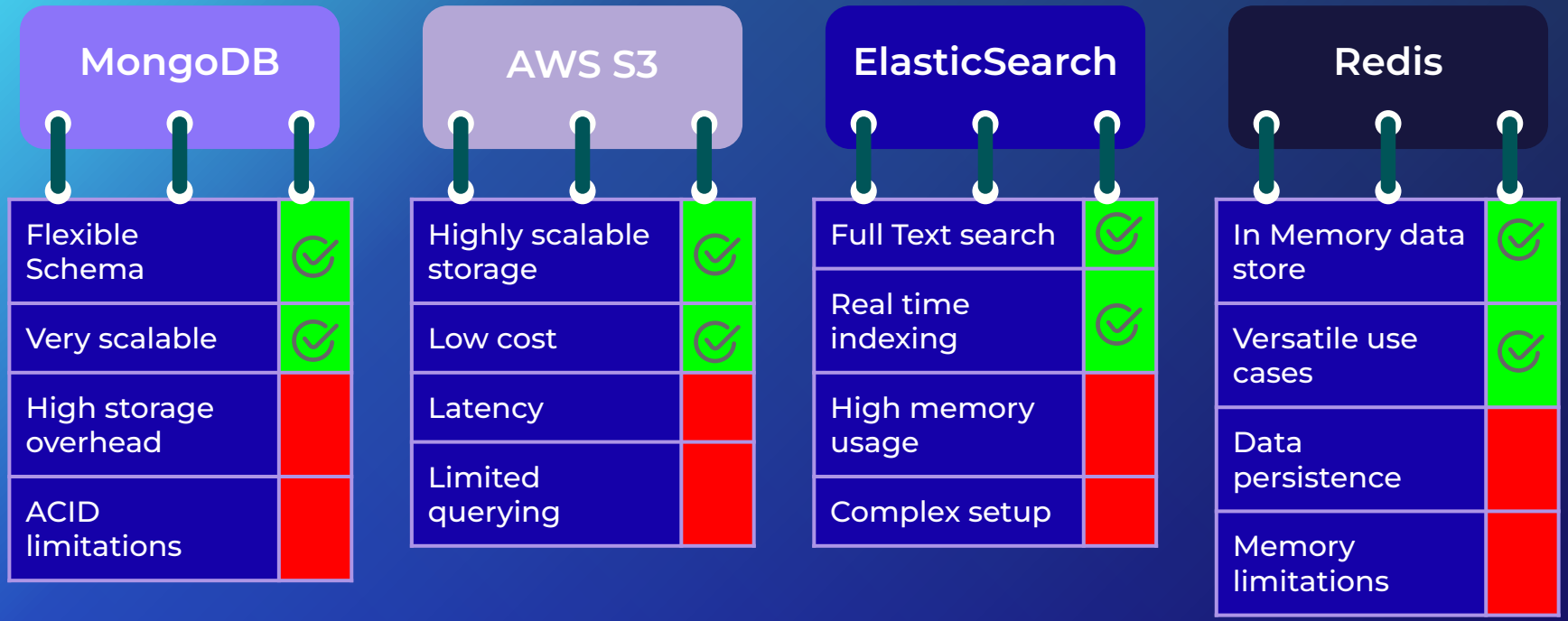
- Scalability Issues
- Limited Support for Advanced Features

PostgreSQL

- Advanced Features
- Extensibility

- Slower in simple readheavy scenarios
- More Complex to Set Up

Pros & Cons (NoSQL DBs & Others)



Conclusion And Recommendations

Small Projects:

- Recommended to use SQL databases for simple, structured projects, and easier management

Large Projects:

- Recommended to use NoSQL databases or cloud solutions for better scalability

Docker:

- Should be used to avoid the “it works on my machine” problem
- Great for portability, scalability, and security
- Very lightweight and easy to orchestrate

Thanks!

This was a presentation presented by Small Data