DOMS | Page No.
Date / /

Database and Database Management System :

1. What is database? and its advantages.

Ans A database is a collection of related information stored, so that it is available to many users for different purpose.
In other words a database is an organized collection of structured information or data, typically stored electronically in a computer system.

Advantages of database:

- Minimum data redundancy
- Improved data security
- Lower updating errors
- Increased consistency
- Reduced costs of data entry, data storage and data retrieval.
- Improved data access using host and query languages.
- Higher data integrity from application programs.

2. Introduction of DBMS.

Ans. A database management system is a collection of interrelated data and a set of programs to access those data.
- DBMS is used to organize the data in the form of table, schema, view and report etc.
- The primary goal of a DBMS is to

provide a way to store and retrieve database info that is both convenient and efficient.

- The software which is used to manage the database is called database management system. It provides us with an interface or a tool to perform various operations like:

1. Creating the database.
2. Manipulating the database.
3. Storing and retrieving the data from the database
4. Deleting data from the database etc.

Metadata: Metadata means "data about data". Metadata is defined as the data providing information about one or more aspects of the data, it is used to summarize basic information about data that can make tracking and working with specific data easier.

For example, a digital image's metadata can include information related to the image's resolution, size, colour depth, and time of creation. This information can be used for data classification, labeling, organization, sorting, tracking, searching and analysis.

Data Abstraction: It is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data.

The database system consist of complicated data structures and relations. For users to access the data easily, these complications are kept hidden, and only the relevant part of the database is made accessible to the users through data abstraction.

Levels of abstraction for DBMS

Database Systems include complex data-structures. In terms of retrieval of data, reduce complexity in terms of usability of users and in order to make the system efficient, developers use levels of abstraction that hide irrelevant details from the users. Levels of abstraction simplify database design.

Mainly there are three levels of abstraction for DBMS, which are as follows —

- Physical or Internal Level
- Logical or Conceptual Level
- View or External Level

Physical or Internal level

It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database. It is a very complex level to understand. For example, customer's information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.

Logical or Conceptual Level

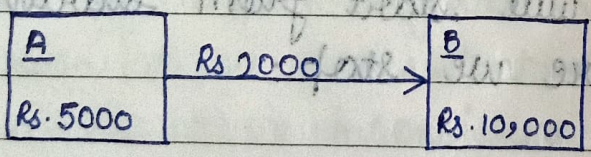
Logical level is the intermediate level or next higher level. It describes what data is stored in the database and what relationship exists among those data. It tries to describe the entire or whole data because it describes what tables to be created and what are the links among those tables that are created. It is less complex than the physical level, logical level is used by developers or database administrators (DBA). So, overall, the logical level contains tables (feilds and attributes) and relationships among table attributes.

View or External Level

It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database. View level can be used by all users (all levels' users). This level is the least complex and easy to understand.
For example, a user can interact with a system using GUI that is view level and can enter details at GUI or screen and the user does not know how data is stored and what data is stored, this detail is hidden from the user.

Transaction: A transaction is a single unit of logic or work, sometimes made up of multiple operations. Any logical calculations done in a consistent mode in a database is known as a transaction.

e.g.



3 Step Process:

1. Check if A has Rs 2000 to transfer?
2. Then deduct Rs 2000 from A's account
3. And add Rs 2000 to B's account.

DOMS | Page No.
Date / /

Properties of Transaction :-

A = Atomicity → The entire transaction takes place at once or doesn't happen at all.

C = Consistency → Database must be consistent before & after the transaction.

I = Isolation → Multiple transaction occur independently without interference.

D = Durability → The changes of a successful transaction occur even if the system failure occurs.

Atomicity : Either all or null.
It states that all operations of the transaction takes place at onces if not, the transaction is aborted.
If transaction executed till half, then after it will start from starting. Not from half where we stop.

Abort : J half ch shd diti tan starting ton start krni pavgi / Roll back.

Commit : J transaction end tkk hogi finish tan changes update hojange.

Example :- Transaction T - T_1 , T_2

Before A - 600 , B - 300

$A \xrightarrow{100} B$

T_1	T_2
read (A)	read (B)
A = A - 100	B = B + 100
Write (A)	Write (B)
A = 500	B = 400

If any transaction fail, then the whole rollback.

A = 600 , B = 300

Consistency : The integrity constraints are maintained so that the database is consistent before and after transaction.

example :

A = 600 B = 300 y 900

T_1	T_2
read (A)	read (B)
A = A - 100	B = B + 100
Write (A)	Write (B)
A = 500	B = 400 y 900

If Sum of initial & final account are same then it is consistent data. If not same, then it is inconsistent state.

Isolation: It shows that the data which is used at the time of execution of a transaction cannot be used by the second

transaction until the first one is completed.

Example: T_1 T_2 T_3 T_4
A B C D E F G H
When one transaction complete then second perform.

Durability: Duration of time. Within the duration transaction get completed and get updated. It presist (Rehana chaeda) in the software or the database. Recovery Management component takes care of durability.

Concurrent: Multiple users can access and use the same database at one time, which is known as the concurrent execution of the database.

Data Integration: Data integration refers to the process of bringing together data from multiple sources across an organization to provide a complete, accurate and up-to-date dataset business intelligence, data analysis and other application and business process to help

* Integrity: The quality of being honest and accurate.

* Types of users: Database users can be categorized based on their interaction with the database. We can categorize them into 8 categories.

Who are database users: Any person who used a database and avails benefits from the database is known as database user in DBMS. Database users can access the database and retrieve the data from the database using applications and interfaces provided by DBMS. Interact with software at diff. levels like view, External, Conceptual level.

Database Users:

1. Database Administrator (DBA): DBA is a person/team who defines the schema for the database. The DBA will then create a new account id and password for the user if he/she need to access the database. DBA is also responsible for providing security to the database and he allows only the authorized users to access/modify the data base. DBA is responsible for the

problems such as security breaches and poor system response time.
- DBA also monitors the recovery and backup and provide technical support.
- The DBA has a DBA account in the DBMS which called a system or superuser account.
- DBA repairs damage caused due to hardware and/or software failures.
- DBA is one having privileges to perform DCL (Data Control Language) operations such as GRANT and REVOKE, to allow/restrict a particular user from accessing the database.

2. Naive/Parametric end Users: Parametric end Users are the unsophisticated who don't have any DBMS knowledge but they frequently use the database applications in their daily life to get the desired results. For example, Railway's ticket booking users are naive users. Clerks in any bank is a naive user because they don't have any DBMS knowledge but they still use the database and perform their given task.

3. System Analyst: System analyst is a user who analyzes the requirement of naive

end users. They check whether all the requirements of end users are satisfied.

4. Sophisticated User: These can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own database applications according to their requirements. They don't write the program code but they interact the database by writing SQL queries directly through the query processor.

5. Database Designers: Database designers are the users who design the structure of database which includes tables, indexes, views, triggers, stored procedures and constraints which are usually enforced before the database is created or populated with data. He/she controls what data must be stored and how the data items to be related. It is responsibility of database designers to understand the requirements of different user groups and then create a design which satisfies the need of all the user groups.

6. Application Programmers: Application Programmers also referred as System Analysts or simply

Software engineers, are the back-end programmers who writes the code for the application programs. They are the Computer professionals. These programs could be written in programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc. Application programmers design, debug, test and maintain set of programs called "canned transactions" for the Naive (Parametric) users in order to interact with database.

7. Casual Users/Temporary Users: Casual users are the users who occasionally use/access the database but each time when they access the database they require the new information, for example, Middle or higher level manager.

8. Specialized Users: Specialized users are sophisticated users who write specialized database application that does not fit into the traditional data processing frame work. Among these applications are computer aided-design Systems, Knowledge-base and Expert Systems etc.

⇨ Database Schema: A database schema defines how data is organized within a relational database. It provides a logical view of

database. This is inclusive of logical constraints such as table name, fields, data types and the relationships between these entities. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.
Schema is a structure or skelton.

Types of Schema:
Physical Schema, Logical Schema, View.

- Physical Database Schema: It defines how the data will be stored in a secondary storage.
- Logical Database Schema: This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views and integrity constraints.

Eg: Create table Student (AUID Int, Name VARCHAR(30), MobileNo Int, AS UID, AS M-N)

This is logical Schema

(Show as physical)
This is view for the user

It is constraint (Limit that name is not bigger than 30 alphabets)