

XII - CHAPTER - 8

DataBase Management System (DBMS)

1. Database : A database is a collection of data.

DBMS : is a software used for storage, retrieval and management of databases.

Eg : MySQL, Oracle, SQLite , DB-2

2. Advantages of DBMS

- 2.1 **Controlling Data Redundancy :** DBMS reduces data redundancy.
- 2.2 **Data consistency :** By controlling redundancy , inconsistency is also controlled.
- 2.3 **Data integrity :** It ensures the data in the database is correct.
- 2.4 **Data security :** It ensures data security and privacy.
- 2.4 **Sharing of data :** Data can be shared among several different users.
- 2.5 **Efficient data access :** Data efficiently store and retrieve
- 2.6 **Crash recovery :** It provides a mechanism to recover data from crashes.
- 2.7 **Enforcement of standards :** It ensures that all the data follow certain standards.

3. Components of DBMS :

- 3.1 **Hardware :** Devices that are used for keeping and retrieval of data.
- 3.2 **Software :** DBMS acts as an interface between user and database.
- 3.3 **Data :** Important component of DBMS.
- 3.4 **Procedures :** It refers to instructions and rules that perform a specific task.
- 3.5 **Users :** There are four type of users of database :-

3.5 Four types of users of the database.

3.5.1 Database Administrator (DBA) : Fully controls the database.

Duties of database administrator

3.5.1.1 Design of the conceptual and physical schemas

3.5.1.2 Security and authorisation

3.5.1.3 Data availability and recovery from failures

3.5.2 Application programmer : Application developers who interact with the DBMS.

3.5.3 Sophisticated users: This includes engineers, scientists, and business analytics

3.5.4 Naive users : They are not concerned with or even aware of the details of the DBMS.

4. Data abstraction : It is a process of hiding unwanted or irrelevant details from the end user

Three levels of abstraction for DBMS

4.1 **Physical Level :** It defines how the data is actually stored.

4.2 **Logical level :** It describes what data is stored in the database.

4.3 **View level :** It only defines a part of the entire data.

5. Data independence : To modify the schema without affecting programs and applications.

5.1 **Physical data independence :** Changing physical level without affecting logical level.

5.2 **Logical data independence :** changing logical level without affecting view level.

6. RDBMS : Relational DataBase Management System

Eg : Oracle ,Microsoft SQL Server, MySQL, DB2 ,Informix and Ingress

7. Terminologies in RDBMS (RDBMS ലെ സാങ്കേതികപദങ്ങൾ)

7.1 **Entity** : A person or a thing in the real world that is distinguishable from others.

7.2 **Relation** (Table) : collection of data organized in terms of rows and columns.

7.3 **Tuple** : The rows(records) of a relation

7.4 **Attribute** : The columns of a relation.

7.5 **Degree** : The number of attributes or columns in a relation (table).

7.6 **Cardinality** : The number of rows or tuples in a relation.

7.7 **Domain** : Pool of values from which actual values appear in a given column.

7.8 **Schema** : Structure of a database.

7.9 **Instance**: A set of tuples in which each tuple has the same number of fields as the relational schema.

8. KEY used in RDBMS : A key is an attribute or collection of attributes that uniquely identifies each record (tuple) in a table.

8.1 **Candidate Key** : Minimal set of attributes that uniquely identifies a row in a relation

8.2 **Primary key** : one of the candidate keys that identify tuples within the relation.

8.3 **Alternate key** : A candidate key that is not the primary key.

8.4 **Foreign key** : A key in a table can be called foreign key if it is a primary key in another table.

9.Relational operations

Operation	Symbol	Use
SELECT	σ	Select rows(tuples) from a relation that satisfies a given condition
PROJECT	π	Select certain attributes from the table and form a new relation.
UNION	\cup	It Returns a relation containing all tuples appearing in either or both of the two specified relations
INTERSECTION	\cap	It returns a relation containing the tuples appearing in both of the two specified relations
SET DIFFERENCE	$-$	It returns a relation containing the tuples appearing in the first relation but not in the second relation
CARTESIAN PRODUCT	\times	It returns a relation consisting of all possible combinations of tuples from two relations.