

Lecture one: Intro to Databases

Sunday, 29 September 2024

20:08

Database: A structured collection of data stored electronically. Can be as simple as a list or a

Examples: Customer lists, employee records, or web application data.

DBMS (Database Management System): Software used to manage databases. It helps users

Examples: MySQL, Oracle, Microsoft SQL Server, PostgreSQL.

Types of Databases:

1. Text-Based Databases:

- Structure: Often stored as simple text files (e.g., CSV, JSON).
- **Advantages:** Easy to read, portable.
- **Disadvantages:** Difficult to search, lacks consistency checks, limited to small-scale

Example- A CSV file storing student grades:

Name, Subject, Mark

John, Math, 85

Lisa, Physics, 90

2. File System Databases:

- Structure: Data is stored in files and directories, with each file representing one
 - **Advantages:** Structured and simple to implement.
 - **Disadvantages:** Difficult to search or manipulate the data, prone to inconsistency
- Examples: Early database systems like NTFS, FAT32.

s complex as interconnected tables of information.

s define, create, maintain, and control access to the database.

scale data storage.

e record.

cies due to lack of validation.

3. Hierarchical Databases:

- Structure: Data is organized in a tree-like structure, where each parent node has one or more child nodes.
 - **Advantages:** Ensures data integrity through parent-child relationships.
 - **Disadvantages:** Only supports one-to-many relationships, difficult to handle many-to-many relationships.
- Example: IBM's IMS system (Information Management System).

4. Relational Databases:

- Structure: Data in tables, checks for consistency and relationships through foreign keys.
 - Introduced: By Edgar F. Codd in 1969. Data is stored in tables (relations) with rows and columns.
 - Advantages: Supports complex queries and relationships (one-to-one, one-to-many, many-to-many).
 - Disadvantages: More complex to implement and requires knowledge of SQL (Structured Query Language).
- Examples: Oracle, MySQL, PostgreSQL.
- Example Query:

```
SELECT * FROM Students WHERE Grade > 80;
```

5. NoSQL Databases:

Types:

- Document-Oriented: Stores data as documents (e.g., JSON, BSON, XML) in a collection.
- Key-Value Stores: A collection of key-value pairs, similar to a dictionary.
- Graph Databases: Uses nodes and edges to represent relationships between data.
- Column-Oriented: Stores data in columns rather than rows.

Advantages: **Highly scalable and flexible**, excellent for handling **unstructured data**.

Disadvantages: **Lack of consistency** in some cases, no standardized query language.

Examples: MongoDB (Document-oriented), Redis (Key-Value Store), Neo4j (Graph Database).

as one or more child nodes.

any-to-many relationships.

ries.

rows and columns.

many, many-to-many). Data integrity ensured through constraints (primary and foreign keys).

Structured Query Language), hard to make efficient queries.

flexible schema.

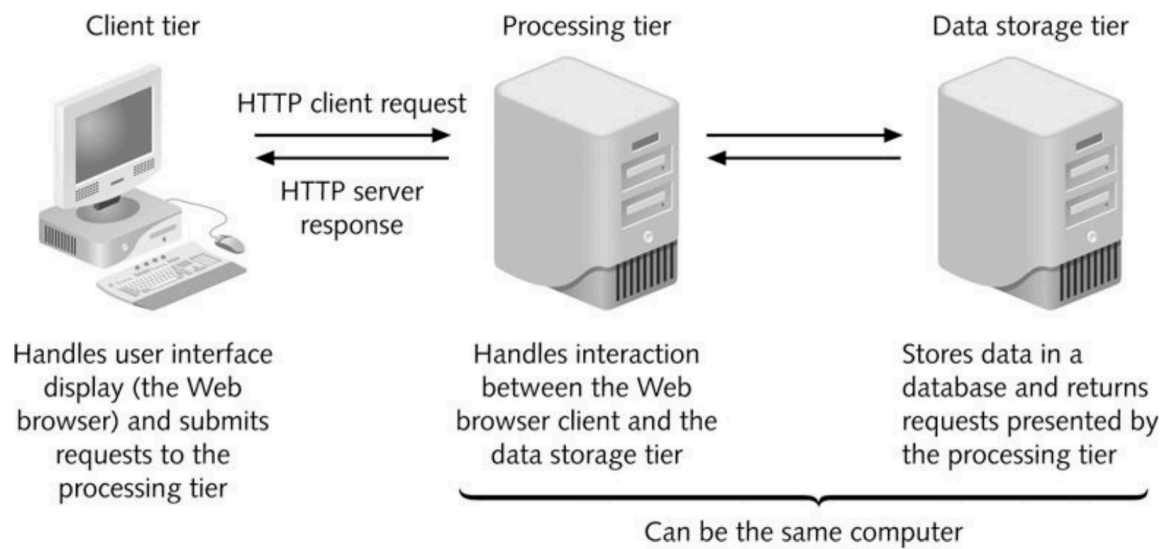
data.

ata.

uage- writing lots of code.

h).

Client/Server Architecture:

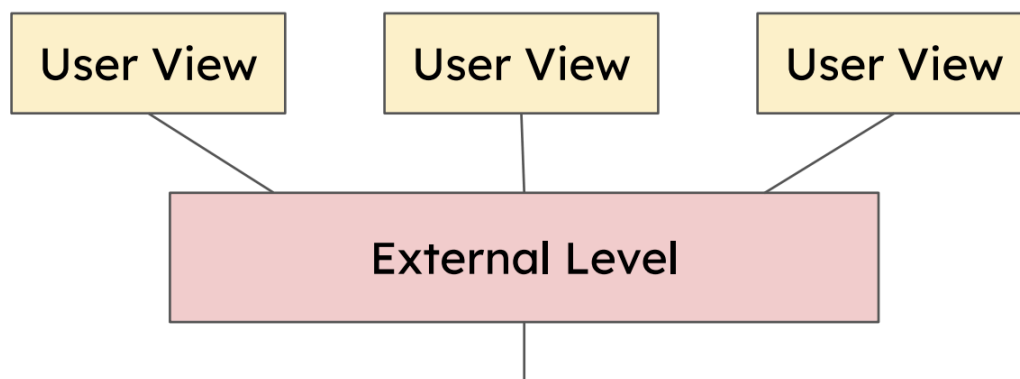


The design of a three-tier client/server system

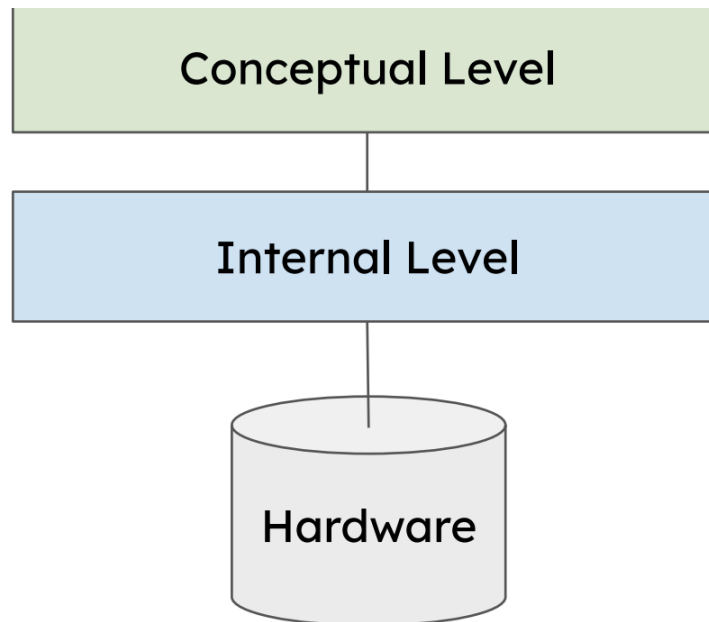
Key Concepts in Relational Databases

- **Tables** (Relations): Represents a collection of related data, arranged in rows (records).
- **Attributes** (Columns): Defines a property of the data (e.g., student name, age, etc.).
- **Tuples** (Rows): Each row represents a single record in the table.
- **Primary Key**: A unique identifier for each record (e.g., a student ID).
- **Foreign Key**: A reference to a primary key in another table, used to link tables.

Database Architecture



ds) and columns (fields).



1. **Internal Level:** The lowest level of the architecture, concerned with how data is physically stored.
2. **Conceptual Level:** The middle level, dealing with how data is logically structured, or the logical data model.
3. **External Level:** This is what end users interact with. It provides multiple user views of the data.

Entity-Relationship (ER) Model

- Entity: Any object, real or abstract, about which data is stored (e.g., customer, product).
- Attributes: Characteristics that describe an entity (e.g., a student's name, ID, or age).
- Primary Key: A unique identifier for each entity instance.
- Relationships: Defines how entities relate to one another.

Client/Server Architecture

- Client: The front-end application or browser interface interacting with the user.
- Server: Handles requests from the client and communicates with the database.
- Middleware: Acts as a bridge between the client and server, often managing business logic.

Examples of Database Use

physically stored.

Defining what data is stored and the relationships between them.

s of the same data, ensuring that different users can access and work with the data relevant

uct, event).

e).

ess logic and database queries.

to them.

- Organizational Information Systems: Employee directories, payroll systems.
- Booking & Scheduling: Airline ticket bookings, university course sign-ups.
- E-commerce: Online stores like Amazon or eBay, where product inventory is managed.
- Web Automation: Websites like Google that store and retrieve massive amounts of data.

ged via a database.

from databases to provide search results and other services.

