Jan 20, 25

DATE: / /20
PAGE NO.

DBMS

Data

Data is a raw material or it is a collection of small units of info. It can be used in a variety of forms like texts, no., bytes etc.

It can be stored in piece of paper or electronic memory.

Database

A database is an organised collection of data so that it can be easily accessed. It can be create in such a way that only one of software programme provides access of data to all the users.

The main purpose of the data-base is to operate a large amount of info. by storing, retrieving and managing data.

for eg. →

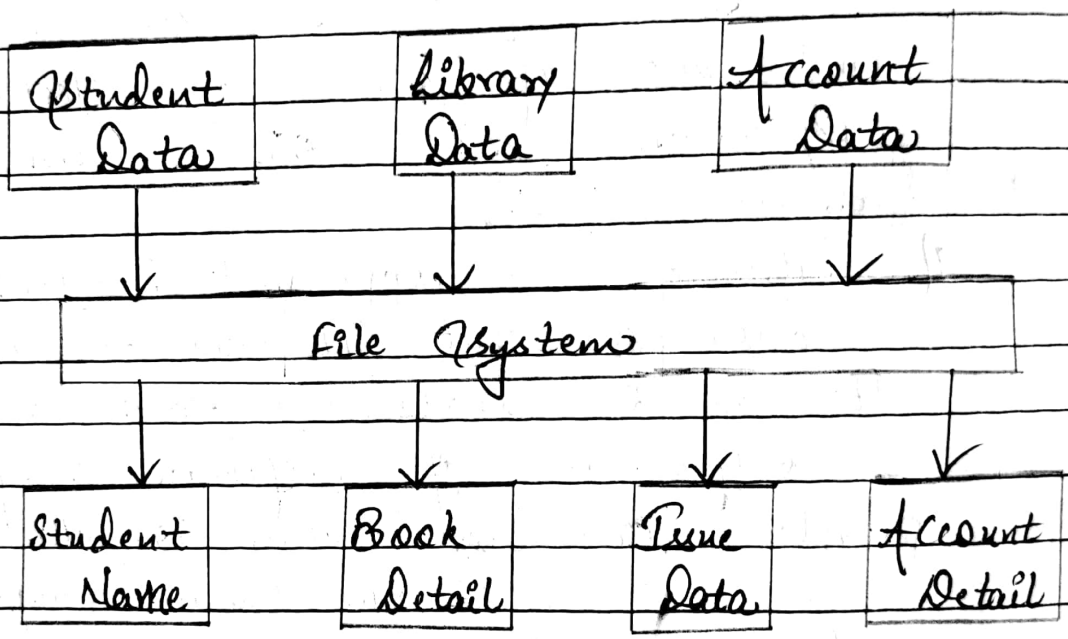
My SQL, Oracle, SQL Servers etc.

File System Approach

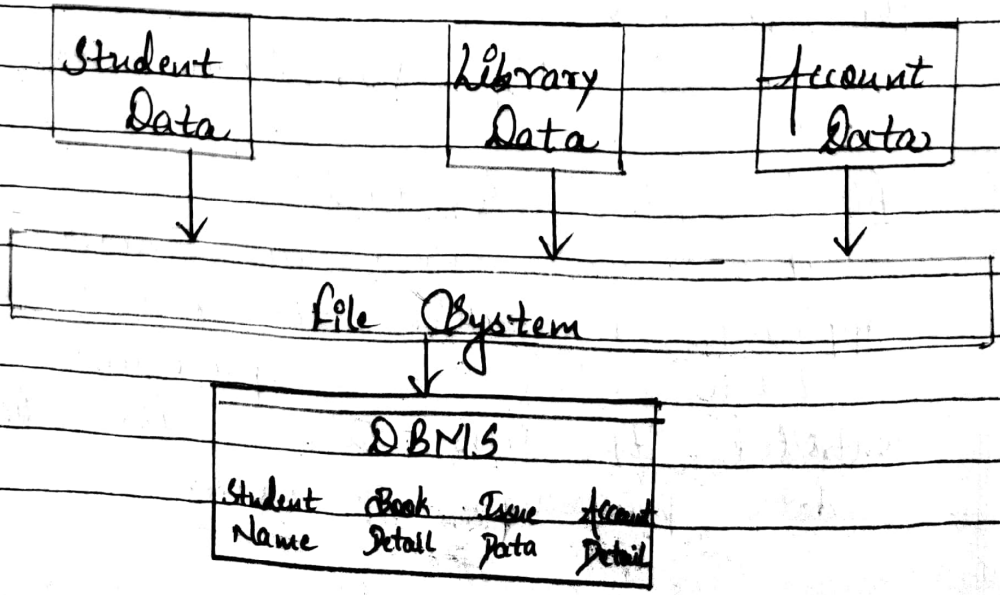
It is also called a traditional based approach in which decentralised approach was taken where each department stored and controlled its own data with the help of data processing specialist.

Here, some fields are duplicate which leads to data redundancy. So, to overcome this problem, we need to create centralized approach, i.e; DBMS.

File-based System Approach



DBMS-based Approach



DATE : / /20
PAGE NO.

Data-base Management System

DBMS is a data-base management system which is a collection of records or you can say that it's a software system that allows the user to define, create & maintain database & provide controlled access to the data.

It's basically a collection of programs that enables users to store, modify and extract info. from a database as per the requirements.

DBMS is an intermediate layer b/w programs and the data.

There are different types of DBMS from small systems to huge systems that run on different computers.

Applications → DBMS is used for computerised library systems, ATM, flight reservation systems, inventory systems.

Who makes DB software?

There are lot of DB software manufacturers depends on price, size, speed and functionality.

for eg → Oracle Enterprise

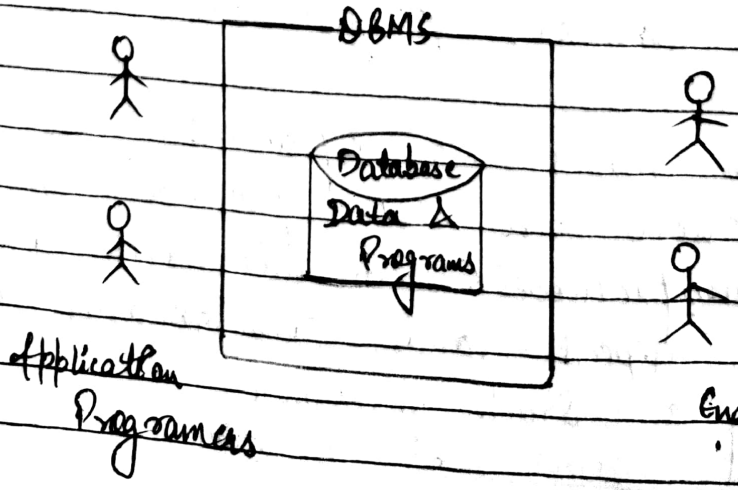
DATE : / /20
PAGE NO.

* Development and Evolution of DBMS:-

DBMS has its roots in the 1960s.
Apollo Moon Landing Project which was initiated by USA's president Kennedy of landing a man on the moon by the end of that decade, but at that time there was no system available that would be handle and manage the vast amount of info. that the project would generate.
As a result, GUAM — Generalised Update Access Method software was developed by IBM — International Business Machine.

In the mid 1960's — IDS (Integrated Data Store).
In 1967 — DBTG (Data Base Task group).

* Components of the Database System Environment



There are 5 components in the database system environment and their inter-relationship.

1. **Hardware**

The hardware is the actual computer system used for keeping & accessing the database. It consists of secondary storage devices like hard disk on which the database physically resides together with the associated I/P - O/P devices, control devices etc.

2. **Software**

The software is the actual DBMS b/w the physical database itself and the user of the system is a layer of software usually called DBMS.

The DBMS allowed the computer user to communicate within the database. It is a mediator b/w the database & the user.

3. **Data**

It's the most imp. component. The data acts as a bridge b/w the machine component & user component. The database contains the operational data & the meta data, i.e., data about data.

The actual data are separated from the program.

4. Users

There are a no. of users who can access or retrieve data on demand using the applications and interface provided by the DBMS.

4. i) Naive Users

Naive users are those users who need not be aware of the presence of the database system. This user only work through a menu-driven application programs. ATM ~~is~~

eg.

ATM user fall in this category.

ii) Online Users

who only communicate with the database directly via mail or indirectly via user interface. They should have some knowledge about the database.

eg.→ Dashboard.

iii) Specialized Users

Such users are those users who writes specialised database applications.

for eg.

Computer Added Design System (CAD).

iv) Sophisticated User

They work using query language. Each query is submitted to a processor and breakdown in DML statement into I/s that the storage manager understands

23-1-25

v) Application Programers

Professional Programmers are those who are responsible for developing application programs or user interface.

vi) Data Base Administrator (DBA)

He's a person or group incharge for implementing the data base within an organisation. The DBA has all the system priveledges allowed by the DBMS and can assigned or grant and remove levels of access to and from other users.

He's also responsible for the evaluation, selection and implementation of DBMS package.

5. Procedures

Procedures refer to the I/s and rules, that govern the design and use of the data base. The user of the system & the staff that manage

the DB required documented procedures on how to use or run the system.

* Advantages of DBMS

① Controlling Redundancy (duplicated)

In file system, each application has its own private file often lead redundancy in data. It's the wastage of storage space. DBMS control this redundancy.

② Integrity can be ~~enforced~~ ~~imposed~~ enforced

It means, data in database is always accurate. Incorrect info. can't be stored in database. For integrity some integrity constraints (rules) are enforced on the database.

③ Inconsistency can be avoided

When the same data is duplicated and changes are made at one side which isn't propagated to other side, it means it is inconsistency and the two entries regarding the same data will not agree.

(4) Data can be shared

In DBMS, we can share data in multiple applications as compared to file system.

(5) Restricting

(6) Restricted unauthorised access

When multiple users share a data base, some users will not be authorised to access all info. in the data base.

(1) Providing backup and recovery

If the comp. system fails in the middle of the complex update program, the recovery subsystem is responsible for making sure that the database is restored to the state it was in before the program started executing.

(7) Cost is low (8) Concurrency control

(9) Data model can be developed

(10) Standard can be enforced.

Disadvantage :-

(1) Complexity
(2) Size
3) Performance
4 Higher impact of failure

⇒ Comparision of File Management System with Data-Base management System

	File Management	Data base Management
(1)	Small System	large System
(2.)	Relatively cheap	Relatively Expensive
(3.)	Few Files	Many Files
(4)	Files are Files	Files are table
(5.)	Simple Structure	Complex Structure
(6.)	Redudant data	Reduce Redundancy.
(7.)	Chances of inconsistency	Consistent
(8)	Isolated data	Data can be shared
(9)	No Security	Secure data.
(10)	Simple, backup and Recovery are Complex.	Compler, Simple Backup and Recovery
(11)	Single user	Multiple users

Page No.
Date:

Views of data in Database

- It is the way of data representation to users. This is through via data abstraction

- Data abstraction : It is used to represent data to users and hiding complex details which is the process of hiding irrelevant details from users.

Types of data Views :

1. Internal Views
   - Deals with the physical storage of data.

2. Conceptual View (Logical View)
   - Describes the logical structure and relationship of data.

3. External View :
   - Provides customized data, access for individual users or groups

Page No.:
Date:

Levels of Abstractions :

(1) Physical level
- Defines how data is stored and how it can be accessed.

2. Logical level
- What data is stored in the database ? , How it is related ?, and What rules are applied to it ?

3. View level
- Describes how users retrieve information from the database.

Benefits of Data abstraction :

(1) Makes it easier for user to interact with Data base.

(2) Allows users to acess data without needing to understand the physical location of the data.

(3) Allows developers to limit what users see and how they see it.

DATE : / /20
PAGE NO.

1-25

* Master and Transaction file

A master file stores static data. It changes occasionally and stores all the details of the object.

for eg.→

In case of banking software, the customer file which contains the data about the customer like customer I'd, account no., account type, name, address etc. is a master file; because it contains the static data which is the whole information about the customer.

The other file which contains the data about the customer's transactions is called a transaction file. It's a dynamic file & updated each time for any withdrawal & deposit on a given account no.

Instances, Schemas and Subschemas

Database changes over time when info. is inserted or deleted. The collection of info. stored in a data base at a particular moment is called an instance. The overall design of the DB is called the DB schema.

for eg.→ a table is formed like following :-

DATE : / /20
PAGE NO.

Instance of a Schema/Table

Entity	Product	Price	Company
P1	Keyboard	2,000	H.P.
P2	Printer	3,000	HP
P3	Monitor	15,000	H.P.

This diagram shows record type (entities) and name of data items (attributes) only. This table doesn't show the relationships among the various files.

The schema will remain the same while the values filled into it change from instant to instant. When the schema framework is filled in with data items values, it's referred to an instance of the schema.

In other words, the description of the DB is called the DB schema, which is specified during DB design & is not expecting to change frequently.

Product - Schema

Entities	Product	Price	Company

The schema display only the name of data items and some types of constraints. Other aspects are not specified in the schema.

DATE : / /20
PAGE NO.

Subschema

A subschema is a subset of the schema & inherits the same property that the schema has. The plan (schema) for a view is often called subschema.

Subschema refers to an application programmer's view of the data item types and record types which he/she uses.

29-1-25

* Architecture of DBMS :

There are following 3 types of architecture of DBMS.

① External level or View level

It's the user's view of database. This level describes that part of the DB that is relevent to each user. It's the one which is closest to the end users. This level deals with the way in which an individual users view data according to their requirement. Therefore, same DB can have different views for different users.

The external view insulates users from the details of the internal and conceptual levels. for eg.→ One user may view dates & another user may view years.

DATE: / /20
PAGE NO.

② **Conceptual level or Logical level**

This level describes what data is stored in the DB & the relationships among the data. The middle level of the architecture is known as conceptual level.

This contains the logical structure of the entire database as seen by DBA. It's a complete view of data requirements of the organization, i.e., independent of any storage consideration.

The conceptual level represents →

- all entities, their attributes & their relationships.

⇒ entity → object of data is about (stored)
⇒ attributes → their features
⇒ relationship → connectivity

- An entity is an object about which info. is stored in the DB.

for eg → In student DB, the entity is Student.

- An attribute is a characteristics of an entity.

for eg → Student's roll no., name, class etc.

(3) **Internal level or Storage level**

It's the physical representation of the DB in the comp. This level describes how a data is stored in the DB & also concern the way the data are physically stored in the hardware.

It also cover the physical implementation of the DB to achieve run time performance and storage space utilization. It interface with the O.S. to place the data on the storage devices & so on.

5-2-26

Data Models

Data Models can be defined as an integrated collection of concepts for describing & manipulating data, relationships b/w data & constraints on the data in an organisation.

A data model has 3 components.

1. A structural part consisting a set of rules.
2. Manipulative part defines the types of operations
3. A set of integrity rules which ensures that the data is accurate.

Purpose The purpose of data model is to represent data and to make the data understandable.

There are 3 types of data models.

1. Object based Data Models
2. Physical Data Models
3. Record based Data Models (Logical)

1. Object based Data Models

It's used to concept such as an entities, attributes & relationships.

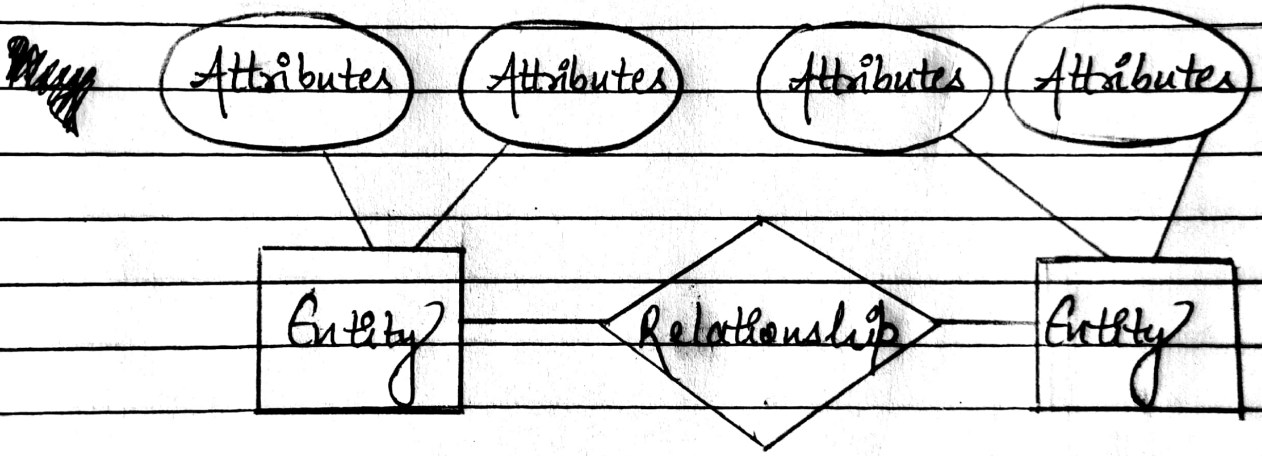
- An entity is an object (a person, place, concept, event) in the organisation, ie, to be represented in the DB.

DATE: / /20
PAGE NO.

- An attribute is a property that describes aspects of the objects.
- And, relationship is an association b/w entities.

→ Some of common object-based data models are -

(i) E-R Model → Entity Relationship Model
(ii) Object Oriented Models
(iii) Cementic Model
(iv) Functional Model



(2) Physical Data Models

They describes how data is stored in the computer representing info. such as records, structures, ~~reports~~ records, ordering & access path.

Eg. → Unifying Model

(3) Record base Logical Model

These models are used to describing the data at the logical and view levels.

DATE : / /20
PAGE NO.

bz, the data base is structured in fixed format records of many types. Each record type define a fix number of fields (attributes) and each field is used fixed lengths.

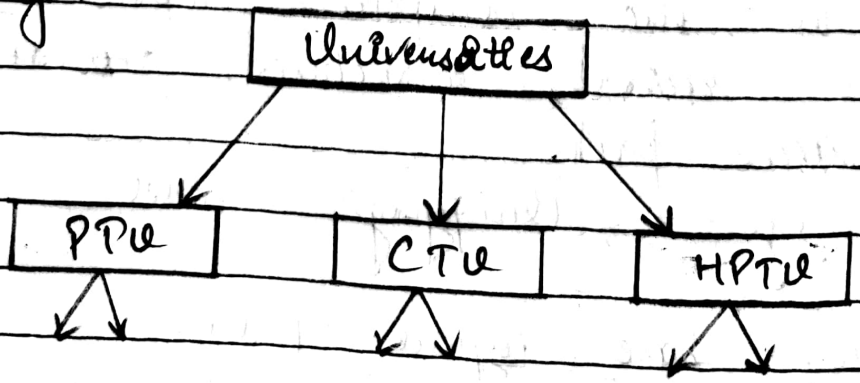
There are 3 most widely used record - based Data Models.

1. Hierarchical Model
2. Network Model
3. Relational Model

(i) Hierarchical Model

It is one of the oldest database models dating from late 1950s. One of the first hierarchical database Info. Management System (IMS) was developed by North-American Rockwell Company & IBM. This model is like a structure of a tree with the records forming the nodes & fields forming the branches of the tree.

for eg:→



In this Model, reports are linked with other superior reports on which they are dependent and also on the reports which are dependent on them. A tree structure represent one to many relationship

Operations on Hierarchical Model

(1) **INSERT** operation

It's not possible to insert info. of the suppliers, ie., S4 who doesn't supply any part. This is because a node can not exist without a root.

(2) **UPDATE** operation

Suppose, we wish to change the city of supplier S1, then we will have to carry out 2 operations such as searching S1 for each part and then multiple updations for different occurances.

→ The suppliers records

S.No.	Name	Status	City
S1	Sumeet	20	Jadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

The Part Records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment Records

SNo.	PNo.	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

Advantages

Simplicity → The relationships b/w the various layers is logically very simple.

Data Security → It provides security offered by the DBMS

Data Integrity → Because this model is based on parent - child relationship, there's always a link b/w a parent segment & the child segment under it.

The child segment are always automatically referred by its parent. So, this model promotes data integrity.

④ Efficiency → It contains a large number of 1:N relationship (one to many) and when the users required large number of transactions using data whose relationships are fixed.

→ Disadvantages

① Implementation Complexity → Although this model is conceptually simple, but it's quite complex to implement

② Data base Management Problem → If you make any changes in the database structure, then you need to make the necessary changes in all the application programs that access the DB.

③ Program Complexity → This model requires knowledge of complex pointer systems, which is not a cup of tea of ordinary users (who have no programming language).

④ Operational Anomalies (Errors) → This model suffers from the insertion, updation & deletion anomalies.

⑤ Lack of Structural Independence → Structural independence exists when the changes to the DB structure doesn't affect the DBMS ability to access data.

DATE : / /20
PAGE NO.

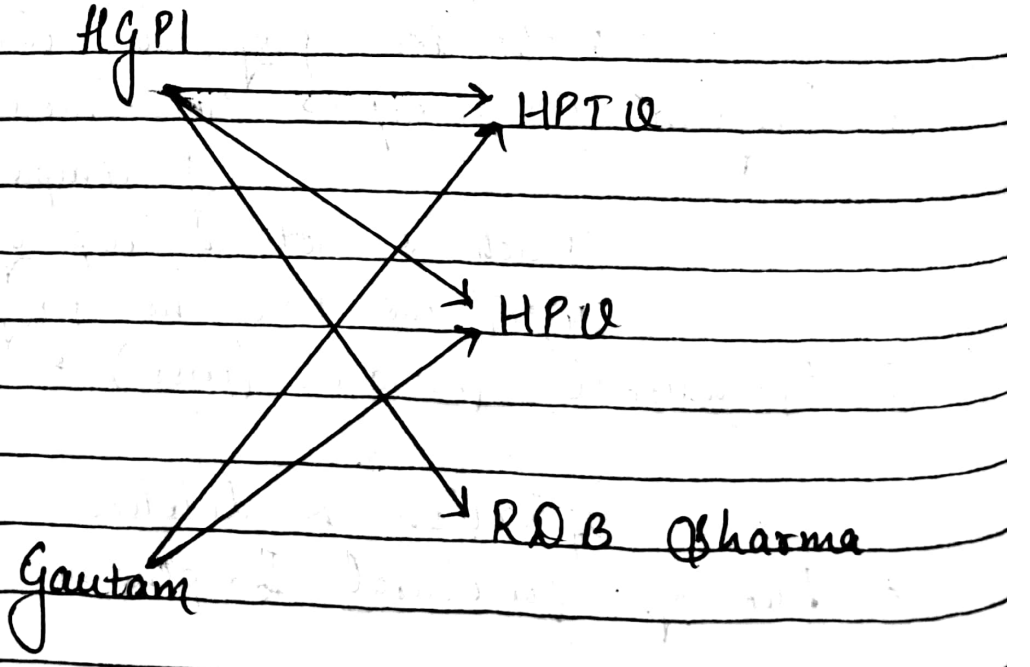
(ii) # Network Model

It replaces the hierarchical tree with the graph. Thus allowing more general connections among the nodes.

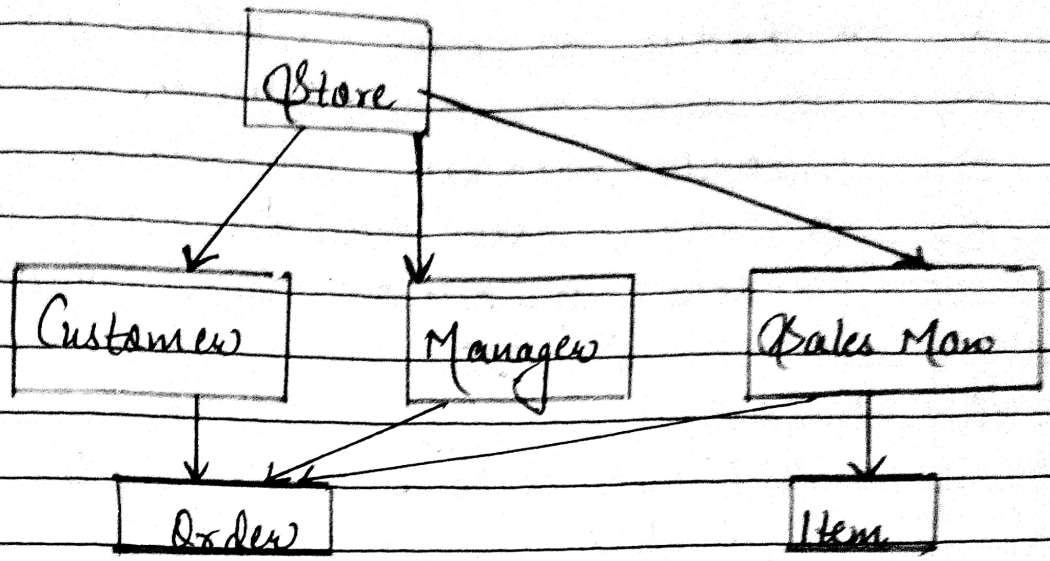
The main difference of the ntwk model from the hierarchical model is its ability to handle many to many (N:N) relationships. In other words, it allows a record to have more than one parent.

→ Suppose, an employee works for 2 departments. In network DB terminology a relationship is a set. Each set is made up of atleast 2 types of records - an owner record & a member record.

Diagram



17-2-25



features

① Ability to merge relationships
In this model, bz of more relationships the data is more related.
Both one to one and many to many.

② Many Paths
There can be more than one path to the same record bz of more relationships.

③ Circular linked list
The operations in this model are done with the help of the circular linked list.

Advantages

① Ntwk models represent complex data relationships better than the hierarchical model.

② It handles so many relationship's types.
③ Data access is more flexible.
④ Improved database performance.
⑤ It includes data definition lang. DDL & data manipulation lang. DML.

Disadvantages

① Database contains a complex array of pointers.
② System complexity limits efficiency.
③ Structure changes require changes in all application programs.
④ Keep heavy pressure on programmers due to complex structure.
⑤ Any change like updating, deletion & insertion is very difficult.

(iii) Relational Data Model

It's the primary model which is used widely around the world for data storage & processing.

Concepts →
- Tables
- Tuples
- Relation instances (a set of tuples represent relation)
- Relation schema (table name, attribute & their name)
- Relation key (primary, candidate, foreign)

- Constraints (Key constraints, Domain constraints, referential Integrated constraints)

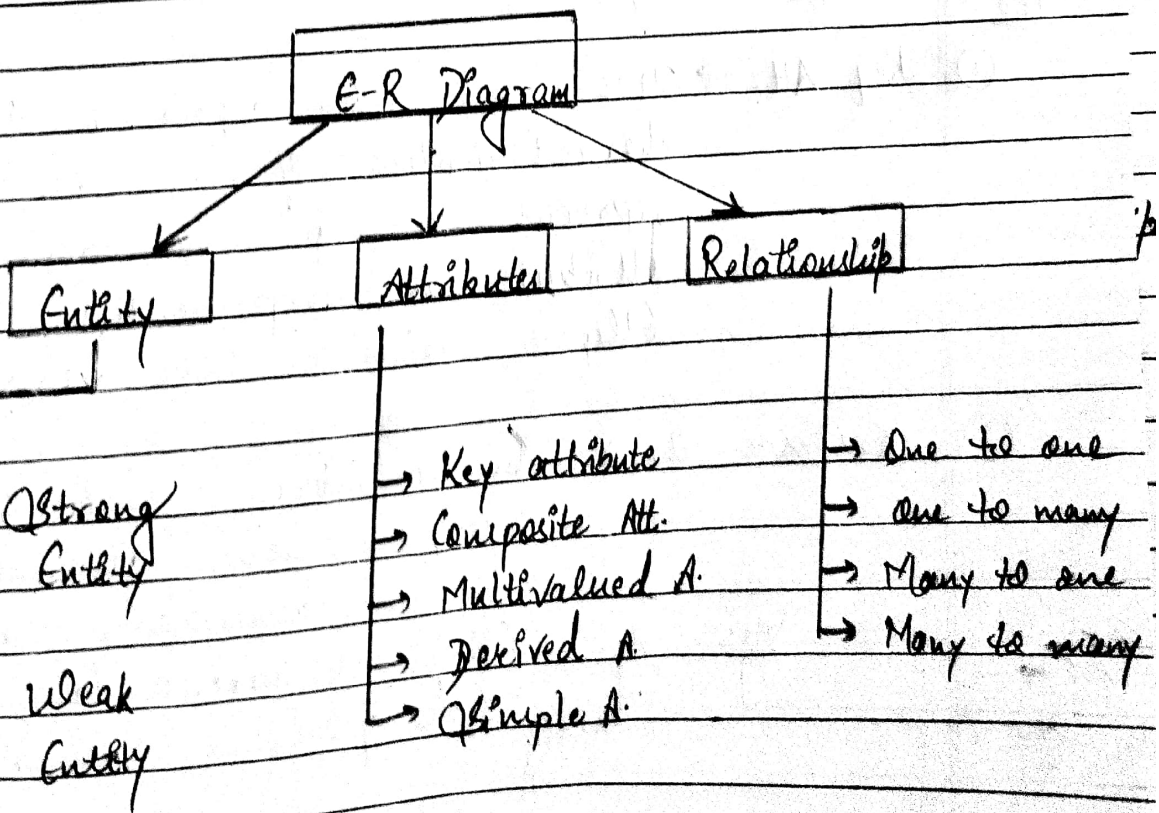
* E-R Model

① E-R model stands for Entity - Relationships Model. It's a high level data model used to define the data elements and their relationships.

② It develops a conceptual design for the database which is very easy and simple.

③ The database structure is ~~portrayed~~ portrait as a diagram called as E-R diagram.

→ Components of E-R Diagram



18-2-25

- **Entity**
  An entity maybe any object, class, person or place.
  In the ER diagram, an entity can be represented as Rectangle.
  eg. department, organisation etc.
  Strong → uniquely identified
  Weak → depend on other

- **Attributes**
  It is used to describe the property of an entity. Ellipse is used to represent an attribute.
  eg. → age, roll no., contact no. etc.

TYPES

① Key Att. → It is used to represent the main characteristics of an entity. It represent a primary key. The key attribute is represented by an Ellipse with a text underline.

② Composite Att. → An attribute that composed of many other attributes is known as the composite attribute.
  eg. age (birth + current date),
  average (no. of values + values).

③ Multivalued attribute → An attribute can have more than 1 value, these attributes are known as multivalued Attribute.

eg. phone number.

④ Derived Att. → An attribute that can be derived from other attribute is known as derived attribute.

eg. age (as current age increases per day).

⑤ Simple Att. → It's a fundamental component of data that can not be broken down into smaller parts.

eg. salary, gender.

- **Relationship**

A relationship is used to describe the relation b/w entities.

Diamond is used to represent the relationship

Types

① One to one → when only 1 instance of an entity is associated with the relationship.

for eg. a female can marry only one male & one male can marry only one female.

② One to many → when only 1 instance of an entity on the left and more than 1 instance of an entity on the right associates with the relationship.

DATE: / /20
PAGE NO.

eg. → scientists can invent many inventions but the invention is done by only specific scientist.

(3) Many to one → when more than 1 instance of an entity on the left and only 1 instance of an entity on the right associate with the relationship

eg. → students enroll for only one course but the course can have many students.

(4) Many to many → when more than 1 instance of an entity on the left and more than 1 instance of an entity on the right associate with the relationship

for eg. → employee can assign by many projects and project can have many employees.

2-25

Database languages

Once data is stored or filled, it requires manipulation like insertion, deletion, updating and modification of data. For these operations, a set of languages are provided by DBMS. There are 4 types of DBMS languages –

1. DDL
2. DML
3. DCL
4. TCL

DATE: / /20
PAGE NO.

① DDL → Data Definition language

It's the lang. that allows user to define a data & their relationship to other types of data.

The DDL commands are -
Create, Rename, Drop, Truncket.

② DML → Data Manipulation language

It's a lang. that provides a set of operations implement for the basic manipulation.

The DML commands are -
Insert, Delete, Update, Select etc.

③ DCL - Data Control language

DCL is used to access the store data. It's mainly used for revoke and grant the user to access to a database.

DCL commands are -
Grant, Revoke

④ TCL Transaction Control language

TCL is a language which manage the transactions within the database. It's used to execute the changes made by the data manipulation language statements.

The TCL commands are -
Commit, Rollback

Database Access for Application Programs in DBMS

DBMS provides access to data for application programs through tables, queries & other tools. for eg. → IBM DB2, MY SQL, Oracle DB, MS Access.

- Database Access

① Application programs can access data in a DB through tables, queries & other tools.

② Relational DBMS stores data in a row based table that connects related data elements.

③ Relational DBMS includes function in that manner that maintain security, accuracy, integrity & consistency of the data.

④ Most DBMS use logical & physical data backup.

Data Base ~~Abstraction~~ Administrator

→ Role of DBA

A database administrator is a person or group incharge of implementing DBMS in an organisation. The DBA job requires a high degree of technical expertise. DBA consists of a team of people rather than just 1 person.

The primary role of a DBA is -

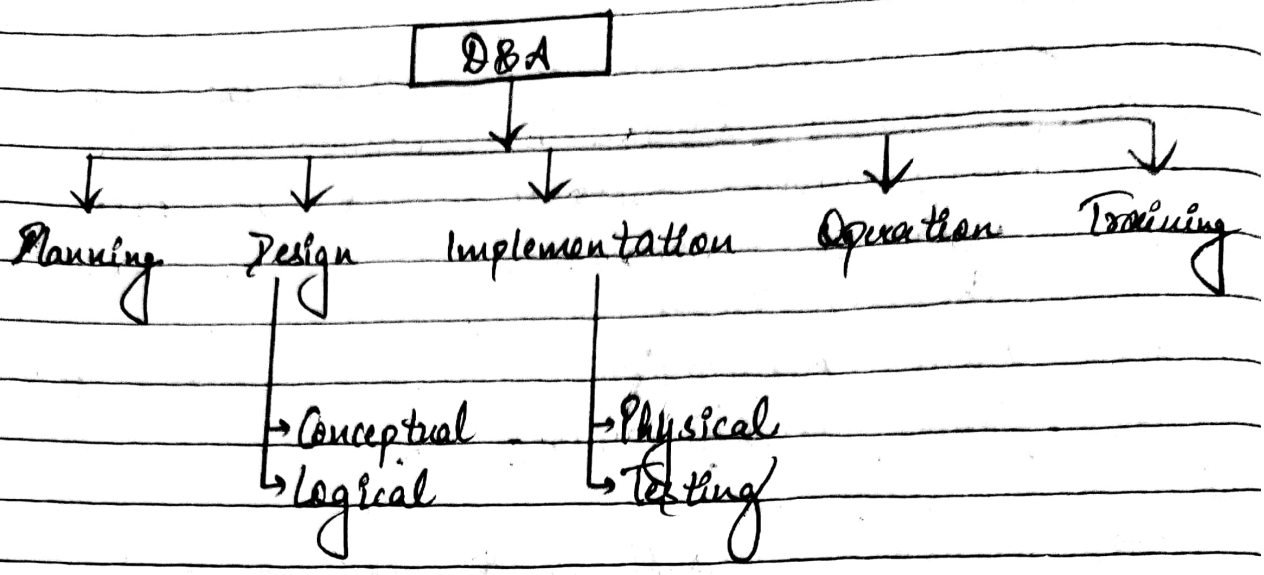
1. Database Design
2. Performance Issues
3. Database Accessibility
4. Capacity Issues
5. Data Replication
6. Table maintainance

→ Responsibilities of DBA

1. Makes the decision concerning the content of the database.
2. Plan the storage structure and access strategy.
3. Provides the full support to the users.
4. Defines the security and integrity checks.
5. Interpreter backup and recovery strategies.
6. Monitoring the performance and responding to the changes in the ~~envi~~ requirements.

→ Skills required for DBA

1. Database Designing.
2. Knowledge of SQL - structure Query language
3. Knowledge about distributed architecture.
4. Knowledge on different operating servers.
5. Idea on relational DBMS.
6. Ready to face challenges and resolve the problem quickly.



24-2-25

Challenges faced by DBA

DBA faced many challenges including security, compliance and data loss. These challenges can lead to downtime & business disruption.

→ Security and privacy ensuring the security & privacy of sensitive data. → Compliance & regulatory meeting requirements of compliance & security guidelines are followed

Datalos → Back up & recovery

→ Integration

→ Cost

→ Database Mantainance

25-2-25

Storage Manager in DBMS

A storage manager is a software that allows users to define, store, maintain and manage data in a structured and efficient manner. They not only store & manage large amounts of data but also provide functionality of performance and security.

Storage Manager is a program that provides an interface b/w the data stored in the database & the query received. It is also known as DB Control System. It maintaines the consistency & integrity of the DB by applying the constraints and executing the DCL statements. It is responsible for updating, storing, deleting & retrieving data in the DB.

Components of Storage Manager

① **Authorization Manager :-**
It ensures role based control access, i.e., checks whether the particular person is priviledged to perform the requested operation or not.

② **Integrity Manager :-**
It checks the integrity constraints when the DB is modified.

④ Transaction Manager :-
It controls concurrent access by performing the operations in a scheduled way that it receives the transaction.

⑤ File Manager :-
It manages the file space and the data structure used to represent the information in the DB.

⑥ Buffer Manager :-
It's responsible for cache memory & the transfer of data b/w the secondary storage and main memory.

Query Processor in DBMS
A query processor in DBMS is a component that processes user request to access data. It translates the user queries into actions that retrieve data from the DB.

→ How does a query processor work?
1. Receives a query in the form of SQL text
2. Passes the query for syntax & logical meaning.
3. Generate an execution plan.
4. Executes the plan to retrieve data.
5. Return the result to the user.

DATE : / /20
PAGE NO.

→ Components of Query Processor

1. Query Parcel :
   It converts the queries into DB language

2. Query Rewritter :
   Rewrites query to improve performance.

3. Query Optimizer :
   Execute the queries efficiently.

4. Query Executer :
   Executes the whole query plan.

3-3-25

* Database Schema

A database schema is a structure that represents the logical view of the entire database. It defines how the data is organized & how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities, & the relationship among them. It contains the descriptive detail of the database, which can be depicted by means of schema diagrams, its designers & its programmers.

DATE: / /20
PAGE NO.

* Views of Schema

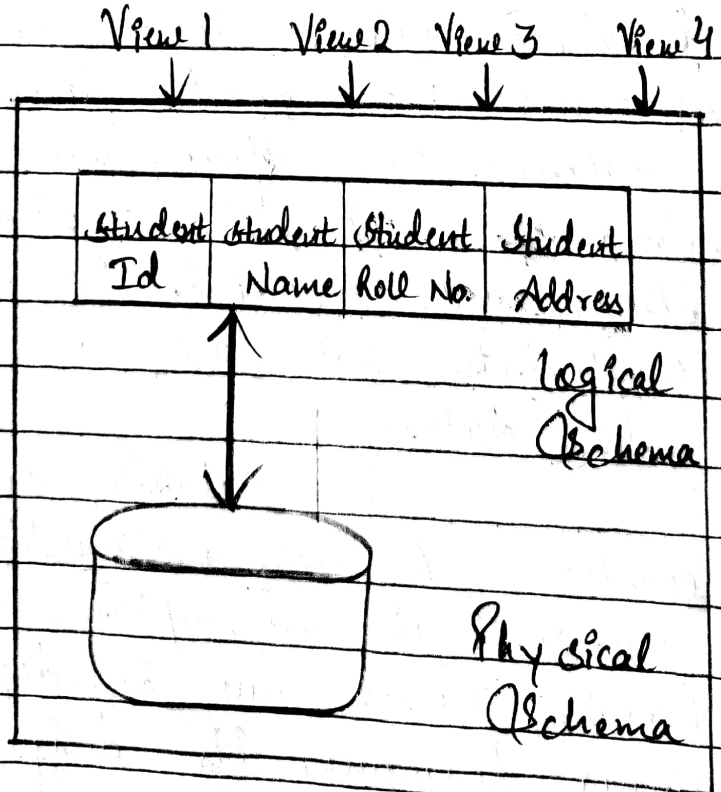
① **Physical DB Schema**

This Schema represent the actual storage of data & its form of storage like files, indexes etc.

It defines how the data is stored in a secondary storage.

② **Logical DB Schema**

This Schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views & integrity constraints.



Keys

Keys play an important role in the relational database. It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

for eg.→ Roll number is used as a key in the student table, bz it's unique for each student.

→ Why are keys necessary?

In order to define different kinds of integrity constraints in the DB, we employ keys. In contrast, a table is a collection of records for different occurencies for any relation.

→ Types of Keys

① Primary Key

It's the first key used to identify one & only one instance of an identity uniquely.

eg.→ Roll No., student_Id.

4-3-25

→ Objectives of Primary Keys

① Data Uniqueness is enforced by the primary key which stops duplicate reports from being inserted.

DATE: / 120
PAGE NO.

② It offers a rapid and effective means of gaining permission to access particular reports from the table.

② Candidate Key

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple (row).
- Except for the primary key, the remaining attributes are considered a candidate key.
- The candidate keys are as strong as the primary keys.
- For eg.→ In the student table, roll no. is the primary key & the rest of the attributes like phone no., passport no., license no., are considered as the candidate key.

→ Objectives

① A candidate key's main goal is to make sure that no 2 items in a table have the same set of attributes values combine.

② It offers a dependable way to identify records individually.

③ Choosing primary keys, creating database schemas & maintaining data integrity, all depend on candidate keys.

DATE : / /20
PAGE NO.

③ Super Key

It's an attribute set that can uniquely identify a tuple. A super key is a super set of a candidate key.

for eg → in student table, (roll no & student's name) the name of 2 students can be the same but their roll no. can not be the same.

Hence, this combination can also be a key

→ Objectives

① Making ensuring that no 2 reports in a DB have a same mixture of values of attributes is the main function of the super key.

④ Foreign Key

They are the columns of the table used to point to the primary key of another table. Every student study in a college are 2 different entities (student & college). So, we can not store the college information in the student table.

That's why we link these 2 tables through the primary key of 1 table. In the college table, college Id is a foreign key & in the student table, student's roll no. is a primary key.

→ Objective

① Referencial Integrity refers to the validity and consistency of relationships across tables. It is an essential goal of a foreign key.

⑤ Alternate Key (A.K. = total no. of candidate − primary key)

There may be one or more attributes or a combination of attributes that uniquely identify each tuple in the relation. These attributes or a combination of attributes are called candidate keys.

for eg.→

Student relation has 2 attributes, Student roll no. & passport no. both are candidate keys but on dividing them, roll no. is primary key & passport no. is alternate key.

Relational Model

→ Properties of relation

1. In a table, order of rows is immaterial.
2. Order of columns is immaterial.
3. In a table, there are no duplicate rows.
4. Each row of the table contains only 1 value.
5. In a table, the values in each columns must come from the same attribute domain

Degree & Cardinality

The no. of columns in a table is called degree of relation. A relation with only 1 column is called unary relation, with 2 columns is called binary relation and 3 columns is called ternary relation and for n columns is called n-ary.

By contrast, the number of rows in a table called the cardinality of the relation.

eg. table of Student

Roll No.	Name	Class	Subject
1	A	4th	OS
2	B	4th	DS
3	C	4th	DBMS

cardinality of relation = 3

degree of relation = 4

* Relational Constraints

(1) Domain Constraints

It specifies that the value of each attribute must be an atomic value from the domain of attribute. The data type associated with domains include integers, characters, string, data and time etc.

for eg.

Roll No.	Name	Class	Subject
1	A	4th	DBMS
2	B	4th	DBS
3	C	4th	DBMS
A	d	4th	MC

↑ not allowed, bz no. is 'int' attribute

(2) TUPLE UNIQUENESS Constraints

Relation is defined as a set of tuples. By definition, all elements of a set of tuples must be unique.

for eg.

Roll No.	Name	Class	Subject
1	A	4th	DS
2	B	4th	DS
3	C	4th	MC
A	d	4th	MC

↓ not allowed

DATE : / /20
PAGE NO.

(3) Key Constraints

Primary key must have unique value in the relational table.

for eg. if roll no. is a primary key then,

Roll No.	Name	Class	Subject
1	A	4th	Qs
2	B	4th	Qs
3	C	4th	MC
1	A	4th	Qs

not allowed

* Integrity rules

(i) no null value.
(ii) primary key in I table is related to the foreign key of II table.

10-3-25

Relational Algebra

It is a procedure language. It specifies the operations to be performed on existing relations to derive result relations.

→ Relational operators

① Union
② Intersection
③ Project
④ Select
⑤ Difference
⑥ Product
⑦ Join
⑧ Division

for union, intersection and difference operations the 2 operand relations must be compatible.

① Union (∪)

The union of 2 relations A & B, A∪B is the set of all tuples (T) belongs to either A or B (or both).

A

B.No.	Name	Major
123	James	History
158	Parks	Math
271	Smith	History

B

S.Number	Name	Interest
105	Andarson	Maugment
123	James	Hist

DATE / /20
PAGE NO.

A∪B

S.No.	Name	Major
123	James	History
158	Parks	Math
271	Smith	History
105	Anderson	Management

② Intersection (∩)

The intersection of 2 relations A & B, A∩B is the set of all tuples (T) belongs to both A and B.

A∩B

S.No.	Name	Major
123	James	History

③ Difference (−)

The difference b/w 2 relations A & B, A−B is the set of all tuples (T) belongs to A and not to B.

A−B

S.No.	Name	Major
158	Parks	Math
271	Smith	History

④ Product (A×B)

The product of 2 relations (some times called the cartesian product) is the concatenation of every tuple of 1 relation with every tuple of second relation. The product of relation A and relation B has m times n tuples, (m×n).

The product is denote A×B or A times B.

A×B

S.No	Name	Major	S.No	Name	Interest
123	James	History	105	Andarson	Mngmnt
123	James	History	123	James	History
158	Parks	Maths	105	Andarson	Mngmnt
158	Parks	Maths	123	James	History
271	Smith	History	105	Andarson	Mngment
271	Smith	History	123	James	History

⑤ Project (π)

Projection is an operation that selects specified attributes from a relation. The result of the projection is a new relation having the selected attributes, in other words, projection picks columns out of a relation. The general form of the project operation is π (Attributelist)

11-3-25

(6) Select (σ)

Where as the projection operation takes a vertical subset (columns) of a relation, the selection operation operator takes a horizontal subset (rows). The select operation is used to select a subset of the tuples from a relation that satisfy a selection condition. The general form of the selection operator is

σ < selection condition > R

σ is a symbol for selection operation, selection condition is a boolean expression and, R is a relation.

S.No.	Name	Major	Age
101	James	History	25
102	Smith	Math	26

σ age > 25 student

(7) Join operation

We can join 2 tables (relations) if both tables have a column (attribute) defined on some common domain. The join operation is the combination of product (cartesian product) of selection operation and projection operation.

DATE: / /20
PAGE NO.

Operation
- form the product A×B
- Do a selection to eliminate some tuples.
- Then reboot duplication attribute with project.

12-3-25

Difference b/w DBMS and RDBMS

	DBMS	RDBMS
1.	Concept of relationship is missing.	Based on concept of relationship.
2.	Very slow speed.	Very fast speed.
3.	less H/w & s/w needed.	High H/w & s/w needed.
4.	Limited facilities and utilities.	Many facilities and utilities.
5.	DOS (Disk OS) platform is used.	DOS, UNIX, Windows platforms are used.
6.	Uses the concept of file.	Uses the concept of tables.
7.	Use a 3GL (3rd generation language). (C, C++)	Use a 4-GL. (SQL)
8.	Eg. are dBASE & foxBASE	Eg. are ORACLE, FOCUS, etc

DATE : / /20
PAGE NO.

17-2-25

Relational Calculus

In a relational calculus, a query is expressed as a formula consisting of a number of variables and an expression involve these variables. It's upto the DBMS to transform these non-procedural queries into equivalent, efficient procedural queries.

The relational calculus is used to measure the selective power of relational languages. It's based on predicate calculus, which is calculating with predicates.

Propositions specifying a property consists of an expression that names an individual object called the predicate. If for instance, p and q are propositions we can build for these propositions for the expressions "NOT p", "p AND q", "p OR q" and so on. In other manner, we can build a proposition that specifies a certain property & characteristic of an object.

for eg. → ① B-Tech is a class. → Class (B-Tech)
② Ajay is a student. → Student (Ajay)
③ India is a country → Country (India)

In these eg., the part is a class/ like eg. of predicates. (Student/ Country)

A convenient method in which we use symbols for both the predicate & the object, we can write the statement P(X)

predicate → variable

→ P(X)
class (B-Tech)
student (Ajay)
country (India)

→ X is the argument is a one place predicate
eg. Class (X) → one place predicate
this X is replacable by constant such as Class (B-Tech).

→ Relational Calculus can be classified into 2 categories:
1. Tuple Oriented Relational Calculus
2. Domain Oriented Relational Calculus

1. Tuple Oriented Relational Calculus
It is based on specifying a no of tuple variables. Each such tuple variables ranges over a particular database relation. This means that a

variable can take any individual tuple from the relation as its value. It's written in the form of $\{t \mid (condition\ C)\}$

where, t is a tuple variable
condition key is a conditional expression involving t.
| → OR → joins 2 variables.

The result of such a query is a relation that contains all the tuples that satisfy condition.
for eg. → the query is
Book(t) | t.Price > 100

→ A general expression of tuple relational calculus is as follows:-

$\{t_1.A_1, t_2.A_2, \dots t_n.A_n \mid Cond(t_1, t_2, \dots t_n)\}$

where,
t_1, t_2, t_n are tuple variable.
A_1, A_2, A_n are attribute of relation.
Cond is condition or formula.

→ A formula is defined as follows -

- Every condition is a well form formula (wff).
Here, A wff is constructed from conditions, boolean operations (AND, OR, NOT) and

quantifiers like for all ($\forall$) values (all $\forall$) or there exist ($\exists$).

→ There are following rules which are applicable on WFF :-

- Every condition is WFF.
- If f is a WFF then (F & not f are also WFF) (F = formula)
- If f_1 and f_2 are WFF then (f_1 AND f_2), (f_1 OR f_2) are also WFF.
- If F is a WFF in which T occurs as a free variable. (∵ T = Tuple)

⇒ Free variables are those variables when the meaning of formula change If all the occurances of range variable say 'x' were replaced by some other variable say 'y'. Then $\exists T(F)$ or $(\forall T(F))$ are WFF

- Nothing else is WFF.

⇒ Bound variables

Bound variables are those range variables when the meaning of the formula would remain unchanged if all the occurance of range variable say x were replace by some other

variable say y, then range variable x is called as the bound variable.

for eg. → Exists x (x>3)

Here, WFF simply state that there exists some integer x that is greater than 3.

⇒ Free variables

Free variables are those range variables when the meaning of formula is changed, if all the occurances of range variable say 'x' where replace by some other variable say 'y'.

for eg. → x (x>3)

24-3-25

② Domain Oriented Relational Calculus

The Domain Calculus differs from the tuple Calculus, in the type of variables used in formulas. In Domain Calculus the variables range over single values from domains of attributes rather than ~~the~~ ranging over tuples. An expression of the Domain Calculus is the following -

$\{X_1, X_2, \ldots (X_n \mid COND (X_1, X_2, X_{n+1}, \ldots X_{n+m})\}$

Where $X_1, X_2, X_n, X_{n+1}, X_{n+m}$ are domain variables

that range over domains of attribute.
(COND) is a condition or formula of the domain relational calculus.

It consists of elements =
① Domain Variables
② Conditions having 2 forms
(i) Simple Comparison of the variab
(ii) Membership Conditions of the form → R (Term1, Term2, ---)

Here, R is a relation and each Term is a pair A:V
Ex, A is an attribute and V is variable or constant.

for eg → EMP(Emp No: 100, Emp Name: 'Raj');
is a membership conditions, means it evaluates to TRUE if and only if there exists an EMP tuple or relation having Emp No. = 100 & Emp Name = Raj.

It can be WFF found formed in accordance with rules of tuple calculus.

⇒ free and Bound Variables
Same as Tuple relational calculus

→ Eg of Domain Relational Calculus →

Emp. City = Nahan

Expression →

Emp X where Emp No. (I.d: Emp
City: Nahan)

the first expression denotes the set of all employee numbers; the second denotes the set of all employee numbers in the ~~location~~ relation Emp located in Nahan.

* Query Based Example (QBE) } will discuss
SQL Query language } later

SQL language

→ Intro to SQL structured Query
SQL is an advance relational databa
language. SQL commands can be combine
with the DB commands to define &
access data from a DB. It provides a
small & concise set of commands. Th
helps us to save time & reduce the
amount of programming required to
perform complex queries.

→ Two types of SQL
(1) Interactive SQL
(2) Embedded SQL

(1) Interactive SQL
In this form, we enter a command. It's executed and we can get the O/P.

(2) Embedded SQL
It consists of SQL commands put inside of programs that are written in other languages such as C, Pascal etc. This can make these programs more powerful & efficient. The O/P of SQL commands is passed to a variables enabled by the program in which it's embedded.

Advantages of SQL

- Easy & Simple
- SQL for relational DB
- for DB query → SQL
- Reduce programming complexities
- Portable
- Time Saving
- Affordable

25-8-25

DATE: / /20
PAGE NO.

Components of SQL

(1) Data Definition Language (DDL)

DDL is used to define the DB, modify its structure after it has been created and destroy it after it's no longer needed.

DDL commands are -

1. Create table
2. Create view
3. Alter table
4. Drop table

(2) Data Manipulation Language (DML)

It's used to manipulate data such as to insert, modify, delete or query the data in the database.

DML commands are -

1. Insert
2. Delete
3. Update
4. Select

(3) Data Control Language (DCL)

DCL is used to grant and revoke authorization for database access. & Auditing the DB use and dealing with transactions, there are many ways that a DB becomes corrupted. The tools provided by the DCL

will prevent many of those problems.
DCL commands are —

① Commit
② Rollback
③ Grant
④ Revoke

Commands

DDL commands

Create table → It is used to define or create a table.

Syntax is : CREATE TABLE < Table Name >
< Column specification >, < Column specificat

where, Table Name gives a name of table and, column specification includes

① Column name
② Data type
③ length of column (if any)
④ constraints (if applicable)

eg: CREATE TABLE EMPLOYEE
(Emp_Code char (4), Salary int....);

This will create the employee table with 4 columns.

Integrity Constraints

Constraints are used to ensure accuracy & consistency of data in a relational database.

(1) NOT NULL Constraint

It specifies that a value of a column can not be null.

eg. → Create table Employee
(Emp_Code char(4) NOT NULL, - - -);

(2) DEFAULT Constraint

It's used to define a default value for an attribute.

eg. → (Gender Character (1) Default = 'M');

(3) UNIQUE Constraint

It can be specify that the values of a rows for a specified column must be unique or different.

eg. → (Phone integer Unique);

(4) CHECK <Condition> Constraint

It can be utilised to check the validity of data entered into a particular table column.

eg. → Age Int
CHECK (Age >= 18);

DATE : / /20
PAGE NO.

(5) PRIMARY KEY Constraint
It specifies that 1 or more attributes that make up the primary key of the table identified uniquely.

eg. → (Roll_No. int PRIMARY KEY);

(6) FOREIGN KEY Constraint
It is used to enforce referential integrity b/w tables in a relational DB. A column defined as a foreign key is used to reference a column defined as a primary key in another table.

eg. → Primary Key (Roll_No.) → table of class
Foreign Key (Roll_No. REFERENCES Student(Id)) → table Stude

(ii) Alter table → This command is used to add columns to tables & modify columns of table.

Syntax is :- ALTER TABLE <Table Name> ADD <Column Specification>

:- ALTER TABLE <Table Name> MODIFY <Column Name>

where ADD is used to add a column & Modify to make changes.

(iii) Drop table → Table maybe destroyed/deleted with the SQL command drop table

Syntax is :- DROP TABLE < Table Name >

(2) DML commands

(i) Insert Command → It's used for inserting the data into tables.

Syntax is :- INSERT INTO < Table Name > [< Column Name > Values < Value >]

(ii) Update Command → To modify data values, within ~~tiny~~ one or more columns for one or more rows of a table, you can use update command.

Syntax is :- UPDATE < Table Name >
SET < Column Name > = Value Expression
WHERE < Condition >

eg. UPDATE Employee
SET Salary = 40,000 WHERE Emp_Code = '101';

(iii) Delete Command → Rows of table maybe deleted using a delete command.

Syntax is :- DELETE FROM <Table Name> WHERE <Condition>

eg :- DELETE FROM Employee WHERE Emp_code = 1

(iv) Select Command → Querying on the DB is done using the SQL command Select. It can be used for simple to complex queries

Syntax is :- SELECT <Column list> FROM <Table Name>

eg :- SELECT Emp_code, Name, Salary FROM Employees.

1-4-25

③ DCL Commands

(i) Grant → This command is used to give users access priviledges to the DB.

Syntax is :- GRANT <Privileges name (Table name)> ON <object name> to <user name>

eg :- GRANT SELECT ON Employees to user_id 123.

It means, the user whose Id is 123 gives the permission to read data from the employee table.

(ii) **Revoke** → This command is used to remove previously granted access privileges from a user.

Syntax is :- REVOKE <privileges name> ON <object (attribute) name> from <user name>

eg :- REVOKE SELECT ON Employees from User Id 123

It removes permission to read data from the employee table.

(iii) **Commit** → Commit commands is used for storing changes performed by a transaction. When a commit command is issued, it saves all changes since last commit used.

(iv) **Rollback** → This command is used for reverting changes performed by a transaction. It reverts all the changes since last commit used.

DATE: / /20
PAGE NO.

Set of Operations in SQL

Union, Intersection, Find

Functions or Aggregates Functions in SQL

An aggregate function is a function that performs a calculation on a set of values and returns a single value. These functions are often used with the GROUP BY CLAUSE of the SELECT statement.

The most commonly used SQL functions are — ① MIN() ② MAX() ③ COUNT() ④ SUM() ⑤ AVG()

① MIN() → It returns the smallest value within the selected column.

② MAX() → It returns the largest value within the selected column.

③ COUNT() → It returns the no. of rows in a set.

④ SUM() → It returns the total sum of the numerical column.

⑤ AVG() → It returns the average value of the numerical column.

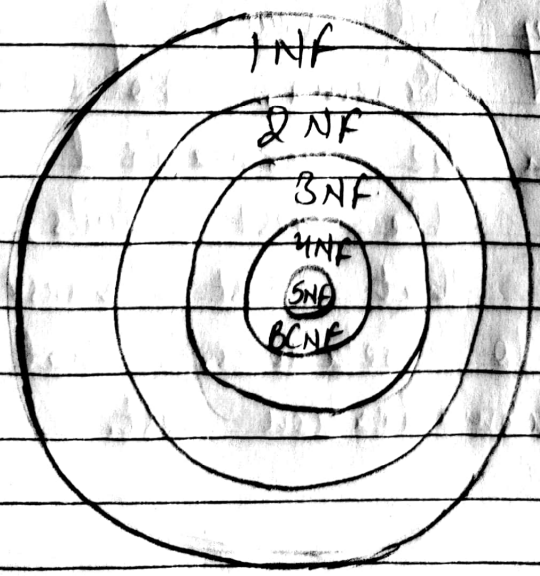
Unit-3

(Accessibility Consistency ACID property Integrity Durability)

Normalization in DBMS

It's a process of organising data in the DB to reduce redundancy and improve data consistency. Primary keys are really important in organising information in a DB. They help to make sure that every row in a table has a unique identification so that nothing gets mixed up or lost.

Normalization is a process of organising the data in the database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. There are 5 normal forms (5NF) in DBMS —



DATE: / /20
PAGE NO.

Objectives of Normalization

1. To create a formal framework for analyzing relation schemas, based on their keys and on the functional dependencies among their attributes.
2. To obtain powerful relational retrieval algorithms based on a collection of relational operators.
3. To free relations from undesirable insertion, updation & deletion anomalies.
4. To reduce the need for restructuring the relations as new data types are introduced.
5. To carry out series of tests on individual relational schema, so that the relational DB can be normalised to some degree. When the test fails, the relation violating that test must be decomposed into relations, that individually meet the normalization test.