# Database Fundamentals

# What Is Database

A database is an organized collection of structured data that is stored electronically and can be easily accessed, managed, and updated.

# Why Use Database

- Organize large amounts of information in a structured way.
- Using rules and constraints, databases ensure data remains valid and reliable.
- Easy Access & Management by SQL queries
- Databases offer user roles, permissions, and encryption to control who can see or edit the data.
- Relationships Between Data.
- Support for automated backups and restoration, helping to prevent data loss.
- Enables real-time updates and access in web and mobile apps.

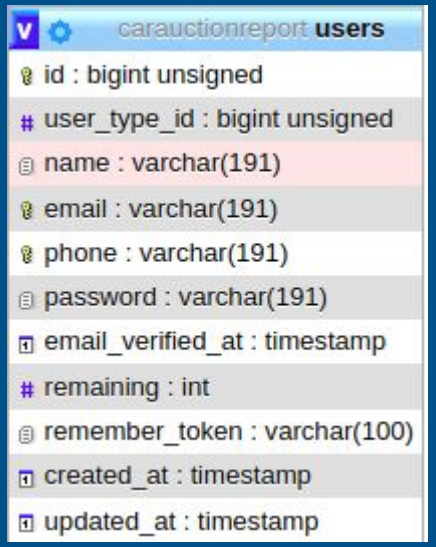
# Database Table

- A table stores data in rows and columns.
- Each table represents an entity (e.g., Users, Orders).

# Columns & Data Types

- Columns represent attributes of an entity.
- Each column has a name and a data type.
- Common Data Types: INT (Numbers), VARCHAR (String), DATE, BOOLEAN, FLOAT
- Column properties: NOT NULL, DEFAULT, AUTO_INCREMENT, UNIQUE, PRIMARY KEY

# Example Table



# Database Design Principles

- Data normalization
- Should have Primary Key
- Use meaningful table and column names
- Ensure scalability and maintainability

# Keys

Keys are special fields (columns) used to identify, relate, and maintain integrity of data in tables.

# Primary Key

- Uniquely identifies each row in a table.
- Cannot be NULL or duplicate.
- user_id in the Users table.

# Foreign Key

- A field in one table that refers to the Primary Key in another.
- Used to create relationships between tables.
- Enforces referential integrity.

# Unique Key

- Ensures that all values in a column (or group of columns) are different.
- Unlike Primary Key, it can have one NULL.

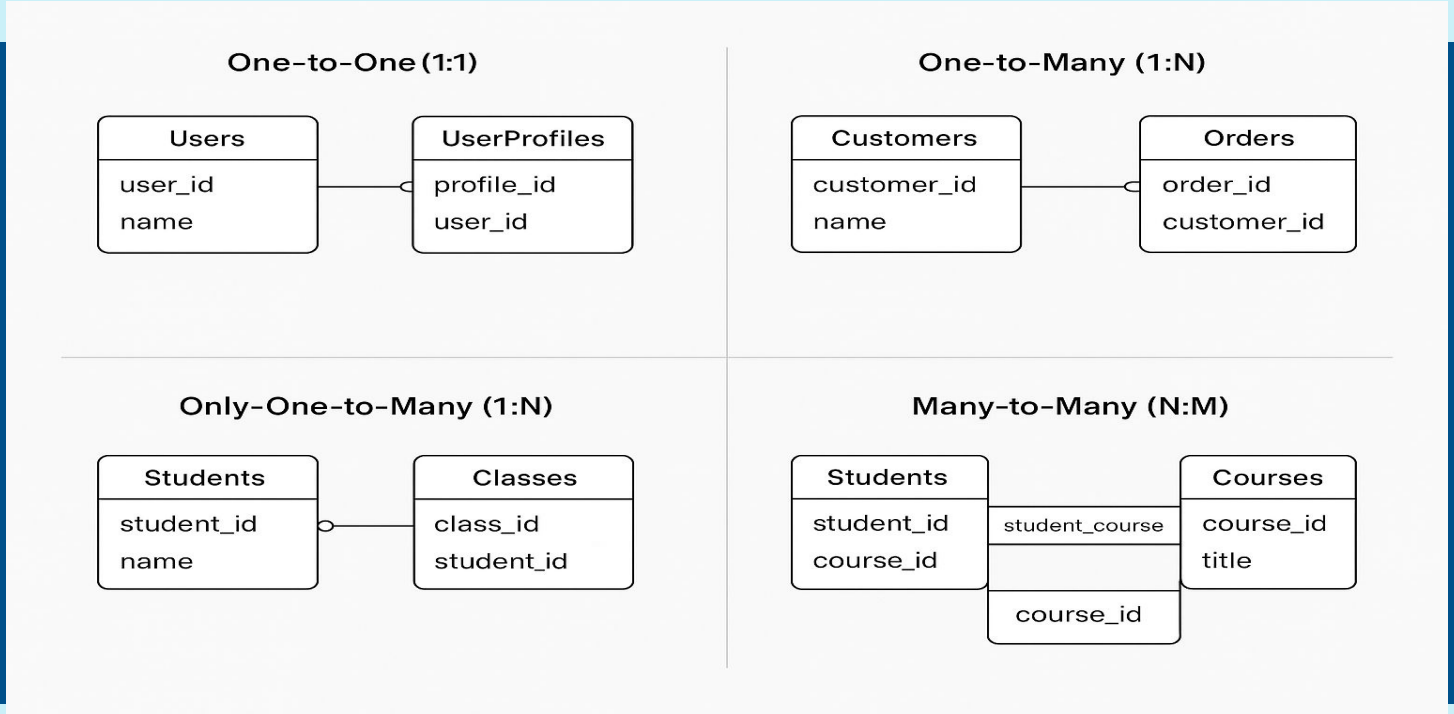
# Constraints in Database

- Rules enforced on data in tables
- Types of Constraints:
  - NOT NULL
  - UNIQUE
  - DEFAULT
  - FOREIGN KEY
  - PRIMARY KEY

# Database Relationship

- One-to-One (1:1) - One record is linked with another table.
- One-to-Many (1:N) - One record is related with many tables.
- Only-One-to-Many (1:N) - Similar to 1:N but a child record must have only one parent
- Many-to-Many (N:M) - Records in Table A can relate to many in Table B and vice versa

# Relationship Example



# Q&A

ANY QUESTIONS?