Foundations of Database Management

Why Databases Matter for Software Engineers

semicolon

Introduction

What is Data?

Why do we need Databases?

Importance of Database to the Internet and Smartphone World

1. **PC and Networking Evolution**
   a. PCs became widely available with the Apple II (1977) and IBM PC (1981).
   b. Ethernet networking technology (developed in the 1970s) enabled LANs, standardized in 1983.
2. **Internet Development**
   a. The Internet began as ARPA-NET in 1969 and evolved into a global network.
   b. The World Wide Web (WWW) became accessible in 1993, popularized by web browsers.
3. **E-Commerce Emergence**
   a. Online retail (e.g., Amazon in 1995) and hybrid stores (e.g., Best Buy) facilitated online shopping.

4. **Web 2.0 and Social Platforms**

   a. In the 2000s, Web 2.0 enabled user-generated content.

   b. Platforms like Facebook, Wikipedia, and Twitter emerged and thrived.

5. **Mobile Phone Evolution**

   a. Mobile phones were demonstrated in the 1970s; smartphones emerged with the iPhone (2007) and Android (2008).

   b. Smartphones and tablets became widely used, with apps complementing web applications.

6. **Dependence on Databases**

   a. Databases are essential for Web applications and smartphone apps.

   b. Data (facts and numbers) are stored in databases, processed to provide information.

7. Database Usage in Popular Applications

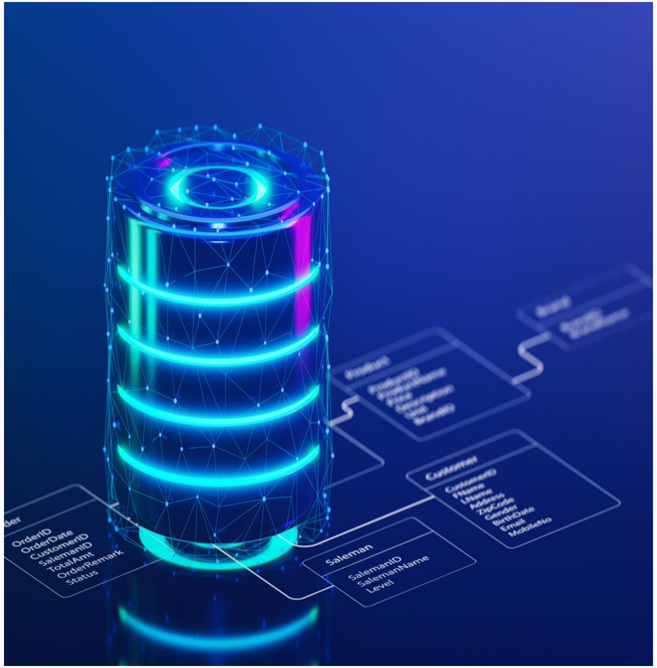
a. **Facebook:** Posts, comments, likes, and photos are stored in databases.

b. **Twitter:** Tweets are stored in databases for retrieval and display.

c. **Amazon:** Search results rely on database queries to match user input.

Characteristics of Databases

1. **Purpose of Databases**
   1. Databases help people keep track of things.
   2. The most commonly used type is the **relational database**.
2. **Data Storage in Tables**
   1. A relational database stores data in **tables** with rows and columns, similar to a spreadsheet.
   2. Each table contains data about a specific type of entity or thing.



semicolon

3. **Multiple Tables**

    a. Databases typically consist of multiple tables, each focused on a different subject.

    b. Example: A database may have a **STUDENT** table for student data and a **CLASS** table for class data.

4. **Rows and Records**

    a. Each **row** represents a specific occurrence or instance of an entity.

    b. Rows are also referred to as **records**.

5. **Columns and Fields**

    a. Each **column** represents a characteristic shared by all rows in the table.

    b. Columns are also called **fields**.

    c. Example: In a STUDENT table, columns might include **`StudentNumber`** and **`LastName`**.

Relationships in Databases

1. **Importance of Relationships**

   a. A database is incomplete without showing the **relationships** among its data.

   b. Data alone (e.g., in a GRADE table) is useless without context (e.g., knowing which students earned which grades).

2. **Complete Database Structure**

   a. A complete database includes both data and relationships among the data.

   b. Example: A student’s grade can be linked to the class they took.

3. **Primary Key**

   a. Each row in a table is uniquely identified by a **primary key**.

   b. Primary keys are used to create relationships between tables.

   c. Example: In the STUDENT table, **StudentNumber** is the primary key, and each value uniquely identifies a student.

4. **Surrogate Key**

    a. If primary keys are automatically generated by the database, they are called **surrogate keys**.

5. **Composite Key**

    a. When more than one column forms the primary key, it is called a **composite key**.

    b. Example: In the GRADE table, **StudentNumber** and **ClassNumber** together form a composite key.

6. **Foreign Key**

    a. A **foreign key** in one table links to the primary key in another table.

    b. In the GRADE table, **StudentNumber** and **ClassNumber** serve as foreign keys to link to the STUDENT and CLASS tables.

7. **Relationships Between Tables**

    a. A **one-to-many** relationship exists when one row in a table (e.g., STUDENT) can be linked to multiple rows in another table (e.g., GRADE).

Databases Create Information

Information is:

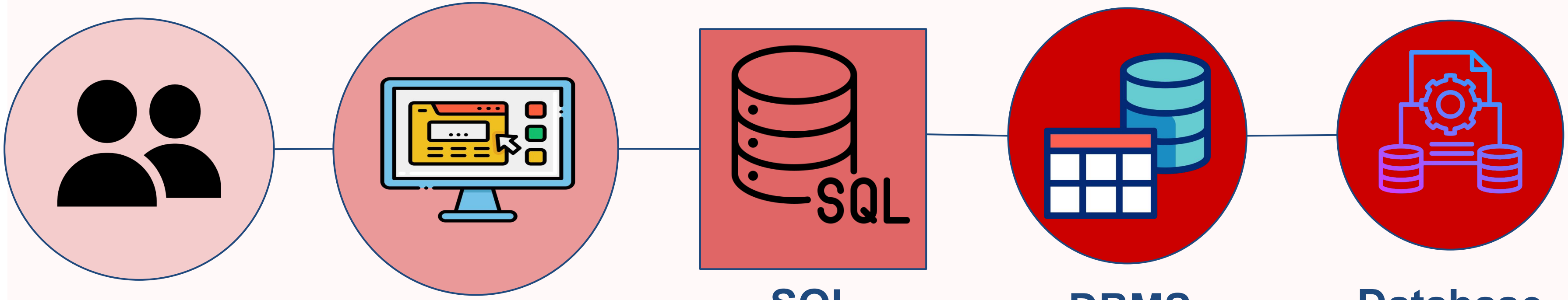
- Knowledge derived from data.
- Data presented in a meaningful context.
- Data processed by operations like summing, ordering, averaging, grouping, or comparing.

- Databases record **data** (facts and numbers) but are designed to produce **information**.
- Data in databases can be manipulated to create useful information (e.g., GPA, average class GPA, average number of students per class).
- **SQL (Structured Query Language)** is used to query databases and produce information.
- Relational databases store data in **tables** and represent the **relationships** among rows.
- This structure facilitates the **production of information** from raw data.

1. **Database**
    - A collection of related tables and other structures.
    - It is a self-describing collection of integrated tables.
    - An Integrated table is a table that stores both data and the relationships among the data.

2. **Database Management System (DBMS)**
    - A computer program used to create, process, and administer the database.
    - Receives SQL requests and translates them into actions on the database.
    - DBMS is a complex program typically licensed from a software vendor.

Components of a Database System



User

Database Application

SQL

DBMS

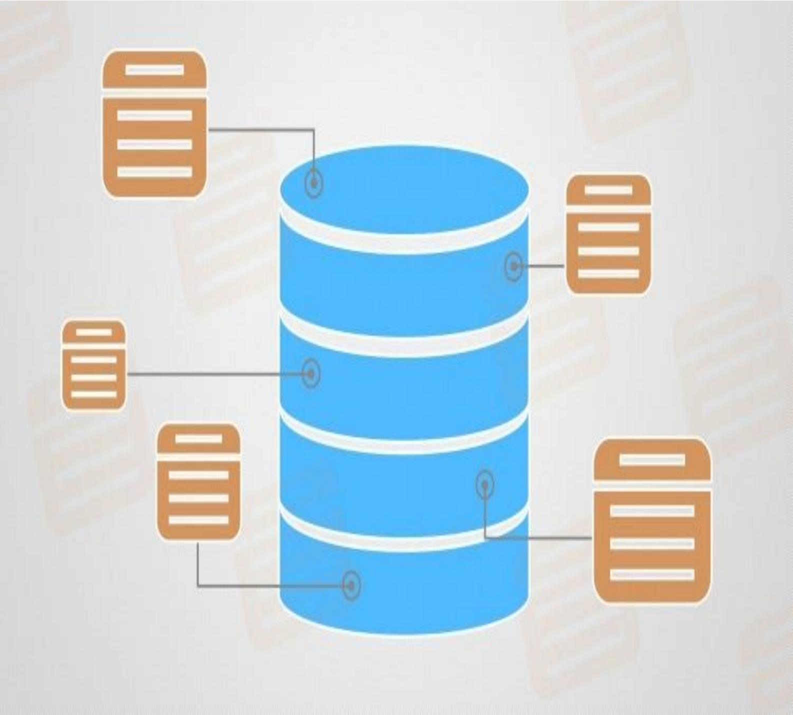
Database

3. **Database Application**
    - A set of programs acting as an intermediary between users and the DBMS.
    - Sends SQL statements to the DBMS to read or modify data.
    - Presents data to users in forms and reports.
    - Can be acquired from software vendors or written in-house.

4. **Users**
    - Employ database applications to track and manage data.
    - Use forms to read, enter, and query data, and generate reports to convey information.

Database Design

Database design (as a process) is the creation of the **proper** structure of database tables, the **proper** relationships between tables, **appropriate** data constraints, and other structural components of the database.



Types of Database Design

- Database design from existing data
- Database design for new systems development
- Database redesign of an existing database

1. **Database Design from Existing Data**

- Constructed from existing data, such as spreadsheets or text files.
- Developers determine the structure of the new database.
- Involves organizing data from multiple sources, sometimes from other databases, such as for business intelligence (BI) systems.
- The process involves considering how tables or data should be related and the use of normalization principles (discussed in Chapter 3).

2. **Database Design for New Systems Development**

- Used for developing new information systems, with a focus on user requirements (data entry forms, reports, use cases).
- The process involves creating a data model (blueprint) from the user requirements and then transforming that model into a database design.
- The data model is used to guide the design process before building the actual database in a DBMS.
- The entity-relationship (ER) model is a popular tool for data modeling (covered in Chapter 5) and is later transformed into the database design (covered in Chapter 6).

3. **Database Redesign**

- **Adapting to New/Changing Requirements** (Database Migration):
  - Changes include adding, modifying, or removing tables, relationships, and data constraints to meet new needs.
- **Database Integration**:
  - Merging two or more databases, often during the adaptation or removal of legacy systems or when integrating enterprise applications.

SQL vs NoSQL

Feature	SQL	NoSQL
Schema	Fixed, Predefined	Dynamic, Flexible
Data Model	Table, Rows	Documents, Key-Value
Use Cases	Structured data	Large, Unstructured

semicolon

Hands-On Setup

Install Mysql:
https://www.geeksforgeeks.org/how-to-install-mysql-in-windows/

Activity

Objective: Create a simple relational database schema.

Steps:

1. Open MySql workbench.
2. Create a database: `CREATE DATABASE user_db;`.

3. **Create a table:**

```
    CREATE TABLE users (
       id SERIAL PRIMARY KEY,
       name VARCHAR(100),
       email VARCHAR(100) UNIQUE
);
```

Assignment

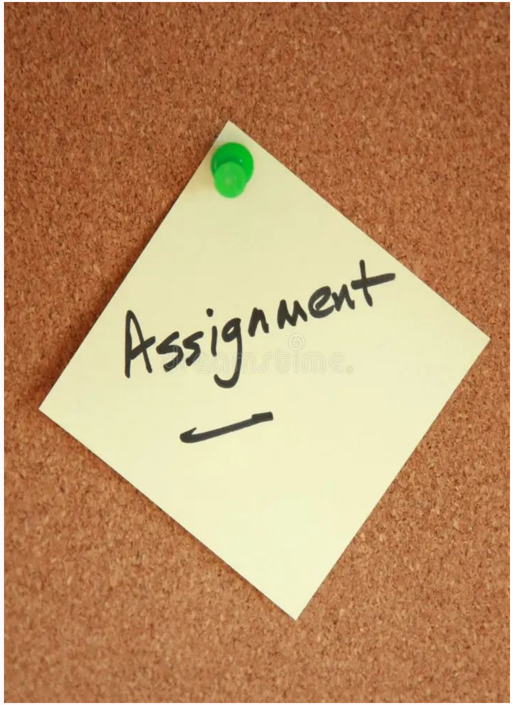
Task:

Design a relational database schema for an **e-commerce project**.

Include:

a. Tables for `users`, `products`, and `orders`.

b. Define primary and foreign keys.



·THANK YOU·

ANY QUESTION?