Lesson 8: Intro to Databases

What are databases / The different kinds of databases / Design a database schema from scratch

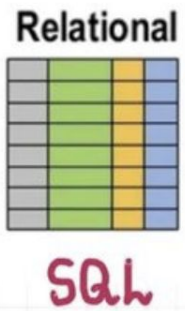
What is a database?

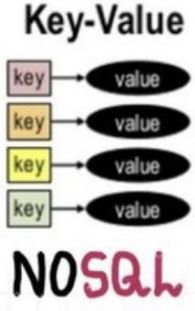
A database is a collection of data that is organized in a specific way to make it easily accessible and manageable.

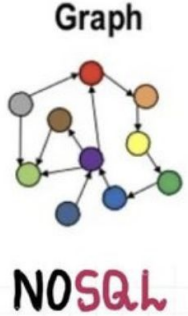
- This means that the data is stored in a structured manner that allows for efficient searching, sorting, and querying of the data.
- Databases can be used to store many different types of data, such as text, numbers, images, videos, and more.

Examples of different kinds of databases

1. **Relational databases:**
   - These are the most common type of databases and are used in a wide range of applications.
   - Relational databases store data in tables, which consist of rows and columns. Each row represents a single record in the database, and each column represents a different attribute or piece of information about the record.
   - Relational databases use a schema to define the structure of the database and the relationships between the tables.
   - Examples of relational databases include MySQL, PostgreSQL, and Oracle.
2. **Document-oriented databases:**
   - These databases store data in a document format, such as JSON or XML.
   - Each document represents a single record in the database and can contain nested or hierarchical data structures.
   - Document-oriented databases are flexible and can handle a wide variety of data types and structures.
   - Examples of document-oriented databases include MongoDB and Couchbase.
3. **Key-value stores:**
   - These databases store data in a simple key-value format, where each record is identified by a unique key.
   - Key-value stores are designed for high-performance and can handle large volumes of data with low latency.
   - Examples of key-value stores include Redis and Amazon DynamoDB.
4. **Graph databases:**
   - These databases store data in nodes and edges to represent complex relationships between data points.
   - Graph databases are designed for applications that require complex querying and analysis of relationships between data points.
   - Examples of graph databases include Neo4j and Amazon Neptune.

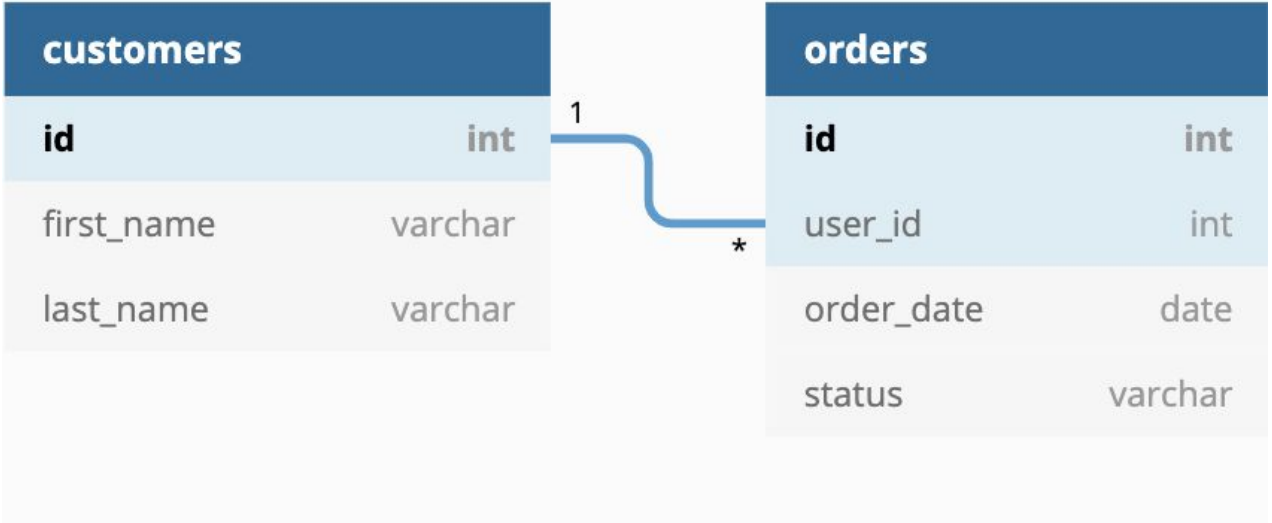






More on Relational Databases

- Relational databases are based on the relational model of data.
  - The relational model is based on the concept of relations, which are tables that represent entities and their attributes.
  - The tables are related to each other through common attributes or keys.
- Relational databases use primary and foreign keys to enforce data integrity and relationships between tables.
  - A primary key is a unique identifier for a record in a table.
  - A foreign key is a reference to a primary key in another table.
  - Foreign keys are used to establish relationships between tables, and ensure referential integrity.



What is SQL?

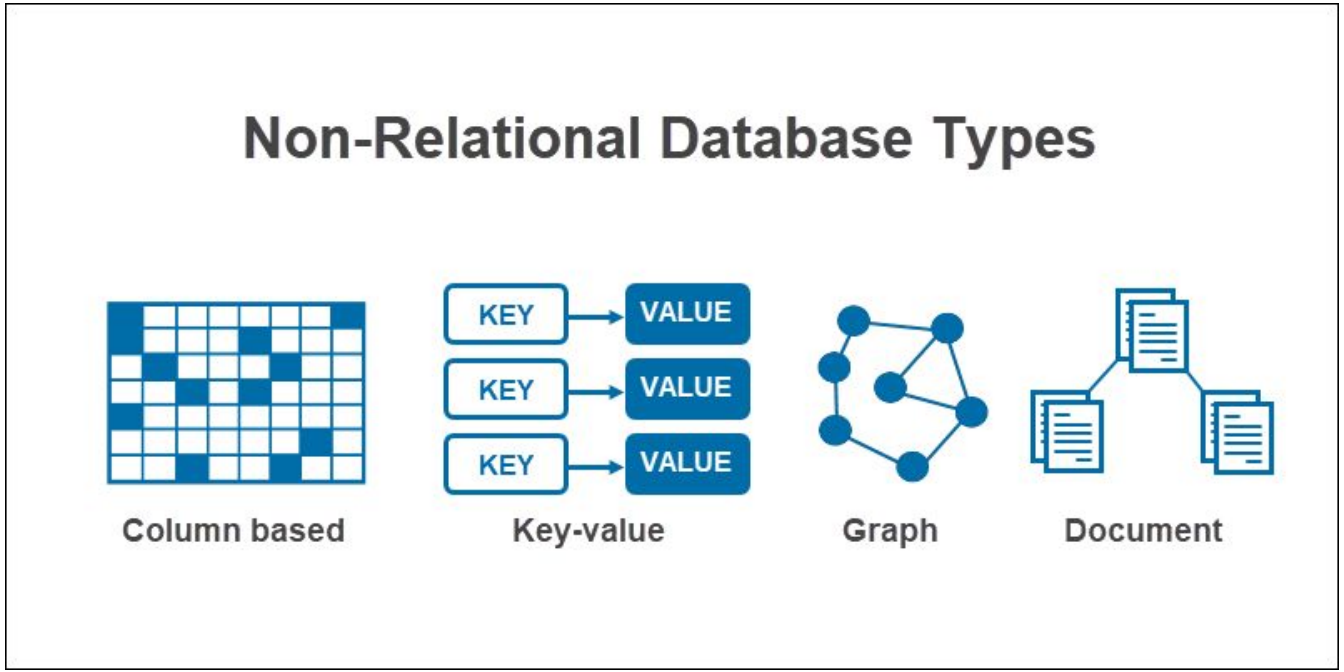
1. **Structured Query Language (SQL):**
   - SQL is a programming language used to manage and manipulate data in relational databases.
   - It was developed in the 1970s and has become the standard language for working with relational databases.
   - SQL is used to create, modify, and query databases.
2. **Using SQL with Relational Databases:**
   - SQL is designed to work with relational databases, which store data in tables.
   - SQL is used to create tables, add and remove data from tables, and query data from tables.
   - SQL allows users to manipulate the data stored in the tables, including creating new records, updating existing records, and deleting records.
3. **Basic SQL Commands: (These commands are the most basic and commonly used SQL commands)**
   - SELECT: retrieves data from one or more tables in a database.
   - INSERT: adds a new record to a table in the database.
   - UPDATE: modifies an existing record in a table in the database.
   - DELETE: removes a record from a table in the database.

We will get hands on with SQL in the next lesson

Non-relational databases or “NoSQL”

“NoSQL databases”

A general term that refers to non-relational databases that use a variety of data models.



SQL vs NoSQL

While SQL databases have been the traditional choice for many years, NoSQL databases have gained popularity in recent years due to their scalability and flexibility. SQL databases are still preferred for applications that require complex transactions and data integrity, while NoSQL databases are better suited for applications that require high scalability and flexibility. Ultimately, the choice between SQL and NoSQL depends on the specific needs of the application and the data being stored.

SQL

Relational Data Model

Pros
- Easy to use and setup.
- Universal, compatible with many tools.
- Good at high-performance workloads.
- Good at structure data.

Cons
- Time consuming to understand and design the structure of the database.
- Can be difficult to scale.

No SQL

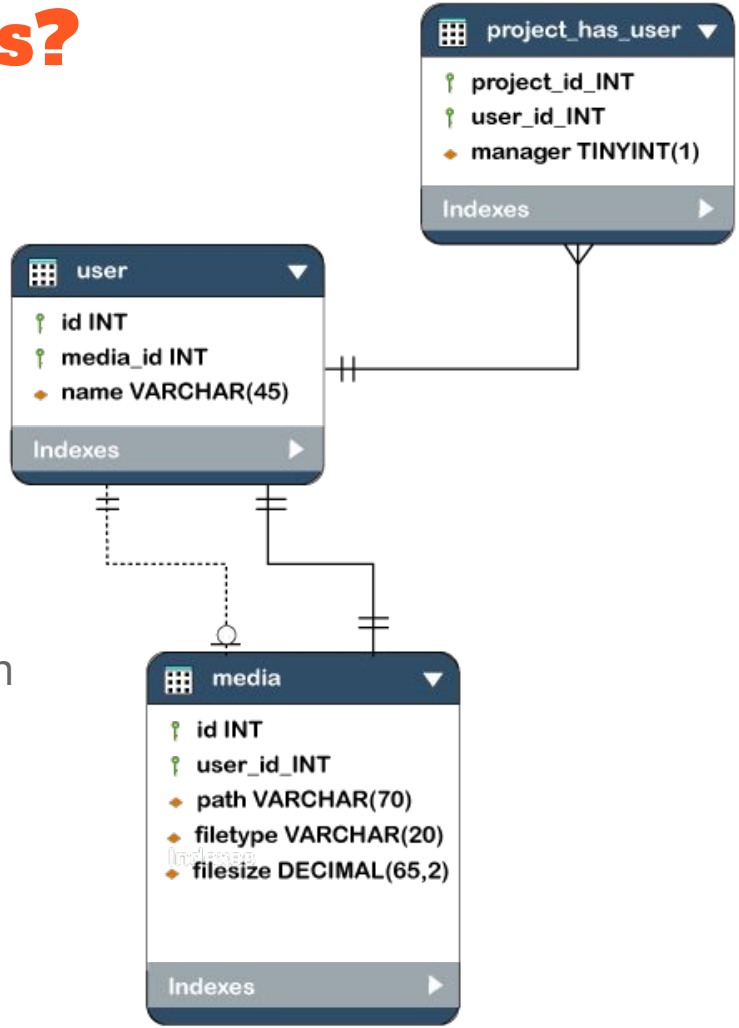
Document Data Model

Pros
- No investment to design model.
- Rapid development cycles.
- In general faster than SQL.
- Runs well on the cloud.

Cons
- Unsuited for interconnected data.
- Technology still maturing.
- Can have slower response time.

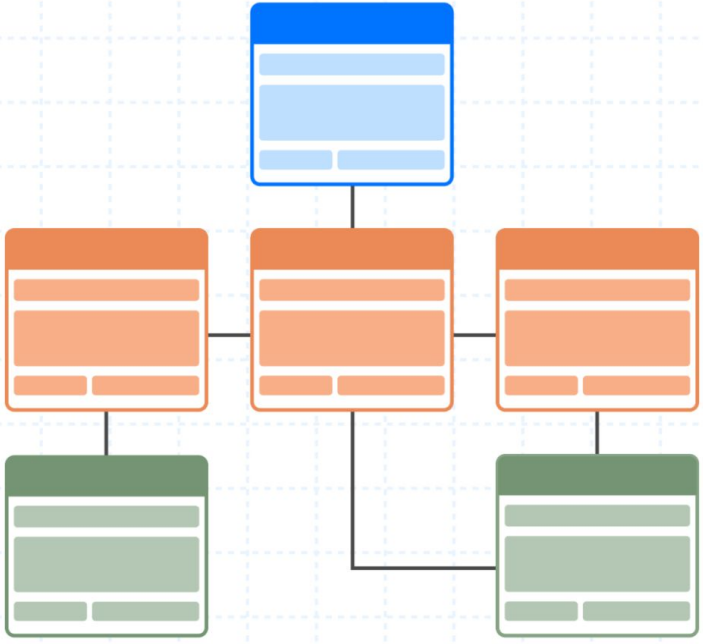
What are database schemas?

- A blueprint for organizing data in a database.
- Describes the structure of a database, including tables, columns, and relationships between tables.
- It defines the data types and constraints for each column, such as whether a column can contain null values or whether it must be unique.
- Helps ensure data consistency and accuracy.



Creating a database schema

- Identify the entities that need to be represented in the database.
- Define the relationships between the entities.
- Create the tables, columns, and relationships in the database schema.
- Populate the tables with data.



Lets create a database schema for pleb-wallet

"Creating a database schema is a lot like a game of Tetris, except the blocks keep changing shape, and the rules change as you go."

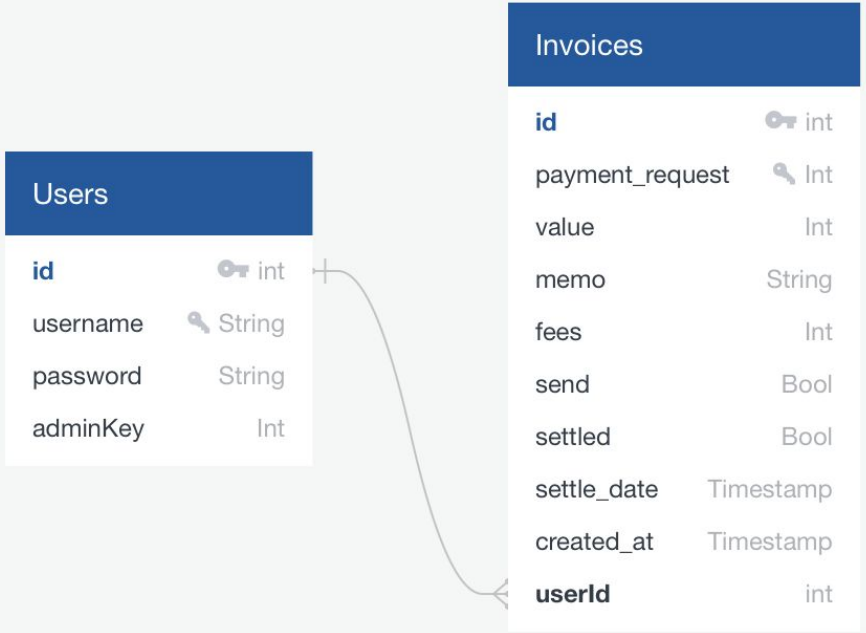
Let's checkout QuickDB to create our schema

Visit https://app.quickdatabasediagrams.com/#/

Our final schema

Users
-
id PK int
username String UNIQUE
password String
adminKey String default=null

Invoices
-
id PK int
payment_request Int UNIQUE
value Int
memo String
fees Int
send Bool
settled Bool
settle_date Timestamp
created_at Timestamp default=GETUTCDATE()
userId int FK >- Users.id



Resources

- Databases overview for beginners (video) - https://www.youtube.com/watch?v=wR0jg0eQsZA
- Introduction to Databases (article) - https://www.prisma.io/dataguide/intro/comparing-database-types
- What is a Database schema? (article) - https://www.educative.io/blog/what-are-database-schemas-examples