**Unit 1 Cheatsheet: Introduction to DBMS**

**1. What is DBMS?**

A **Database Management System (DBMS)** is software designed to store, manage, and manipulate data in a structured way. It helps in creating, retrieving, updating, and managing databases, ensuring that data is easily accessible, secure, and consistent.

* **Examples:**
  + **Oracle DBMS**: A powerful, enterprise-level relational DBMS.
  + **MySQL**: An open-source RDBMS that is widely used in web applications.
  + **PostgreSQL**: A highly extensible open-source RDBMS known for advanced features.

**2. DBMS vs. File Systems**

A file system organizes data into files, but lacks the features of a DBMS, such as relationships between data, consistency, and scalability. Here's the difference:

| **Aspect** | **File System** | **DBMS** |
| --- | --- | --- |
| **Data Redundancy** | High. Same data may be stored in multiple places. | Low. Ensures single, consistent data storage. |
| **Security** | Basic access control. | Advanced access control with roles and permissions. |
| **Concurrency** | Limited. Multiple users can’t easily access the same file simultaneously. | High. Allows multiple users to access and modify data simultaneously (with control). |
| **Backup & Recovery** | Manual. Backups need to be done manually and may be incomplete. | Automatic. Can be scheduled with recovery options. |

**3. Key Concepts**

* **Data Model:** A **Data Model** defines the logical structure of the data in the database. It determines how data is stored, accessed, and manipulated.
  + **Examples:**
    - **Relational Model** (used by RDBMS like MySQL): Data stored in tables with rows and columns.
    - **Document Model** (used by NoSQL databases like MongoDB): Data stored as documents (JSON, BSON).
* **Schema:** A **Schema** is the structure or blueprint that describes the organization of data in the database. It defines the tables, fields, relationships, and other elements in the system.
  + **Examples:**
    - **SalesDB Schema**: Might include tables like Customers, Orders, Products.
    - **StudentDB Schema**: Could contain tables like Students, Courses, Grades.
* **Instance:** An **Instance** refers to the actual data stored in the database at any given moment. It is the data present in the database that complies with the schema.
  + **Example:** If the Students table in a schema has 100 rows of student data today, that’s the current instance of the Students table.

**4. Three-Schema Architecture**

The **Three-Schema Architecture** separates the database system into three layers, each focusing on different concerns to provide abstraction, flexibility, and independence.

1. **External Level (User View):**
   * Represents how users interact with the data (what they see). It hides the complexity of the underlying database structure.
   * **Example:** A user might only see student names and IDs from the Students table but not the full underlying table structure.
2. **Conceptual Level (Logical Level):**
   * Represents the logical structure of the entire database, abstracting away physical details.
   * **Example:** This includes the relationships between the Students, Courses, and Grades tables, showing how they connect without worrying about how the data is stored on disk.
3. **Internal Level (Physical Level):**
   * Describes the physical storage of data, such as how it is stored in files and indexed for efficiency.
   * **Example:** It defines the specifics of how data is stored in memory, like whether the Students table is stored in B-trees or hash-based indexes.

**5. DBMS Environment and Architectures**

* **Centralized Architecture:**
  + All database management is done by a single machine/server that handles storage, retrieval, and access of data.
  + **Example:** A local library system where all data is stored and accessed from a single computer.
* **Client-Server Architecture:**
  + The database is managed by a server, but users interact with the system via client applications (programs or web interfaces).
  + **Example:** A web-based application where users access data from a central server, like a shopping website accessing product data.

**6. Classification of DBMS**

DBMS can be classified based on their data model and use cases.

* **Relational DBMS (RDBMS):**
  + Uses tables (relations) to represent data and relationships.
  + **Examples:**
    - **Oracle DBMS**
    - **MySQL**
    - **PostgreSQL**
    - **SQL Server**
* **NoSQL DBMS:**
  + Used for unstructured data, offering flexibility in data storage (document-based, key-value pairs, etc.).
  + **Examples:**
    - **MongoDB** (Document-based)
    - **Cassandra** (Column-family model)
    - **Redis** (Key-Value store)
* **Hierarchical DBMS:**
  + Data is organized in a tree-like structure (parent-child relationships).
  + **Examples:**
    - **IBM IMS**
* **Network DBMS:**
  + Uses a graph-like structure to represent data relationships (many-to-many).
  + **Examples:**
    - **Integrated Data Store (IDS)**