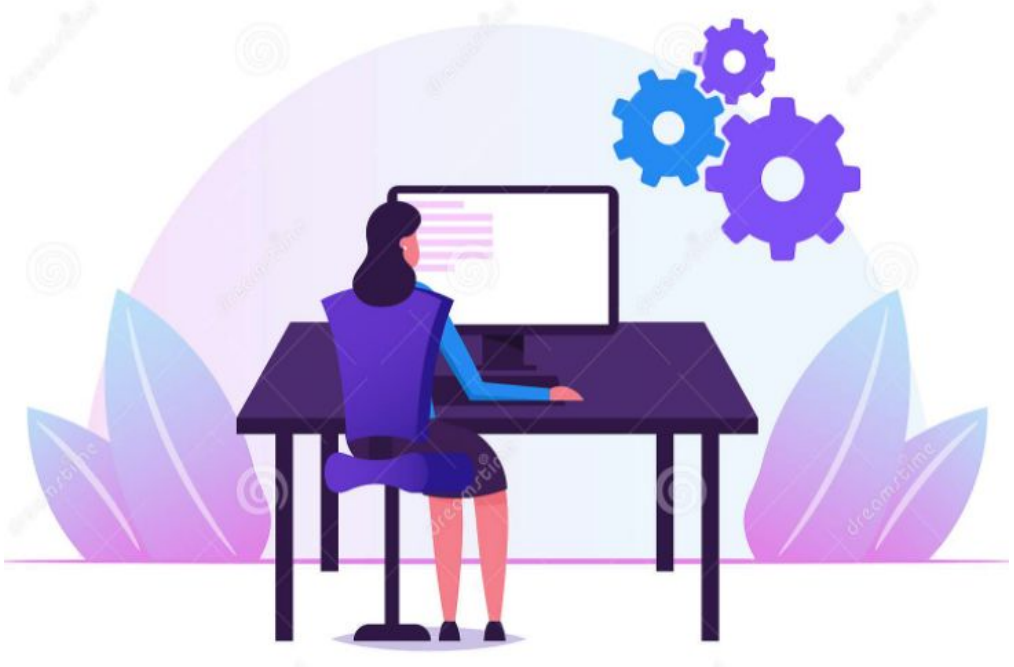
Welcome!



WWCode Digital + Backend

October 26, 2022

- We'll start in a moment. :)
- We are **RECORDING** tonight's event.
- We may plan to take screenshots for social media.
- If you are comfortable, turn the video ON. If you want to be anonymous, then turn the video off.
- We'll introduce the hosts & make some time for Q&A at the end of the presentation.
- Feel free to take notes.
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts.
    - Mute yourself when you aren't talking.
    - We want the session to be interactive.
    - Use the 'Raise Hand' feature to ask questions.
- **By attending our events, you agree to comply with our Code of Conduct.**

WOMEN WHO CODE /san-francisco

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Empower diverse women to excel in technology careers.
- Our vision: A tech industry where diverse women and historically excluded people thrive at any level.

Prachi Shah
Instructor,
Senior Software Engineer.
Director, WWCode SF

Harini Rajendran
Host,
Software Engineer, Confluent.
Lead, WWCode SF

- Database Design:
- What is a Database?
- Relational database
- Non-relational database.
- Relational vs. non-relational database.
- Data modeling.
- Design practices.
- Examples and demo.
- Q & A.

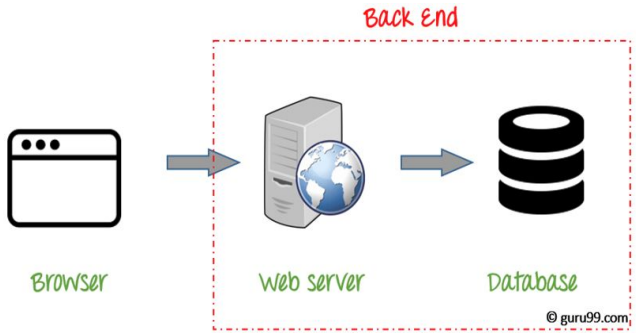
Disclaimer:
- Sessions can be heavy!
- Lots of acronyms.
- Instructor doesn't know everything.

WOMEN WHO CODE /san-francisco

Backend Engineering

- Design, build and maintain server-side web applications.

- Common terms: Client-server architecture, networking, APIs, web frameworks, platform, micro-service, databases, web fundamentals, operating systems, etc.

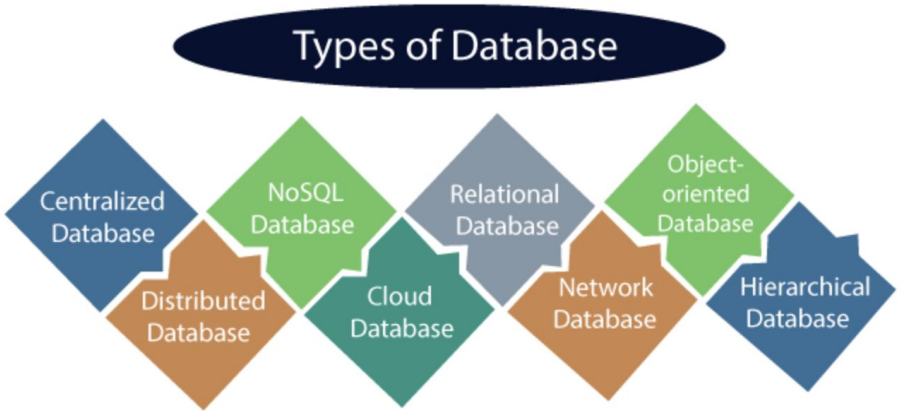


- Tech Stack: Java, PHP, .NET, C#, Ruby, Python, REST, AWS, Node, SQL, NoSQL, etc.

- Other domains: Front end engineering, full stack engineering, design & user experience, mobile development, devOps engineering, machine learning, etc.

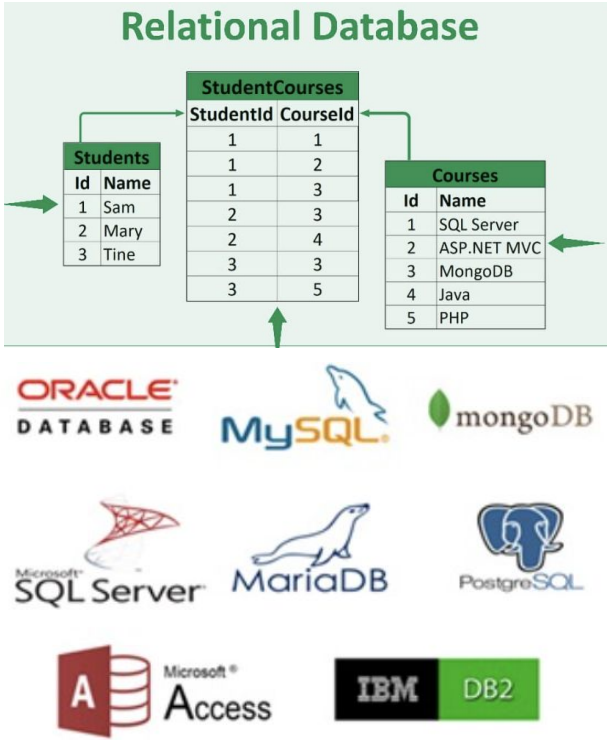
Database

- Collection of organized data.
- To store, retrieve and manage data.
- Supports different data types and formats.
- Tools: MySQL, Oracle DB, MongoDB, DynamoDB, Redis, etc.
- Supports features such as data redundancy, backup, replication, querying and scalability.



Relational Database

- Collection of one or more tables.
- Data is stored in a tabular format.
- Each table has rows and columns.
- Tables have relationships between them.
- Optimal to store structured data.
- Structured Query Language (SQL) to write and query data.
- ACID properties compliant.
- Data modification statements are called transactions.
- Ideal for banking transactions, insurance policies and wealth management data.
- Tools: MySQL, PostgreSQL, Oracle DB.



Relational Database

ACID properties:

- Atomicity: Any data modifications done per transaction is a single operation, completed successfully or fails.
- Consistency: Data is in valid and correct state after the end of a transaction.
- Isolation: Multiple transactions executed in isolation.
- Durability: Data modifications are permanent after successful execution of transactions.

Operations:

- Create, Drop, Alter table.
- Insert, update, delete record.
- Grant, revoke permissions to a user.
- Commit, rollback, save transaction.
- Select data for querying.

Relational Database

Examples:

- Create a table: `CREATE TABLE STUDENT (FIRST_NAME VARCHAR(50), LAST_NAME VARCHAR(50), DATE_OF_BIRTH DATE, STUDENT_ID INT);`
- Drop a table: `DROP TABLE STUDENT;`
- Alter a table: `ALTER TABLE STUDENT ADD (DEGREE VARCHAR(100));`
- Update a table column: `UPDATE STUDENT SET FIRST_NAME = 'John' WHERE ID = '3';`
- Select data from table to display data: `SELECT * FROM STUDENT;`

SELECT * FROM STUDENT;

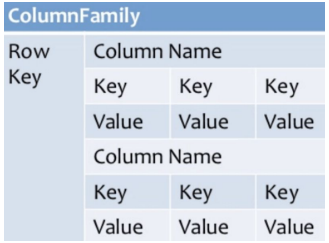
STUDENT_ID	ADVISOR	DATE_OF_BIRTH	DEGREE	ENROLLMENT_STATUS	FIRST_NAME	LAST_NAME	YEAR_OF_ENROLLMENT
1	Prof. Matt Peterson	1990-01-11	COMPUTER_SCIENCE	ENROLLED	John	Doe	2021
2	Prof. Matt Peterson	1990-01-11	COMPUTER_SCIENCE	ENROLLED	John	Doe	2021
3	Prof. Matt Peterson	1990-01-11	COMPUTER_SCIENCE	ENROLLED	John	Doe	2021
4	Prof. Greg Harris	1980-12-10	COMPUTER_SCIENCE	GRADUATED	Karla	Harris	2022
5	Prof. Perry Jameson	1990-11-30	COMPUTER_SCIENCE	ENROLLED	Jamie	Dorian	2020
6	Prof. Turk George	1990-12-11	ELECTRICAL_ENGINEERING	ENROLLED	Elliot	Reed	2019

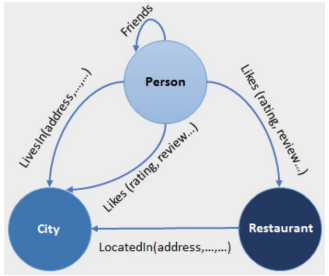
Non-relational Database

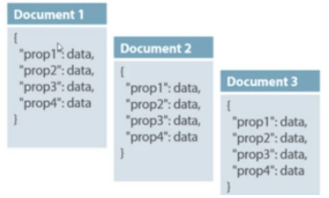
- Non-relational data is persisted in various non-tabular formats.
- NoSQL stands for ‘Not Only SQL’.
- Examples: Documents, key-value pairs, Graph (node, edges),
- Data model varies per database.
- Ideal for big data sets.
- BASE properties compliant.
    - Basically available: Data available as per CAP theorem.
    - Soft state: Data may change without new input/updates.
    - Eventual consistency: Data will be consistency over time with multiple copies of data for high availability and scalability.
- CAP theorem:
    - Consistency: Data consistent across systems.
    - Availability: Data available but not most updated (at a given time).
    - Partition Tolerance: Data available despite of system failures.
    - Only two properties can be achieved.

Non-relational Databases

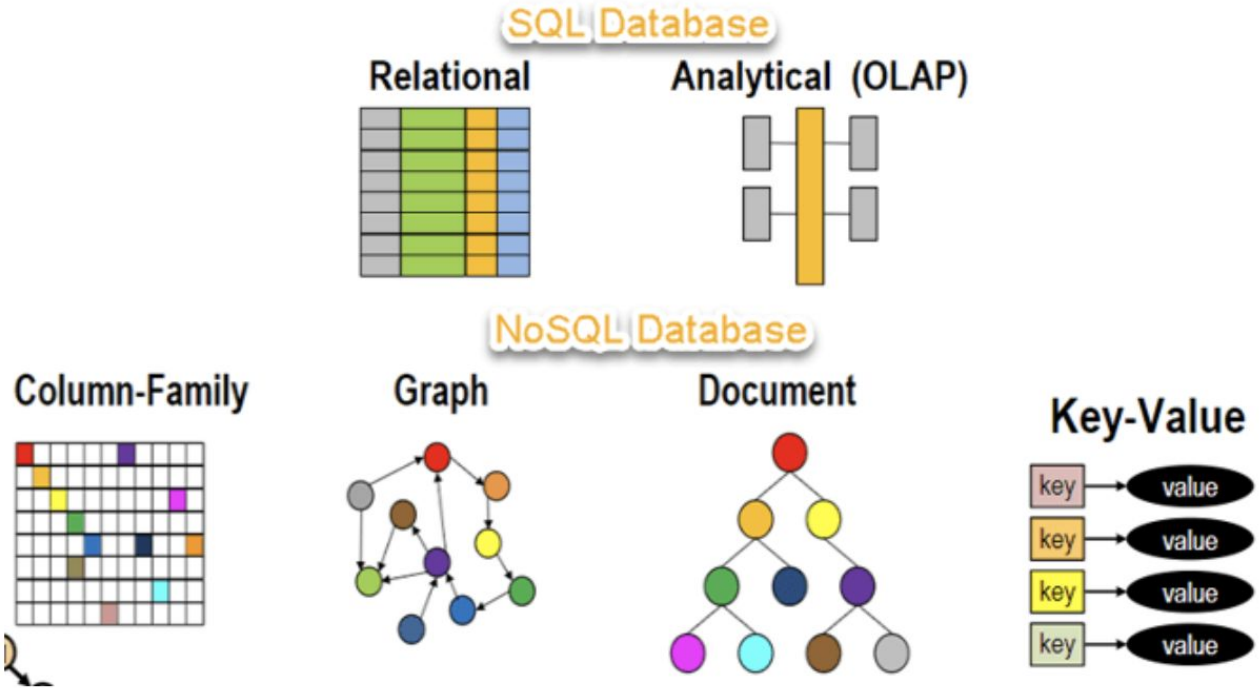
- Key-value pair:
  - Data stored as hash table, with unique key and JSON, BLOB, String value.
  - Schema-less data such as arrays or dictionaries.
  - Tools: Redis, DynamoDB, Memcached.
- Column-wide:
  - Data stored in contiguous large columns.
  - High performance querying for business intelligence & data warehouses.
  - Tools: HBase, Cassandra.
- Graph-based:
  - Data entities stored as nodes and relationships are edges.
  - Multi-relational data ideal for social networks, spatial data and logistics.
  - Tools: OrientDB, NeoJ.
- Document store:
  - Key-value pair where value is a document stored in JSON or XML formats.
  - Ideal for low querying data for blogging platforms and CMS applications.
  - Tools: SimpleDB, CouchDB, MongoDB.







Relational vs. Non-relational Database



Relational vs. Non-relational Database

Relational Database:

- ACID compliant.
- SQL for efficient data querying.
- Supports structured data models.
- Rows and columns data structure.

Non-relational Database:

- ACID sacrificed for high scalability and eventual consistency.
- Offers limited data querying capabilities.
- Supports structured, semi-structured and unstructured data models.
- Graph, JSON, document, column-wide data structures.

Data Modeling

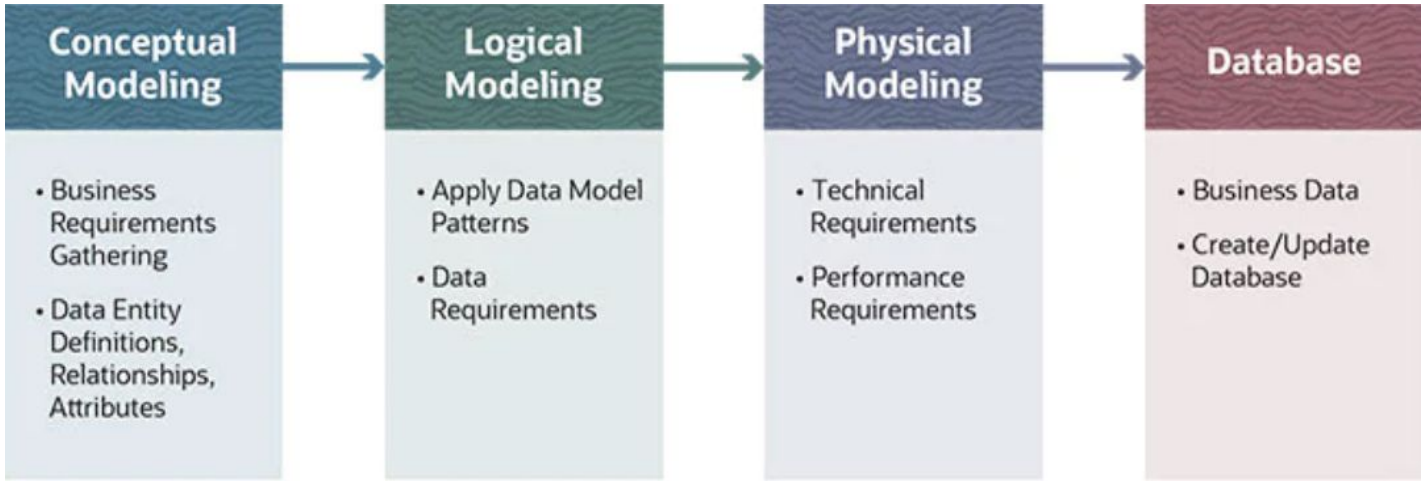
- Organize and determine relationships between data.
- Define constraints and select data types.
- Structure data for business needs.
- Data modeled after real business and product entities.
- Example:

Design Practices

- Keep it simple. Design should be intuitive.
- Consider future modifications and growing volumes of data.
- Choose the correct software for the data type and business needs.
- Reduce data redundancy.
- Accommodate for data analytics and reporting requirements.
- Determine relationships as per business needs.
- Know when to choose SQL vs. NoSQL.
- Optimize data for querying.
- Determine performance requirements.
- Understand maintenance challenges.
- Models:
    - Conceptual.
    - Logical.
    - Physical.
    - Database.

Design Practices

- Conceptual: Entities, attributes, data types and relationships.
- Logical: Apply business rules.
- Physical: Tables, columns, keys and indexes.
- Database: Create, update data and tables.



Demo

Relational database and SQL querying.

Backend Engineering

References:
- Modern databases
- NoSQL Tutorial

Backend Study Group:
- Presentations on GitHub and session recordings are found on WWCode YouTube channel
- Upcoming sessions:
    - November 2nd, 2022 about Improve your code debugging skills
    - December 8th, 2022 about Git and Version Control System

Women Who Code:
- Technical Tracks and Digital Events for more events.
- Join the Digital mailing list for updates about WWCode.
- Contacts us at: contact@womenwhocode.com
- Join our Slack workspace and join *#backend-study-group*!

You can unmute and talk or use the chat.

WOMEN WHO CODE® /san-francisco