

AMITY INSTITUTE OF INFORMATION TECHNOLOGY



SYSTEM SOFTWARE

ASSIGNMENT

SUBMITTED BY: -

ANSH ANAND

BCA

SEM-5

A21704819011

Q1. What is DBMS? Explain its Basic Concepts? What are the levels of data description? What are the uses of DBMS?

Ans.

DBMS software primarily functions as an interface between the end user and the database, simultaneously managing the data, the database engine, and the database schema in order to facilitate the organization and manipulation of data.

Though functions of DBMS vary greatly, general-purpose DBMS features and capabilities should include: a user accessible catalogue describing metadata, DBMS library management system, data abstraction and independence, data security, logging and auditing of activity, support for concurrency and transactions, support for authorization of access, access support from remote locations, DBMS data recovery support in the event of damage, and enforcement of constraints to ensure the data follows certain rules.

A database schema design technique that functions to increase clarity in organizing data is referred to as normalization. Normalization in DBMS modifies an existing schema to minimize redundancy and dependency of data by splitting a large table into smaller tables and defining the relationship between them. DBMS Output is a built-in package SQL in DBMS that enables the user to display debugging information and output, and send messages from subprograms, packages, PL/SQL blocks, and triggers. Oracle originally developed the DBMS File Transfer package, which provides procedures to copy a binary file within a database or to transfer a binary file between databases.

A database management system functions through the use of system commands, first receiving instructions from a database administrator in DBMS, then instructing the system accordingly, either

to retrieve data, modify data, or load existing data from the system. Popular DBMS examples include cloud-based database management systems, in-memory database management systems (IMDBMS), columnar database management systems (CDBMS), and NoSQL in DBMS.

Here are some Concepts of DBMS

1. ACID Properties:

A — Atomicity

C — Consistency

I — Isolation

D — Durability

“A transaction is a very small unit of a program”

Here, the transaction means a **group of tasks**, which means it consists of multiple tasks.

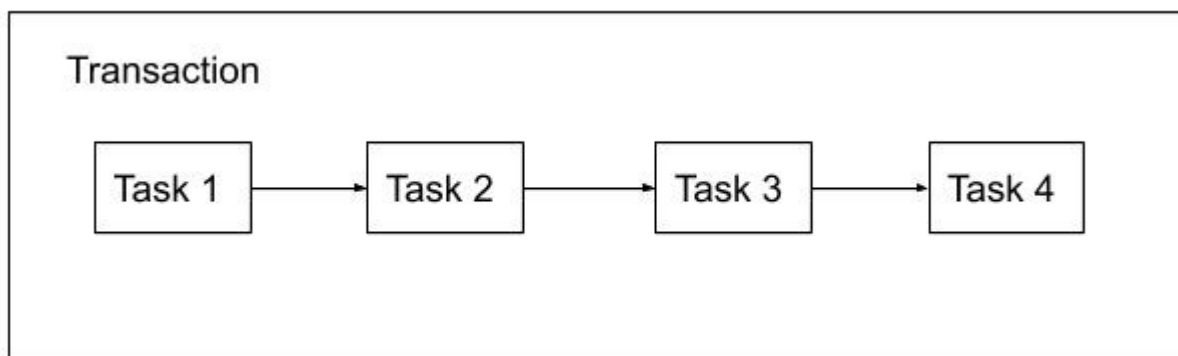


Fig-a

Once all of these tasks are completed, our transaction is complete. It could be any set of flow. A transaction is supposed to have these four properties.

Let's look into it one by one,

Atomicity

It means either an entire transaction takes place at once or nothing. Consider **Fig-a**, If you complete task 1 and task 2, suddenly some system error occurs, you cannot move further to the next task.

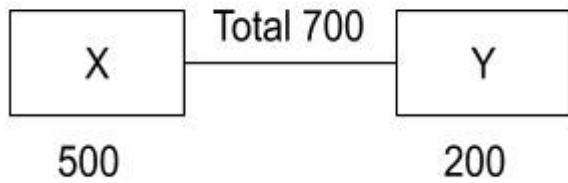
Now, what atomicity says that either you can complete task 3 and task 4 or you can **rollback**. You cannot be in the midway of any transactions.

Here, what rollback means go back to the zero state

Consistency

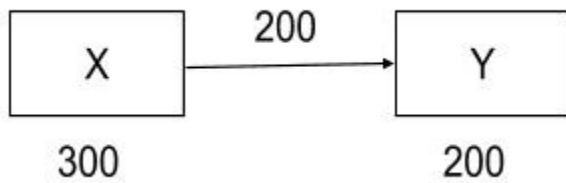
Let's take a quick example,

You had ₹500 in X person's account and you had ₹200 in Y person's account. So the total was ₹700.



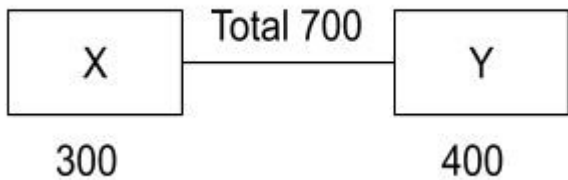
Step-1

Let's say, you do a transaction for transferring ₹200 from X to Y.



Step-2

Now, the Consistency means “**Total amount before transaction**” is equal to the “**Total amount after transaction**”.



Step-3

Isolation

Isolation means the concurrent execution of transactions.

$X = X - 100 \rightarrow \text{TRANS (1)}$

$Y = Y - 50 \rightarrow \text{TRANS (2)}$

$Z = X + Y \rightarrow \text{TRANS (3)}$

Fig-b

Here, TRANS means transaction

You can execute TRANS (1) parallelly you can execute TRANS (2) and then only you can go to the TRANS (3) that is what we refer to as Isolating transactions from one another.

Durability

It simply says that even on system failure, changes that you do to the database at *persistent*.

2. Types Of SQL Statements:

It is classified into four categories,

DDL (Data Definition Language)

DML (Data Manipulation Language)

DQL (Data Query Language)

DCL (Data Control Language)

DDL

It means to define your database or create your database.

Example: CREATE, ALTER, DROP.

DML

It means anything you want to change or update your database.

Example: INSERT, UPDATE, DELETE.

DQL

It means query your database or you want to select something from your database.

Example: SELECT.

DCL

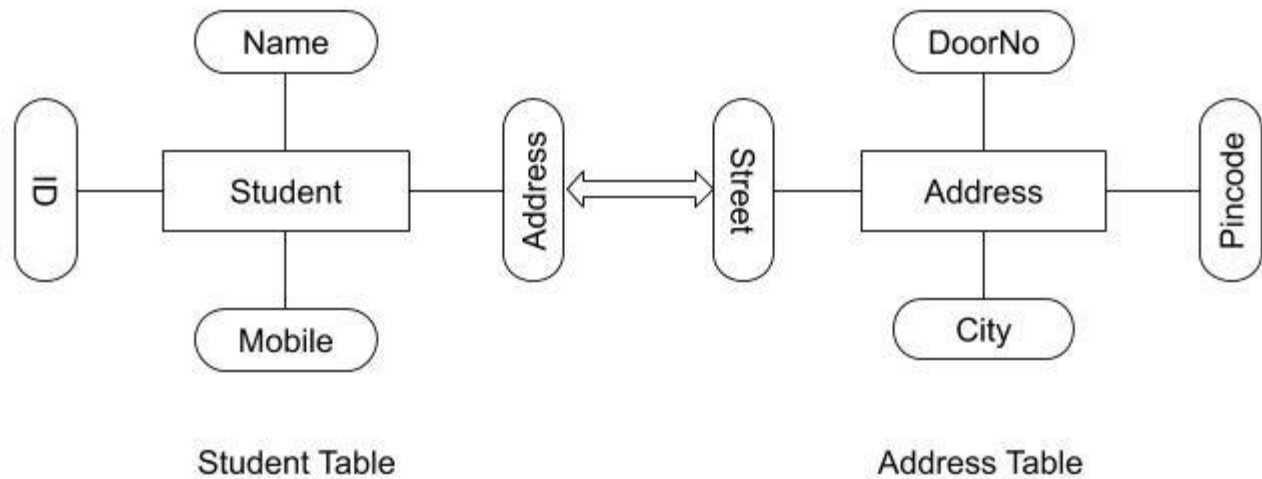
It means to control your database. For example, you want to remove one user from your database or you want to add one user or you want to protect your database from others simply like admin console.

Example: GRANT, REVOKE, ALTER PASSWORD.

3. ER Model:

ER Model defines the conceptual view of the database simply says “**diagrammatical view of database**”. ER diagrams comprise of,

Entity Set
Relationship Set



ER Diagram

Entity and Entity Set

An **entity** is an object and it can be a person, place, or company. The **entity set** means a set of entities of the same type that can share the same properties. Example: All persons having account in the bank.

Relationship and Relationship Set

The association among entities is called **relationship**. Example: An employee works at a department, A student enrolls in a course. A set of relationships of similar types is called a **relationship set**. Example: Association between patient and doctor, Association between teacher and student.

Types of Relationship set

- One to One (Ex: One student have one roll number)
- One to Many (Ex: A doctor have Multiple Patients)
- Many to One (Ex: Many products from one store)
- Many to Many (Ex: Many products from many stores)

4. Normalization:

If a database design is not perfect, that means it may contain some **anomalies**(errors). **Normalization** is a method to remove all anomalies and brings the database to a consistent state.

- Minimizing redundancy.

- Minimizing insertion, deletion, and update anomalies.

Types of normalization

1NF (First Normal Form)

2NF (Second Normal Form)

3NF (Third Normal Form)

BCNF (Boyce-Codd Normal Form)

4NF (Fourth Normal Form)

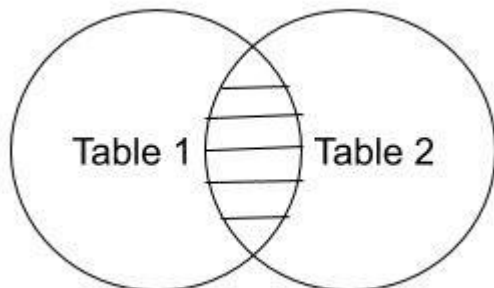
5NF (Fifth Normal Form)

5. SQL Joins:

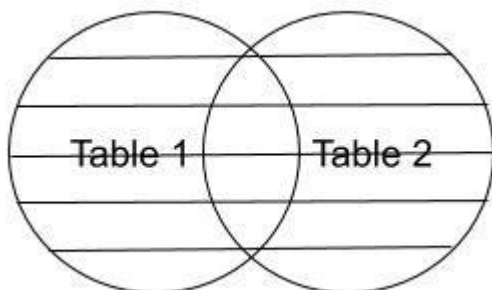
What joins simply do means, it combines rows from **two or more tables** based on the related columns between them.

Types of SQL Joins

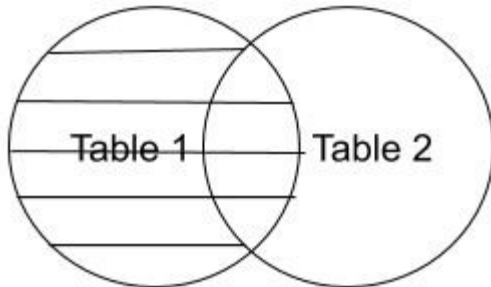
- **Inner Join**
 - It returns the common values in both tables



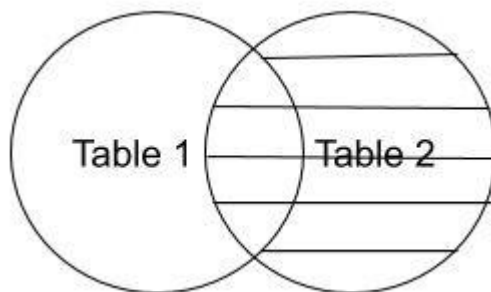
- **Outer Join**
 - It returns all records when there is a match in either left or the right table.



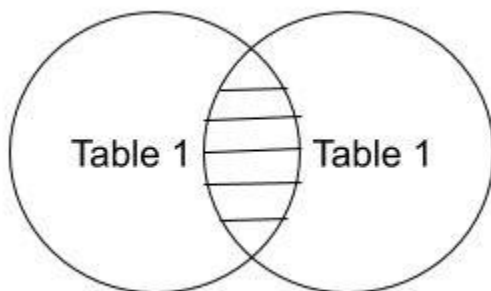
- **Left Outer Join**
— It returns all records from the left table and common records from the right table.



- **Right Outer Join**
— It returns all records from the right table and common records from the left table.



- **Self Join**
— It means to join a table to itself.



6. SQL Constraints:

SQL Constraints are rules that are used to limit the type of data that can go into the table, to maintain the integrity and accuracy of the data inside the table.

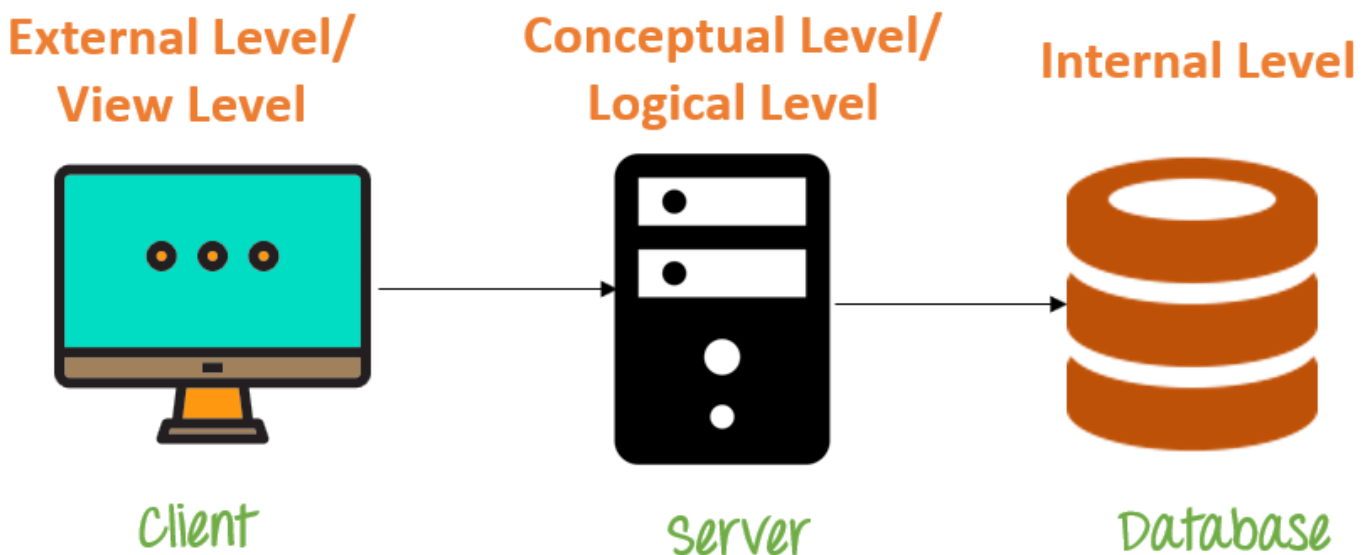
Example: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, DEFAULT.

Levels of Data Description

Database systems comprise of complex data structures. Thus, to make the system efficient for retrieval of data and reduce the complexity of the users, developers use the method of Data Abstraction.

There are mainly three levels of data abstraction:

1. Internal Level: Actual PHYSICAL storage structure and access paths.
2. Conceptual or Logical Level: Structure and constraints for the entire database
3. External or View level: Describes various user views



Internal Level/Schema

The internal schema defines the physical storage structure of the database. The internal schema is a very low-level representation of the entire database. It contains multiple occurrences of multiple types of internal record. In the ANSI term, it is also called “stored record”.

Facts about Internal schema:

- The internal schema is the lowest level of data abstraction
- It helps you to keep information about the actual representation of the entire database. Like the actual storage of the data on the disk in the form of records
- The internal view tells us what data is stored in the database and how

- It never deals with the physical devices. Instead, internal schema views a physical device as a collection of physical pages

Conceptual Schema/Level

The conceptual schema describes the Database structure of the whole database for the community of users. This schema hides information about the physical storage structures and focuses on describing data types, entities, relationships, etc.

This logical level comes between the user level and physical storage view. However, there is only single conceptual view of a single database.

Facts about Conceptual schema:

- Defines all database entities, their attributes, and their relationships
- Security and integrity information
- In the conceptual level, the data available to a user must be contained in or derivable from the physical level

External Schema/Level

An external schema describes the part of the database which specific user is interested in. It hides the unrelated details of the database from the user. There may be “n” number of external views for each database.

Each external view is defined using an external schema, which consists of definitions of various types of external record of that specific view.

An external view is just the content of the database as it is seen by some specific particular user. For example, a user from the sales department will see only sales related data.

Facts about external schema:

- An external level is only related to the data which is viewed by specific end users.
- This level includes some external schemas.
- External schema level is nearest to the user

- The external schema describes the segment of the database which is needed for a certain user group and hides the remaining details from the database from the specific user group

Uses of DBMS

- **Simplified Data Sharing**

One of the main advantages of DBMS is that it allows both onsite and remote users to easily share the data by following the correct authorization protocols. It provides operators access to well-managed data. As a result, they can rapidly respond to variations in the environment.

The purpose of using database management software is that you can yield speedy responses to impromptu queries as the data is properly managed and up-to-date. In case of any ad hoc query, the database software returns a response (known as the query result set) to the application.

- **Enhanced Data Safety**

The threats of data security breaches become more pronounced when several users access the database. A database management software offers better implementation of data confidentiality and safety guidelines through controlled user access.

- **Improved Data Integration**

A simple database management system facilitates an integrated view of the company's data. The company can quickly see how activities in one division of the organization influence other divisions.

- **Better Decision-Making**

The role of DBMS in businesses is to provide access to well-managed data, making it possible for users to make accurate and timely decisions. Data organization software offers a streamlined framework to enable data quality initiatives, improving data management procedures and yielding better-quality information.

- **Improved Efficiency**

Streamlined data access software, along with the tools that convert data into valuable information, enables operators to make swift, knowledgeable decisions. This improves a database's performance and efficiency.

- **Data Integration**

Instead of storing data in isolated database systems, a single interface can be used to manage databases with physical and logical relationships.