॥ श्री महावीराय नमः ॥

Data Base Management System

UNIT-1 (Introduction)

- Overview of DBMS
- Database system V/s File system
- Architecture of DBMS
- Data models
- Entity relationship diagram
- Types of keys
- Integrity rules
- Data dictionary
- Normalization inclusion def dependencies.
- Loss less Join decompositions
- Codd's rules.

DBMS - DBMS in shorts refers to the technology of storing.

This tutorial explains the basics of DBMS such as its architecture, data models, data schemas, data independence, E-R model, relation model, relational data base design and storage and file structure and much more.

* DBMS :- Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

For example - If we have data about marks obtained by all students, we can then conclude about toppers and average marks.

"A database Management System Stores data in such a way that it becomes easier to retrieve, manipulate and produce information."

retrive - पुनः प्राप्त करना, Produce - उत्पादन करना
Manipulate - हेरफेर करना,

* Characteristics of DBMS :-

A Modern DBMS has the following characteristics.

- Real World entity :- A modern DBMS is used real world entities to design its architecture. It uses the behavior and attributes too.

for example - A school database may use students as an entity and their age as an attribute.

architecture - डिजाइन करना
attribute - विशेषता

- Relation based tables :- DBMS allows entities and relations among them to form tables.
A user can understand and architecture of a database just by looking at the table Names.

- Isolation of data and application :-
A database system is entirely different than its data. A database is an active entity, where as data is said to be passive, on which the database works. DBMS also stores metadata, which is data about data, to ease its own process.

- Less redundancy :- DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization rich and scientific process that reduces data redundancy.
Normalization - सामान्यीकरण
Splits - विभाजित

- Consistency :- Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

- Query Language :- DBMS is equipped with query language, which makes it more efficient to retrive and manipulate data. A user can apply as many and as required to retriveve a set of data. Traditionally it was not possible where file-Processing system was used.

* Properties :- A database management system stores data in such a way that it becomes easier to retrive, manipulate and produce information.

- Acid Properties :- DBMS follows the concepts of atomicity, Consistency, Isolation and Durability. These Concepts are applied on transactions, which manipulate data in a database,
Acid properties help the database stay healthy in multi-transactional environments and in case of failure.

- Multiuser and Concurrent Access - DBMS supports multi user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.

Multiple views :- DBMS offers multiple views for different users. A user who is in the sales department will have a different view of database than a person working in the production department. This feature enables the users to have a concentrate view of the database according to their requirement.

- Security :- features like multiple views offer security to some extent where users are unable to access data of other users and departments.
DBMS offers method to impose constraints while entering data into the database and retrieving the same at a later stage.
DBMS offers many different levels of security features, which enables multiple users to have different views with different features.

* Differences between the file management system and database management system.

- File Management System :-

It is nothing but a collection of programs which manage and store data in files and folders in a computer hard disk.
It helps in reading and writing data to the hard disk. It is also called a conventional file system.
Data redundancy is high and cannot be controlled easily in file management systems.

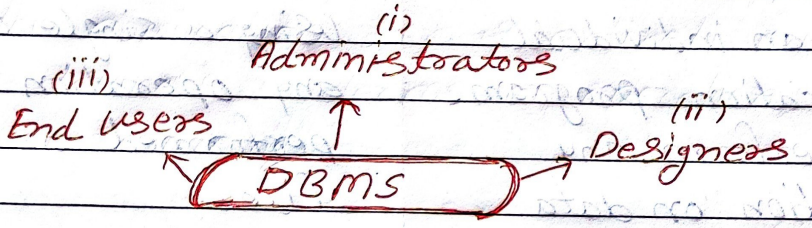
- Database Management system :-

It is defined as a software system that allows the user to define, create and maintain the database and provide control access to the data.
DBMS is a collection of programs used for managing data and simultaneously.
It supports different types of users to create, manage, retrive, update and store information.

⇒ Differences :- The major differences between the file management system and database management system (DBMS) are as follows -

File Management system	Database Management System
- Small system	- Large system
- Relatively cheap	- Relatively expensive
- Few files	- Many files
- Need an individual application program to perform any operation on data files.	- Using a single command any operation can be performed on data files.
- Transaction management is difficult.	- Transaction management is easy
- Programming is done using COBOL, C, PASCAL called as 3GL.	- Programming is done using SQL which is a 4GL
- Simple structure	- Complex Structure
- No security	- Rigorous security
- Simple backup and recovery.	- Complex backup and recovery.
- Single user	- Multiple user
- Duplication of data cannot be minimized	- Duplication of data can be minimized
- Data Consistency is less	- Data Consistency is more, because of Normalization.
- It stores the unstructured data	- It is used for storing structured data.

USERS :- A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows -



(i) Administrators :- Administrators maintain the DBMS and are responsible for administrating the database. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools and other software and hardware related maintenance.

(ii) Designers :- Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.

End users :- end users are those who actually reap the benefits of having a DBMS. end users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

* Architecture of DBMS :-

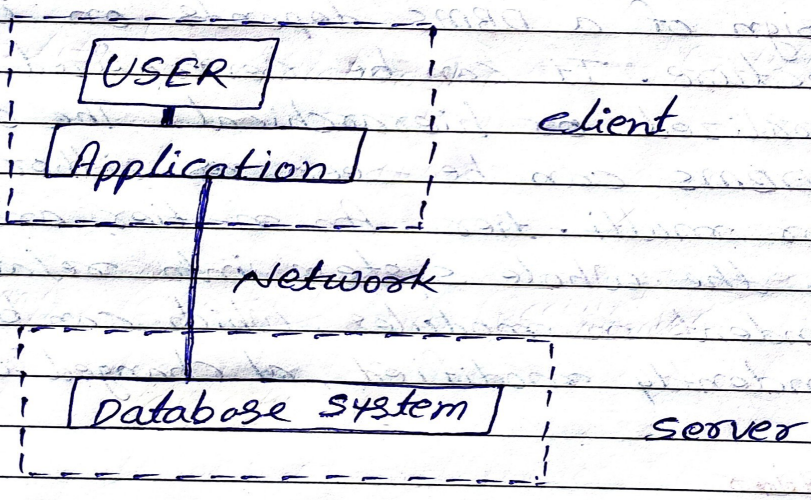
The design of a DBMS depends on its Architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent n modules, which can be independently modified, changed or replaced.

- ~~3 tier~~
- type of Architecture tier.
  - 1-tier
  - 2-tier
  - 3-tier

- 1-tier Architecture :- In 1-tier Architecture the DBMS is the only entity where the users directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users.

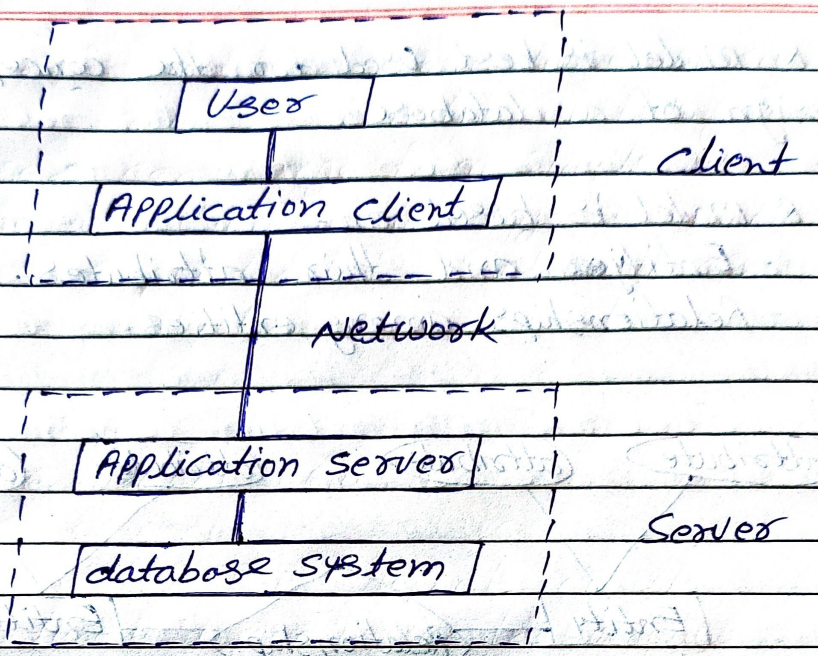
Database designers and programmers normally like to use single-tier Architecture.

- 2 tier Architecture :- Programmers use 2-tier Architecture where they access the DBMS by means of an application. Here the application tier is entirely independent of the database in terms of operation, design and programming.



Two-tier Architecture

- 3-tier Architecture :- A 3-tier Architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used Architecture to design a DBMS.



Three-tier Architecture

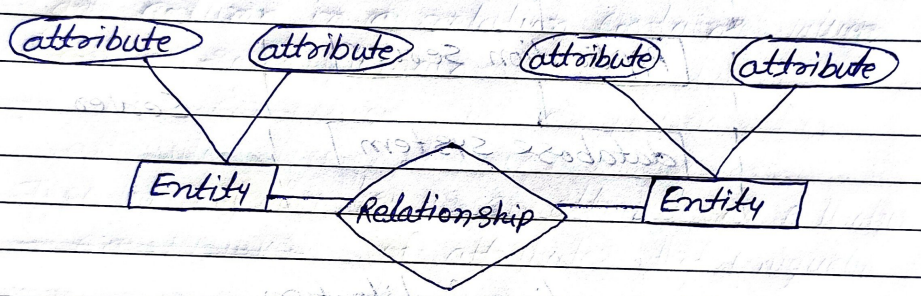
* Data Models :- Data models define how the logical structure of a database is modeled. Data models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

- Entity-Relationship Model :- [E-R Model]

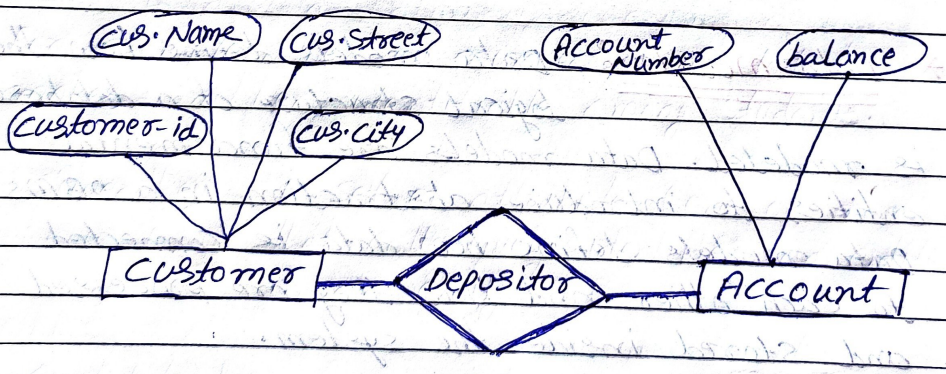
E-R models is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the E-R model creates entities set, relationship set, general attributes and constraints.

- E-R Model is best used for the conceptual design of a database.

- E-R Model is based on -
  - Entities and their attributes.
  - Relationships among entities.



ex.



Entity - An entity in an E-R model is a real world entity having properties called attributes. Every attributes is defined by its set of values called domain.
for example, in a school database, A student is considered as an entity. Student has various attributes like name, age, class etc.

Relationship - The logical association among entities is called relationship. mapping cardinalities define the number of association between two entities.

Mapping cardinalities -
- one to one
- one to many
- many to one
- many to many

Relational Model :- The most popular data model in DBMS is the relational model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an n-ary relation.

Example of tabular data in the relational model.

Attributes

customer ID	customer NAME	customer street	customer city	Account Number
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

The main highlights of this model are -

- Data is stored in tables called relations.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

* Type of keys :- Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll number of a student makes him/her identifiable among students.

- Super key - A set of attributes (one or more) that collectively identifies an entity in an entity set.

- Candidate key - A minimal super key is called a candidate key. An entity set may have more than one candidate key.

- Primary key :- A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

Data Integrity rules :- Data integrity in the database is the Correctness, Consistency and Completeness of data. Data integrity is enforced using the following three integrity Constraints.

- Entity integrity - This is related to the Concept of primary keys. All tables should have their own primary keys which should uniquely identify a row and not be NULL.

- Referential Integrity - This is related to the Concept of foreign keys. A foreign key is a key of a relation that is referred in another relation.

- Domain Integrity - This means that there should be a defined domain for all the columns in a database.

* Data dictionary :- A data dictionary contains metadata i.e data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. The users of the database normally don't interact with the data dictionary. It is only handled by the database administrators.

- The data dictionary in general contains information about the following.
  - Names of all the database tables and their schemas.
  - Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
  - Physical information about the tables such as where they are stored and how.
  - Tables constraints such as primary key attributes, foreign key information etc.
  - Information about the database views that are visible.

- This is a data dictionary describing a table that contains employee details.

Field Name	Data Type	field size for display	Description	Example
Employee Number	Integer	10	Unique ID of each employee	164500000L
NAME	Text	20	Name of the employee	David Heston
Date of Birth	Date/ Time	10	DOB of Employee	08/03/1995
Phone Number	Integer	10	Phone Number of employee	65 8364-8648

⇒ The different types of data dictionary are -
- Active data Dictionary
- Possive data Dictionary

- Active data dictionary :- If the structure of the database or its specifications change at any point of time, it should be reflected in the data dictionary. This is the responsibility of the database management system in which the data dictionary resides. So, the data dictionary is automatically updated by the database management system when any changes are made in the database. This is known as an active data dictionary as it is self updating.

- Possive data dictionary :- This is not as useful or easy to handle as an active data dictionary. A possive data dictionary is maintained separately to the database whose contents are stored in the data dictionary. That means that if the database is modified the database dictionary is not automatically updated as in the case of active data dictionary. So, the possive data dictionary has to be manually updated to match the database. This needs careful handling or else the database and data dictionary are out of sync.