**## DBMS Fundamentals: Week 1 Notes**

**1. Why Use a Database? (The Library vs. a Pile of Books)**

A **File System** is like a pile of books in a room. It's easy to start, but finding information, avoiding contradictions, and managing multiple users is chaotic.

A **DBMS (Database Management System)** is like an organized **Library** with a librarian (the DBMS) who manages everything systematically. The only real advantage of a file system is its **initial ease of setup**. For everything else, a DBMS is superior.s



| Feature | File System (Pile of Books) 📚 | DBMS (Organized Library) 🏛️ |
| --- | --- | --- |
| **Data Consistency** | **Low**. Updating one copy of a file leads to inconsistency. | **High**. Central control ensures data is consistent. |
| **Data Integrity** | **None**. The system doesn't check if data is valid (e.g., age = -5). | **High**. You can set rules (constraints) to ensure data is valid. |
| **Concurrency** | **Difficult**. Multiple users editing at once can corrupt data. | **Managed**. The DBMS ensures simultaneous actions don't cause errors. |

Imagine you are transferring $50 from your **Savings Account** to your **Checking Account**.

**## Data Redundancy 複製**

**Data Redundancy** means storing the same piece of data in multiple locations unnecessarily.

* **What it is:** Imagine your bank stores your home address in three different systems: one for statements, one for marketing, and one for your account profile. If you move and only update one system, the other two now have incorrect, outdated information. This is data redundancy leading to data inconsistency.
* **Why it's bad:** It wastes storage space and, more importantly, creates a high risk of **data inconsistency**. Databases aim to have a "single source of truth" to avoid this problem.

**## The ACID Properties: The Rules of a Reliable Transaction**

ACID is an acronym that stands for **Atomicity, Consistency, Isolation, and Durability**. It's a set of guarantees ensuring that a database transaction (like our bank transfer) is processed reliably.

**### ⚛️ A - Atomicity (All or Nothing)**

This property guarantees that a transaction is treated as a single, indivisible unit of work. Either all the steps in the transaction are completed successfully, or none of them are.

* **Analogy:** Your bank transfer involves two steps:
  1. Debit $50 from your Savings Account.
  2. Credit $50 to your Checking Account.
* If the system crashes after step 1 but before step 2, **atomicity** ensures the entire transaction is cancelled ("rolled back"). The $50 debit from your savings is reversed, so your money isn't lost in cyberspace. The transaction cannot be left half-finished.

**### ✅ C - Consistency (The Rule Follower)**

This property ensures that a transaction can only bring the database from one valid state to another. It prevents any data from being written that would violate the database's rules and constraints.

* **Analogy:** Let's say your bank has a rule that no account balance can go below $0. You have $20 in your Savings and try to transfer $50.
* The transaction starts in a valid state (balances are >= $0). If it were allowed to proceed, it would end in an invalid state (Savings = -$30). **Consistency** prevents this by stopping the transaction before it can break the rules, ensuring the database remains valid.

**### izolasyon I - Isolation (Working Alone)**

This property ensures that concurrently executing transactions do not interfere with each other. It makes it seem as though transactions are running one after the other, even if they're happening at the same time.

* **Analogy:** While your $50 transfer is in progress (the money has left Savings but hasn't yet arrived in Checking), your banking app tries to calculate your total net worth by adding all account balances.
* Without **isolation**, the app might read the balances in that messy, intermediate state and calculate your net worth as being $50 less than it actually is. **Isolation** prevents this. It makes the net worth calculation wait until your transfer is fully complete, ensuring it reads a consistent and valid snapshot of your data.

**### 💾 D - Durability (Written in Permanent Marker)**

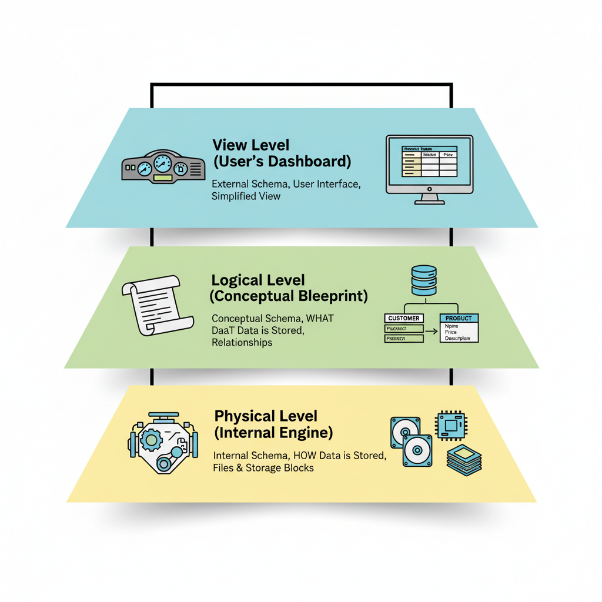
This property guarantees that once a transaction has been successfully completed ("committed"), it will remain committed, even in the event of a system failure like a power outage or crash.

* **Analogy:** Your $50 transfer is successful, and you see the updated balances on your screen.
* A second later, the bank's server loses power. **Durability** guarantees that the result of your transaction is already saved to a permanent log or the disk. When the server reboots, your transfer will still be complete. The change is permanent and survives failures.

**2. The 3 Levels of Data Abstraction (How to View a Car)**

A DBMS hides technical complexity through three layers:

1. **Physical Level (The Engine Room) ⚙️**
   * The **lowest level**. Describes **HOW** data is physically stored on the disk. This is the most complex layer and is managed by the DBMS itself.
2. **Logical Level (The Blueprint) 📝**
   * The **middle level**. Describes **WHAT** data is stored and the relationships between tables. This is the overall database design. Deciding a Product has a price is a logical-level decision.
3. **View Level (The Dashboard) 🚗**
   * The **highest level**. This is what the end-user sees—a simplified, customized slice of the database for a specific task.

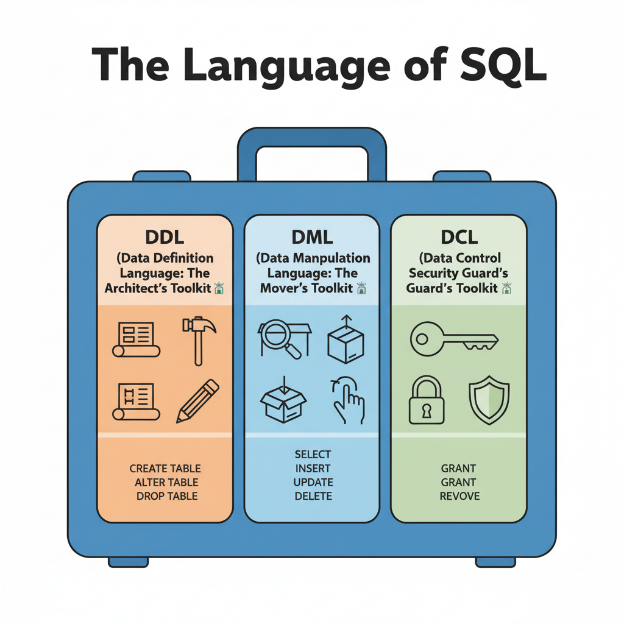


**3. Data Independence (The Freedom to Change)**

This powerful feature means you can change a lower level without breaking the levels above it.

* **Physical Data Independence:** Change the physical storage (e.g., move to a new server) without affecting the logical structure or applications.
* **Logical Data Independence:** Change the logical structure (e.g., add a new column to a table) without breaking existing user views or applications.

**4. The Language of SQL (The Three Toolkits)**



1. **DDL (Data Definition Language): The Architect's Toolkit 🏗️**
   * **Purpose:** To define and manage the database structure (the schema). These commands modify the **Data Dictionary**.
   * **Commands:** CREATE, ALTER, DROP.
2. **DML (Data Manipulation Language): The Mover's Toolkit 📦**
   * **Purpose:** To interact with the data *inside* the tables.
   * **Commands:** SELECT, INSERT, UPDATE, DELETE.
3. **DCL (Data Control Language): The Security Guard's Toolkit 🔑**
   * **Purpose:** To manage user permissions and access.
   * **Commands:** GRANT, REVOKE.

**Week One All Questions and Answer and explanations:**

Here are the answers to your questions.

1. **Which of the following is not a drawback of file systems when compared to DBMS?**
   * **Correct Answer:** Ease of initial setup
   * **Explanation:** File systems are generally simpler and quicker to set up for basic data storage needs, which is an advantage, not a drawback. The other options (inconsistent data, lack of data integrity, and difficulty with concurrency) are all significant drawbacks of file systems.
2. **Which of the following creates and maintains the schema of a database?**
   * **Correct Answer:** Data Definition Language (DDL)
   * **Explanation:** DDL commands like CREATE, ALTER, and DROP are used to define, modify, and remove database objects that form the schema.
3. **Which of the following describes the concept that any change made to the physical schema should not affect the logical level of the DBMS?**
   * **Correct Answer:** Physical Data Independence
   * **Explanation:** This concept ensures that you can change how the data is physically stored (e.g., change the file structure or storage device) without having to change the logical schema that defines the data and its relationships.
4. **Which of the following components of DBMS interacts with the file manager of the operating system?**
   * **Correct Answer:** Storage manager
   * **Explanation:** The storage manager is the component responsible for the interaction between the data stored in the database and the operating system's file system. It handles the physical storage and retrieval of data.
5. **Suppose a company wants to determine whether the 'price' of the commodity will be an attribute in the shopping database or not. Which of the following holds good about this?**
   * **Correct Answer:** This is a logical level decision.
   * **Explanation:** The logical level (or conceptual level) deals with defining what data is stored in the database and what relationships exist among that data. Deciding that a 'commodity' has a 'price' attribute is a fundamental part of this design.
6. **Consider the given statements.**
   * **Statement 1:** DBMS provides an efficient platform for doing complex arithmetic computation on the data.
   * **Statement 2:** It is easier to create access rules in a file system than in a DBMS.
   * **Correct Answer:** Both statements are wrong
   * **Explanation:** DBMSs are optimized for data storage and retrieval, not complex computations (which are better handled in application code). Furthermore, DBMSs have sophisticated, fine-grained access control mechanisms that are far more powerful and easier to manage than the basic permissions of a file system.
7. **Which type of SQL commands can lead to modification in the Data Dictionary?**
   * **Correct Answer:** Data Definition Language.
   * **Explanation:** The Data Dictionary stores metadata (data about data), which includes the database schema. Since DDL commands (CREATE, ALTER, etc.) modify the schema, they directly modify the Data Dictionary.
8. **Which component of DBMS maintains the consistency of a database when multiple transactions are executed simultaneously on the data?**
   * **Correct Answer:** Concurrency Control Manager
   * **Explanation:** This component is part of the transaction manager and is specifically responsible for managing the interleaving of operations from multiple simultaneous transactions to ensure the database remains in a consistent state.
9. **Storing multiple copies of the same data within the system is not advisable, because it increases...**
   * **Correct Answer:** Data Redundancy
   * **Explanation:** Data redundancy is the unnecessary duplication of data. It is discouraged because it can lead to data inconsistency, where updates to one copy might not be reflected in others.
10. **The lowest level of data abstraction is...**
    * **Correct Answer:** physical level
    * **Explanation:** The physical level is the lowest level of abstraction and describes *how* the data is actually stored on the storage media. It is the most complex level, with details that are hidden from higher levels (logical and view).