Database Management System

Data : Data is a collection of a distinct small unit of information. It can be used in variety of forms like text, number, media. etc. it can be stored in a pieces of paper or electronic memory.

Word 'Data' is originated from the word 'datum' that means 'Single piece of information'.

In Computing, Data is information that can be translated into a form for efficient movement and processing

Database : A database is an organised collection of data, so that it can be easily accessed and managed.

DBMS : is a Software for storing and retrieving user's data while considering appropriate Security measures. It consist of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating System to provide the specific data.

Characteristics of DBMS :
Provides Security and removes redundancy.
Self-describing nature of a database System.
Support of multiple views of the data.
It follows ACID Concept.

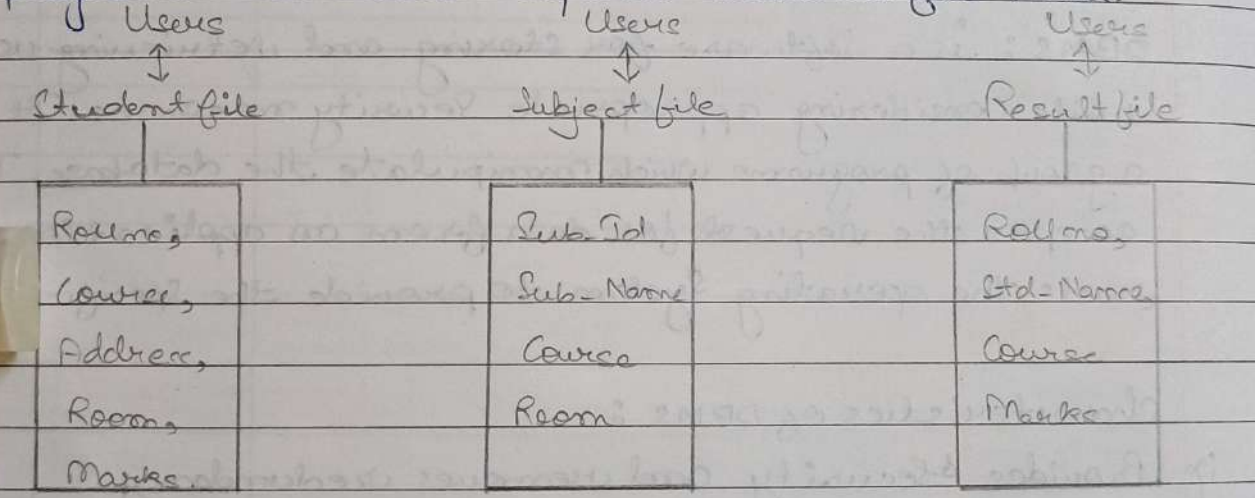
Advantage of DBMS :
It offers variety of technique to store and retrieve data.
Uniform administration and procedures of data.
Offers data Integrity and Security.
Reduced Application Development time.

Need of database :

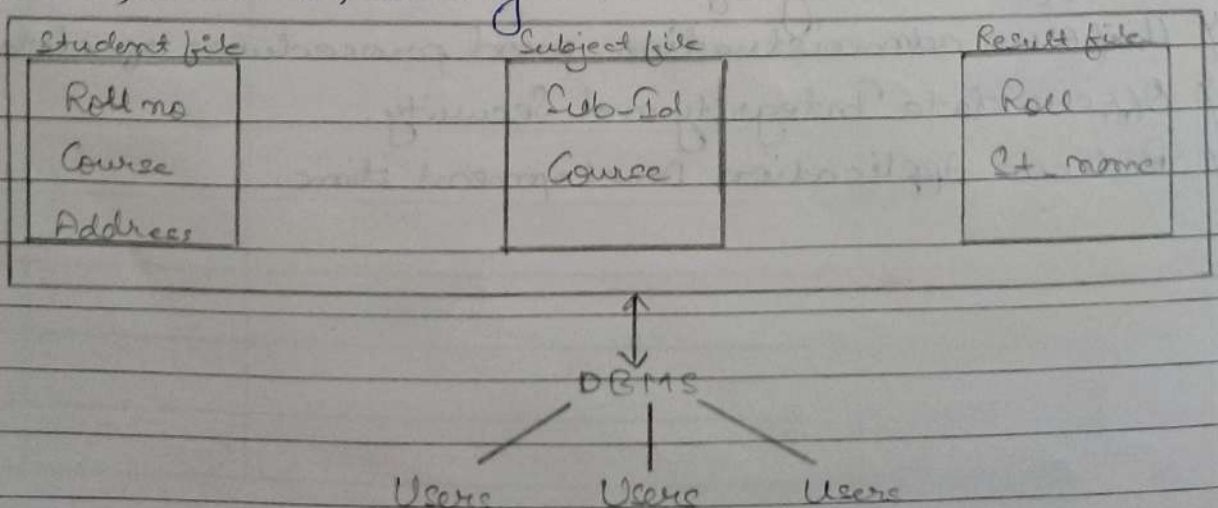
i) Creation of a database
ii) Retrieval of information from the database.
iii) Updating the database
iv) Managing a database.

File Management System and DBMS :

→ File Management Approach : File based systems were an early attempt to computerized the manual system. The main role of data processing specialist was to create the necessary computer file structures, and also manage the data within structures and design some application programs that create reports based on file data.



→ DBMS : A database approach is a well-organised collecti
of data that are related in a meaningful way which can
accessed by different users but stored only once in a s
The various operation performed by the DBMS are : inse
ion, deletion, Sorting etc.



FABER-CASTELL
Date
Page No.

Basis	DBMS Approach	File System Approach
Meaning	DBMS is a collection of data. In DBMS the user is not required to write the procedure.	The file System is a collection of data. In this System the user has to write the procedures for managing database.
Sharing of data	Due to the centralized approach data sharing is easy	Data is distributed in many files and it may be of different formats, so it isn't to share data.

→ Database Administrator?

In a modern IT organization, the database administrator is often ~~use~~ a set of responsibilities, rather a specific title. DBA manage database deployment topologies, data models and application access patterns. They are responsible for capacity planning and infrastructure provisioning.

DBA's also often ensure data security and Compliance ~~work~~ by working with security architects and Compliance officer

Functions are :

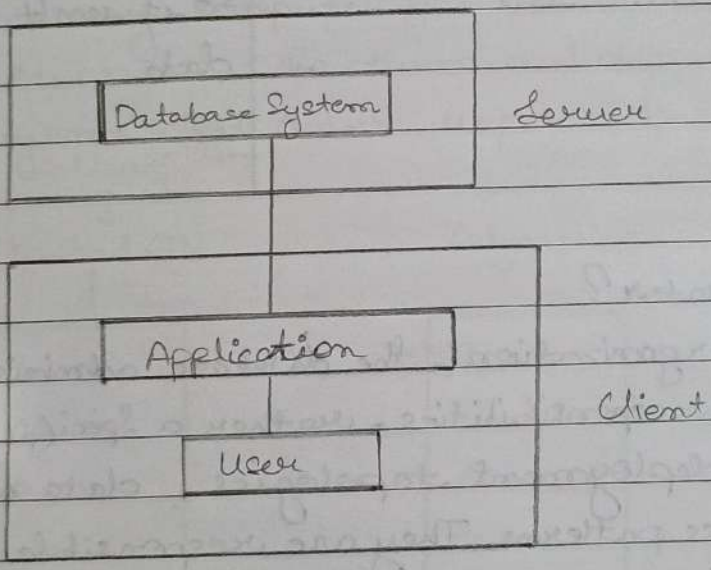
i) Implementing data policies, procedures, standards.
ii) Planning and development of an organization enterprise data models.
iii) Resolving data ~~models~~ conflicts.
iv) Managing data repositories.

Tier 2 and Tier 3 architecture :

1. Two-Tier Database Architecture : The 2 tier architecture is same as basic client-Server. In the two-tier architecture, applications on the client end can directly communicate with the database at the Server side.

The user interface and application programs are run on the client-side.

The Server side is responsible to provide the functionalities like : query processing and transaction management.

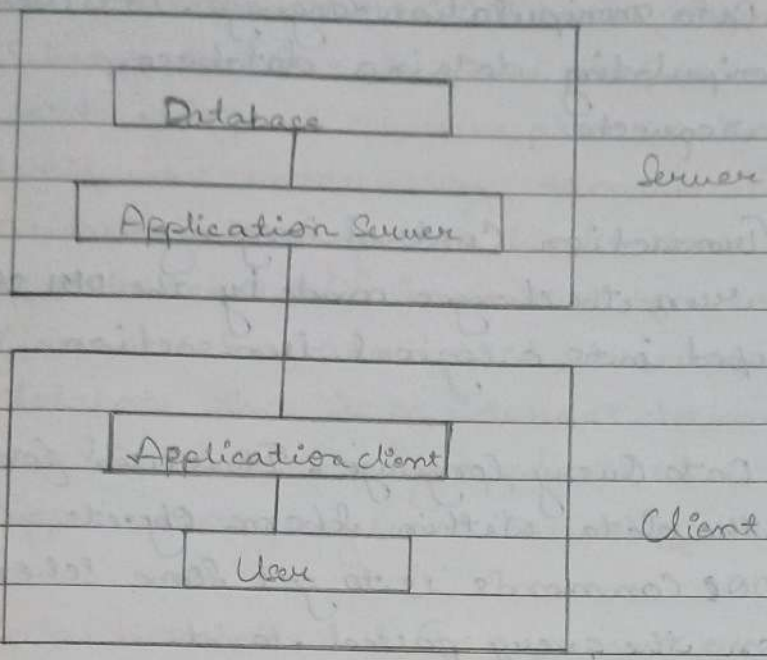


2. Three-tier Architecture : The 3 tier architecture contains another layer between the client and the Server. In this architecture, client can't directly communicate with the Server.

The application on the client end interacts with an application Server which further communicates with database System.

End user has no idea about the existence of the database beyond the application Server. The database also has no idea about any other user beyond the application.

The 3 tier architecture is used in case of large web application.



→ Database Language :

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and updat the data in the database.

Types of Database Langugs :

i) DDL
ii) DCL
iii) DML
iv) TCL
v) DQL

i) DDL stands for Data Definition Language. It is used to define database structure or pattern.

- It is used to create Schema, tables, indexes, constraints in the database.
- Using DDL statements, you can create the skeleton of the database.

ii) DCL stands for Data Control language. It is used to retrieve the stored or saved data.

- The DCL execution is transactional. It also has rollback parameter.

DML : Stands for Data manipulation Language. It is used for accessing and manipulating data in a database.
It handles user requests.

TCL : stands for Transaction Control language.
TCL is used to run the changes made by the DML statement.
TCL can be grouped into a logical transaction.

DQL stands for Data Query language. It is used for performing queries on the data within Schema Objects.
The purpose of DQL Commands is to get some schema relation based on the query passed to it.

Instances : It is the Snapshot of the database taken at a particular moment. It can also be described in more significant way as the collection of the information stored in the database at that particular moment. Instance can also be called as the database State or Current Set of occurence due to the fact that it is information that is present at the Current State.

Everytime we update the state Say we insert, delete, modify the value of the ~~dat~~ data item in the record, it changes from one state to other. At the given time, each Schema has its own set of instances.

Schema in DBMS : It is the overall description or the overall design of the database specified during the database design. Important thing to be remembered here is it should not be changed frequently. Basically, it displays the record types, name of data items but not the relation among the files.

Interesting point is the values in Schema might change but not the structure of Schema.

→ Sub Schema in DBMS : It can be defined as the subset of Schema that has the same properties as the schema. In simp words it is just effective plan or the Schema for the views. Well, it is interesting to note that it provides the us a window through which the users can view only that par of database which is matter of interest to him. It identi fies subset of areas, set, records, data names defined i database that is of interest to him. Thus a portion of data base can be seen by application programs and different application programs has different view of ~~data~~ data.

→ Data Abstraction : Is a process of hiding unwanted or irrelevant details from the end user. It provides a differen views and helps in achieving data independence which is used t enhance the Security of data.

The database consist of complicated data Str ctures and relations.

→ Levels of Abstraction :

i) Physical or Internal level
ii) Logical or Conceptual level
iii) View or External level.

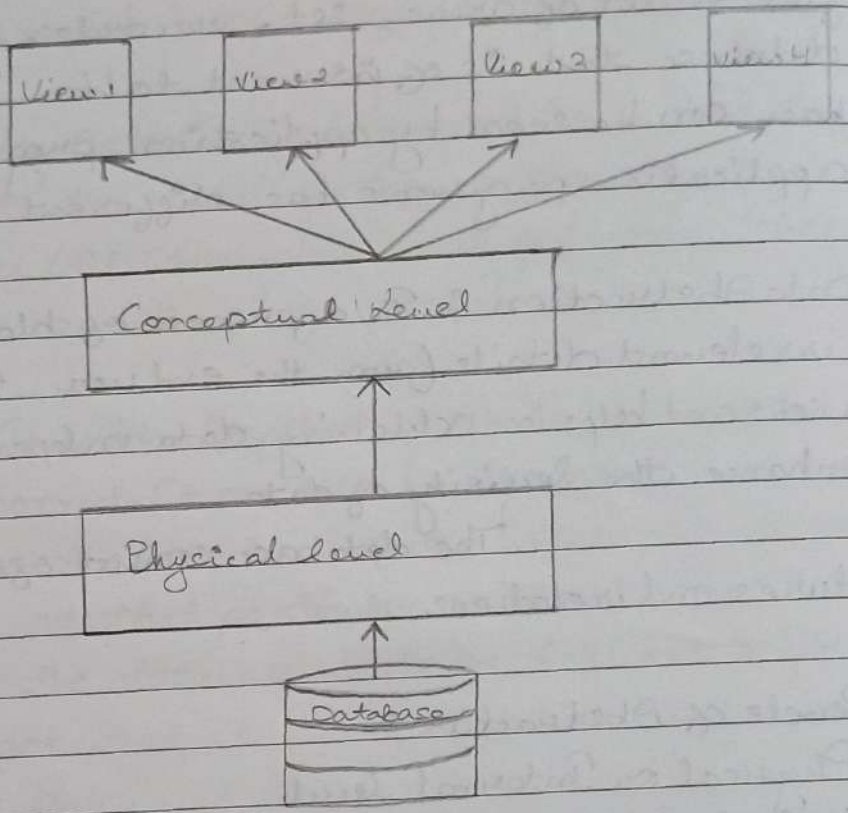
i) Physical or Internal level : It is the lowest level of abstrac tion for DBMS which defines how the data is actually Stored, it defines the data structures to store data and access methods used by database. Actually, It is decided by developers or database application programmers ~~to~~ ho to Store the data in the database.

ii) Logical or Conceptual level : Logical level is the next highe level. It describes what data is stored in the database and what relationship exists among those data. It tries to describes the entire or whole data because it describe wha tables to be created and what are links among those tables are Created.

View or External level: It is highest level. Various level there are different levels of views and every views only defines a part of the entire data. It also simplifies interaction with the user and it provides many ~~the~~ views or multiple view of the same database.

View level can be used by all users.

Diagram :-



Referential Integrity : It is a Rule in DBMS is based on Primary key and Foreign key. The Rule defines that a foreign key have a matching primary key.

<Employee>

Emp Id | Emp_Name | Dept_ID

<Department>

Dept_ID | Dept_Name | Dept_Zone

→ Data Independence : is the ability to modify the sche
Without affecting the programs and the application th
written. Data is seperated from the programs, so th
changes made to the data will not affect the program e
and the application.

We know the main purpose of the three l
data abstraction is to achieve data independence.

→ levels of data Independence :
i) Physical Data Independence
ii) logical Data Independence.

i) Physical Data Independence : means changing the phys
level without affecting the logical level or concep
level. By using this property, we can change storage
of the database without affecting logical Schema.

ii) Logical Data Independence : Logical a view of data is
user view of data. It presents data in the form of tha
be accessed by the end user.

Codd's Rule of logical DI says that
should be able to manipulate the logical view of da
without any information of its physical storage. Sof
Computer program is used to manipulate the logica
of ~~the~~ the data.

Database Administrator is the one ~~at~~ who
what information is to be kept in the database a
to use the logical level abstraction. It provides g
view of data. It also describes what data is to be
in the database along with the relationship.

→ what is RDBMS and how it is stored in memory? iif

:- It stands for Relational Database Management System. All modern database management system like SQL, ML, IBM DB2 etc are based on RDBMS.

It is called Relational Database Management System because it based on the relational model introduced by E.F. Codd.

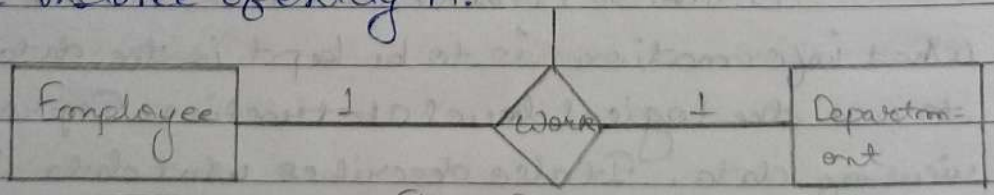
The memory ~~stage~~ storage that is directly accesible to the CPU comes under this category. CPU's internal memory, fast memory and main memory are directy accessible to the CPU, as they are placed on the motherboard or CPU chipset.

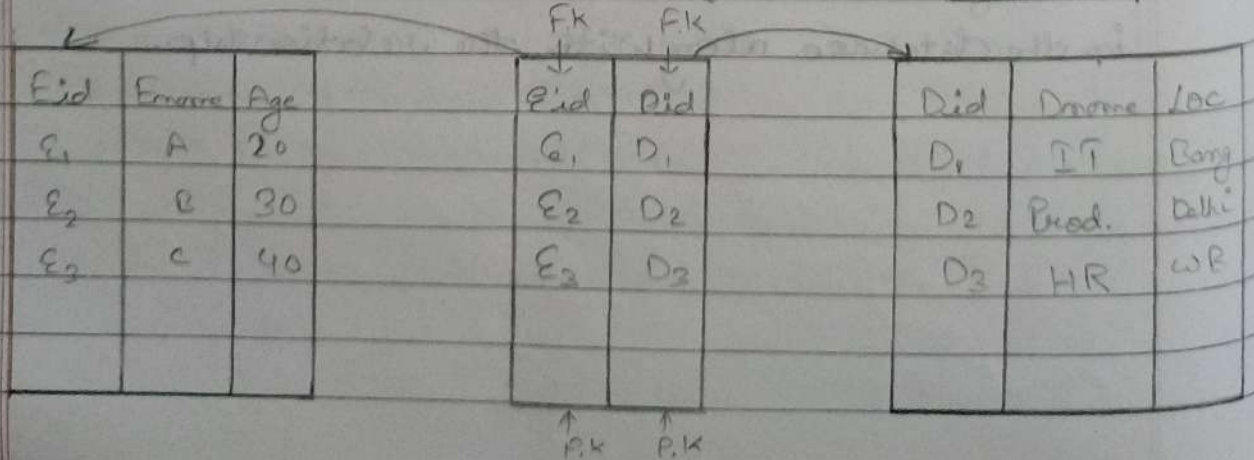
→ What is Relational means in RDBMS?

:- The relational model means that the logical data structures - the data tables, views, and indexes - are separate from the physical ~~strong~~ storage structures. This separation means that database administrators can manage physical data storage without affecting access to that data as a logical structure. iiif

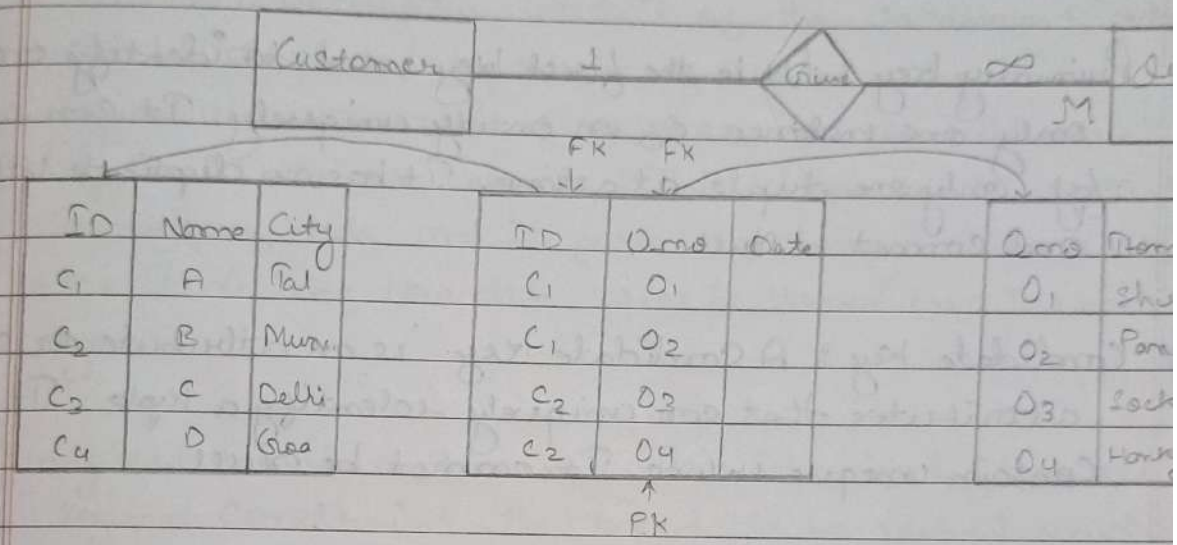
→ Degree of Relation :-

if 1:1 (one to one) : In RDBMS, a one-to-one relationship exists when zero or one instance of entity A can be associated with zero or one instance of entity B, and zero or one instance of entity B can be associated with zero or one instance of entity A.





ii) 1:M (One to many): In RDBMS design, a one to ... relationship exists when, for one instance of entity A ... exists Zero, one, or many instances of entity B; b... one instance of entity B, there exists Zero or one ins... entity A.

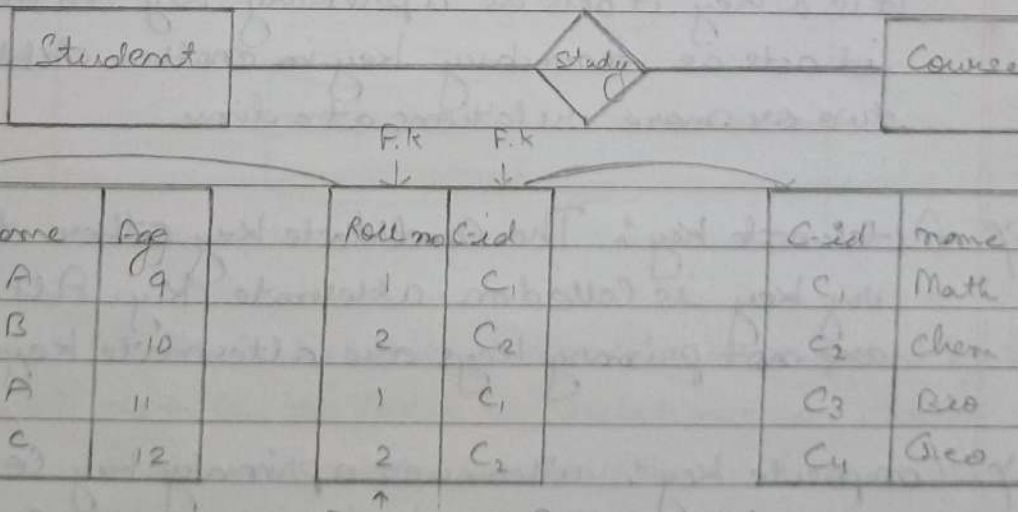


ID	Name	City
C_1	A	Tal
C_2	B	Mum.
C_3	C	Delhi
C_4	D	Goa

ID	O.no	Date
C_1	O_1	
C_1	O_2	
C_2	O_3	
C_2	O_4	

O.no	Item
O_1	Sho...
O_2	Pan...
O_3	Sock...
O_4	Hank...

iii) M:M (Many to Many): In RDBMS design, a many-to... relationship exists when, for one instance of entity A ... exists Zero, one, or many instances of entity B; or ... one instance of entity B, there exists Zero, one, many ... ces of entity A.



Rollno	rname	Age
1	A	9
2	B	10
3	A	11
4	C	12

Rollno	C.id
1	C_1
2	C_2
1	C_1
2	C_2

C.id	name
C_1	Math
C_2	Chem...
C_3	Bio
C_4	Geo...

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 relation between tables.

Primary key: It is the first key used to identify one and only one instance of an entity uniquely. It can identify only one tuple at a time. It has no duplicate values, and cannot be Null.

Candidate key: A Candidate key is an attribute or set of attributes that can uniquely identify a tuple. It must contain unique values. It cannot be null.

Super key: Super key is an attribute set that can uniquely identify a tuple. A Super key is a super set of a Candidate key. It supports null values.

Foreign key: Foreign key are the column of the tables used to point to the primary key of another table. It is a key it acts as a primary key in one table and it acts as a secondary key in another table. It combines two or more relations at a time.

Alternate key: The Candidate key other than the primary key is called an alternate key. All the keys which are not primary keys are alternate keys.

Composite key: Whenever a primary key consist of more than one attribute, it is known as a Composite key. This key is also known as Concatenated key. (Combination of col

Artificial key: An artificial key is one that was created specifically to be the unique identifier of a row.

→ Database Struct database relat define contai can b

i) Physi Stora etc. age.

ii) Logi log stor

→ R

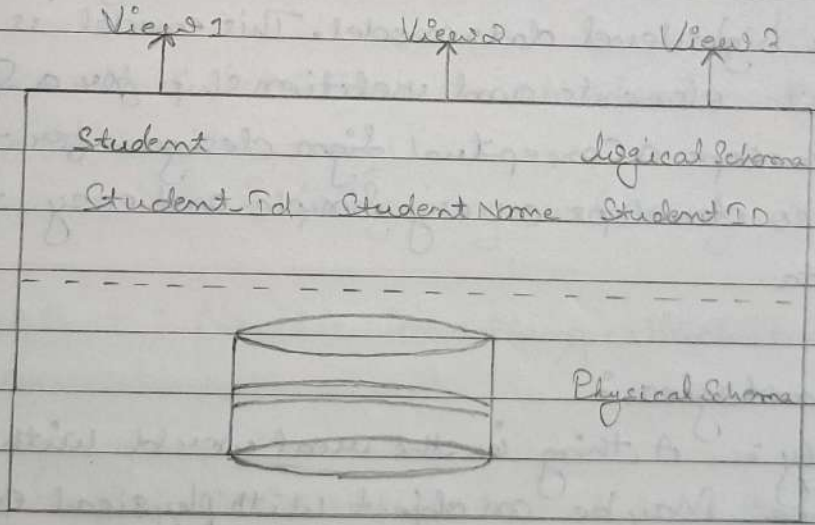
i)

→ Database Schema : A database Schema is the Skeleton Structure that represents the logical view of the entire database. It defines how the data is organised and how the relation among them are association. A database Schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which which can be depicted by means of schema diagram.

→ Logical representation of database is called Schema.

i) Physical Database Schema : This Schema pertains to the actual Storage of data and its form of storage like files, indices etc. It defines how data will be stored in a Secondary storage.

ii) Logical Database Schema : This Schema defines all the logical constraint that need to be applied on the data stored. It defines tables, views and integrity constraint



→ Relational Model : Relational model was proposed by E.F Codd to model data in the form of relations or tables.

Relational model represents how data is stored in Relational Databases. A relational Database stores data in the form of relation.

Terminologies :

i) Relational model represents data as a collection of tables

iii) A table is also called on relation.
iv) Each row is called tuple.
v) Column header is called attributes.
vi) A set of atomic values allowed for on attribute is called Domain.
vii) Relation Schema describes a relation. Made up of a relation name R and a list of attributes $A_1, A_2, A_3 \ldots A_n$
viii) The number of attributes in the relation is known as degree of ~~the~~ relation. example - 6
ix) Cardinality is total number of tuples present in a relation.
x) Relational database Schema Is a set of relation Schema and a set of integrity Constraints.
xi) Relation State or Relation instance is a set of tuples at a given time.

→ ER Model: ER model stands for Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relation ship for a specified System. It develops a Conceptual ~~Sign~~ design for the database. It also develops a very simple and easy to design view of data.

Terminologies:

a) Entity: A thing in the real world with an independent existence. May be an object with physical existence or with a conceptual existence.

b) Attributes: Properties that describe the entities.

Person :- i) Name
ii) Age
iii) Address
iv) Phone Number

a) Composite attributes: Can be divided into further parts
Ex- Name → First name, Middle name, Last Name

b) Simple attribute : Cannot be divided further.
Ex- Weight -> cannot be of further divided.

c) Single Valued attribute : Have a single value for particular entity.
Ex : Age = Single Valued attribute of a person.

d) Multivalued attribute : Can have set of values for particular entity.
Ex : College degree, language known.

e) Derived attribute : Can be derived from other attribut
Ex : Age : Can be derived from date of birth.

f) Stored attributes : From which the value of other attributes are derived.
Ex : Birth Date of a person.

g) Complex attributes : Has multivalued and composite components in it
Ex : { College Degree (College, Year, Degree) }

h) Null Values : Null is something which is not applicable or unknown.

i) Entity Type : A Collection of entities that have the same attributes. Ex : Student.

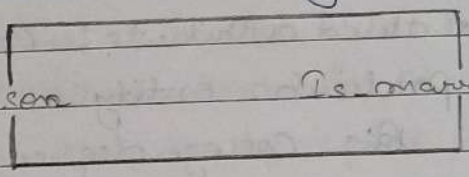
j) Entity Set : Collection of entities of a particular entity type at a point in time.

k) Key attribute : That attribute that is capable of identifying each entity uniquely. Ex- rollno.

c) Relationship : association among 2 or more entities.
Ex: teacher teaches Student
relationship

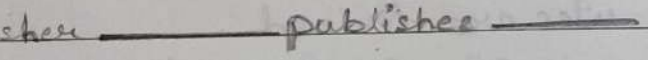
a) Degree of Relationship : Denotes the number of entity types that participate in a relationship.

i) Unary relationship : Exists when there is an association with only one entity.

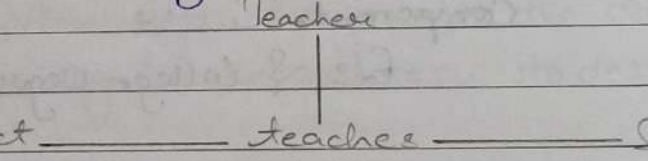


DOR = 1

ii) Binary relationship : Exists when there is an association among two entities.



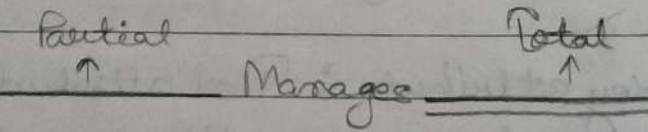
iii) Tertiary relationship : Exists when there is an association among three entities.



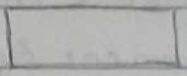
b) Relationship Constraints :

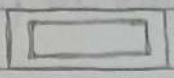
i) Cardinality Ratio : Maximum number of relationship instances that an entity can participate in

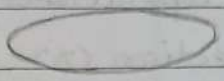
ii) Participation Constraints : Specifies whether existence of an entity depends on its being related to another entity.
Types : Total and Partial Participation.



→ Symbols used in ER diagram :-

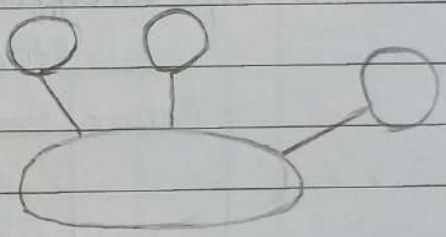
i) Entity → 

ii) Weak Entity → 

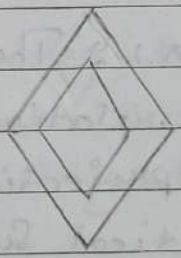
iii) Attribute → 

iv) Key attribute → 

v) Multivalued attribute → 

vi) Composite attribute → 

vii) Derived Attribute → 

viii) Identifying Relationship → 

= Weak & for strong only single line.

→ ER model to Relational model :-
Some methods are automated & some of them are manual.
It mainly comprises of :-
i) Entity and its attributes
ii) Relationship which is association among entities.

Mapping is the process to convert ER to relational model.
i) Create table for each entity.
ii) Entity attribute should become fields of tables with respective datatypes.
iii) Declare Primary Key.

→ Relational Algebra : is a widely used procedural query language. It collects instances of relations as input and gives occurrences of relations as output. It uses various operations to perform this actions.

Operators : Basic Operators :
- i) Projection (π)
- ii) Selection (σ)
- iii) Cross product (×)
- iv) Union (∪)
- v) Rename (ρ)
- vi) Set difference (−)

Derived Operator
- i) Join (⋈)
- ii) intersect (∩)
- iii) Division (/, ÷)

Student

Rollno	Name	Age
1	A	20
2	B	21
3	A	19

i) Projection (π) : The projection eliminates all attributes of the input relation but those mentioned in the projection list. The projection method defines a relation that contains a vertical subset of Relation

Query : Retrieve the rollno from table (student)

→ $\pi_{Rollno, Name}(Student)$

ii) Selection (σ) : The Select operation is used for selecting a subset of the tuples according to a given selection condition.

$\sigma_p(r)$

- σ is the predicate
- r Stands for relation which is the name of the table.
- p is the prepositional logic.

Query: Retrieve the name of student whose Rollno = 2

→ $\pi_{name}(\sigma_{Rollno=2}(Student))$

iii) Cartesian Product: is used to merge columns from two relations.

A	B	C
2	1	3
1	2	3

R1

C	D	E
3	4	5
9	7	9

R2

attribute = m+n

Tuples = x x y

A	B	C	C	D	E
2	1	3	3	4	5
2	1	3	9	7	9
1	2	3	3	4	5
1	2	3	9	7	9

$R_1 \times R_2$

∴ At least one Column should be same

iv) Set difference :- Symbol denotes it. The result of A-B is a relation which includes all tuples that are in A but not in B. Attribute name of A has to match with the attribute name in B.

$A - B \neq B - A$

$(A-B)$ = A but not B

$= A \cap B$

Rollno	Name
1	A
2	B
3	C

EmpNo	Name
7	E
1	A

=

Rollno	Name
2	B
3	C

Student - Employee

Query: Name of person who is Student but not employee

→ $(\pi_{name}(Student) - \pi_{name}(Employee))$

Date
Page No.

* Union : It denote by ∪. It includes all tuples that are in tables A and B. It also eliminate duplicate tuples.

Rollno	Name
1	A
2	B
3	C

Student

Empno	Name
7	E
1	A

Employee

=

Rollno	Name
1	A
2	B
3	C
7	E

(Student) ∪ (Employee)

* Division : /, ÷ by the help of ×, −, π, σ

Sid	Cid
S_1	C_1
S_2	C_1
S_1	C_2
S_3	C_2

'Enrolled' E

Cid
C_1
C_2

'Cource' c

Query : Retrieve Sid of students who enrolled in every Cour

$\pi_{Sid}(Enrolled) - (\pi_{Sid}((\pi_{Sid}(Enrolled)) \times \pi_{Cid}(Course) - (Enrolled)))$

> SQL :
- Stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL became a Standard of the American National Standard Institute in 1986, and ISO in 1987.

- It can execute queries against a database
- Can retrieve data from a database
- Can insert records in a database
- Can create rows in a database.
- Create new tables in a database.
- Can create views in a database.
- Can set permission on tables.

→ SQL vs My SQL :

Structured Query language is the standard language used to operate, manage and access database. By making minor changes in the Syntax, you can add, retrieve, delete data in different databases. The ANSI maintains that SQL is the standard language for managing a relational database management System such as My SQL. In SQL you can use single command to access multiple records in database

My SQL was developed way back in 1995 by My SQL AB. Now it is owned and offered by Oracle Corporation. It is an open source relational database management System that uses SQL Commands to perform specific functions/ Operations in a database. It is written in C and C++ programming languages. It is also a core element of the Open source technology stack, LAMP (linux, Apache, My SQL, PHP). My SQL offers a multi user access to databases.

→ SQL Join : Statement is used to combine data or rows from two or more tables based on common field between them.

i) Natural Join : In both table when there is some or common attribute having same name then we use natural join and when we equal these two attribute this is called Natural join. Join = Cross product + Condition

ii) Inner Join : This keyword selects rows from both the tables as long as condition is satisfied. This keyword will create the result-set by combining all rows from both the table where condition satisfies and both table common in them.

iii) left Join : This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there

is no matching row on the right side, the result set will contain null.

iv) Right Join: This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For rows for which there is no matching row on the left side, the result set will contain null.

v) Full Join: Creates the result-set by combining results of both left join and right join. The result set will contain all rows from both tables. For the rows for which there is no matching, the result-set will contain null values.

vi) Self Join: In Self Join a table is joined to itself. That is, each row of the table is joined with itself and all other rows depending on some conditions. In other words we can say that it is join between two copies of the same table.

→ What is View: Views in SQL are kind of virtual tables. A View also has rows and columns as they are in a real table in database. We can create a view by selecting fields from one or more tables present in database. A view can either have all the rows of a table or specific rows based on certain condition.

The table upon which a view is constructed by view is called base table.

i) It doesn't contain any data itself.

ii) Views are used for security purpose because they provide encapsulation of the name of table.

iii) Data is in the virtual table not stored permanently.

iv) View can display only selected data.

v) View is also known as stored select statement.

Types of View :
i) Simple View
ii) Complex View
iii) Force View
iv) Read only View
v) With check ~~only~~ option View.

→ what is Trigger : A trigger is a stored procedure in base which automatically invokes whenever a Specia in the database occurs. Ex:- Insert, update etc.

→ Primary key vs Unique key :
Primary key is a column that is used to uniquely iden each tuple of the table. Only one primary key is allowe be used in a table. Duplicate and Null values are not in the case of the primary key.

Unique key is a constraint that is used to unique identify a tuple in a table. Multiple unique keys can p in a table. Null values are allowed in case ~~of~~ of a key. These can ~~hold~~ also be used as foreign keys for table.

→ SQL Injection : SQL injection is a code injection t we that might ~~destr~~ destroy your database. It is the most common web hacking techniques. SQL inj is the placement of malicious code in SQL statemen web page input.

→ Delete vs Truncate :
Delete : i) The delete statement is used to remove or multiple thread records from an existing tab rding on the specified condition.
ii) It is a DML Command.
iii) We need to have delete permission to use this
iv) This command eliminates record one by one.
v) This command does not reset the table ident

because it only deletes the data.
vi) It maintains transaction logs for each deleted record.
viii) Its speed is slow because it maintained the log.

Truncate: i) The truncate command removes the complete data from an existing table but not the table itself.
ii) It is a DDL.
iii) We need to have alter permission to use this comm and.
iv) It always resets the table identity.
v) It does not maintain transaction logs for each deleted data pages.
vi) Its execution is fast because it deleted entire data at a time without maintaining transaction logs.

Grant Privilege:
MySQL has a feature that provides many control options to the administrators and users on the database.

The grant statement enables System administrators to assign privileges and roles to the MySQL user account so that they can use the assigned permission on the database whenever required.

Syntax:

```
GRANT privilege_name(s)
ON Object
TO user_account_name;
```

Revoke Privilege:
MySQL has a feature Revoke statements to remove privileges from an user account.

The revoke statement enables System administrators to revoke privileges and roles to MySQL user account so that they cannot use the assigned permission on the database in the past.

Syntax:

→ Sub Query
a query
WHERE c
used in
data to
delete sta
→ Indexing
quickly
a small
rises a
cond Colu
of the
I/O Cos
Types o
i) Prim
ii) Clust
iii) Second
i) Primar
fixed l
primar
data b
entries
Dense
in the
more S

Syntax:

REVOKE privilege_name(s)
ON Object
FROM user_account_name.

→ SubQueries: A Subquery or Inner query or a Nested a query within another SQL query and embedded wit WHERE clause.

A Subquery is used to return data that w used in the main query as a condition to further rest data to be retrieved.

Subquery used with Select, insert, up delete statement.

→ Indexing: is a data structure technique which allows y quickly retrieve records from a database file. An i a small table having only two columns. The first colum rises a copy of the primary or candidate key of a table. cond column contains a set of pointers for holding th of the ~~disk~~ disk block where that specific key value s I/O cost is reduced by Indexing. It also reduces calling of m

Types of Indexing:

i) Primary: a) Dense b) Sparse
ii) Clustering
iii) Secondary

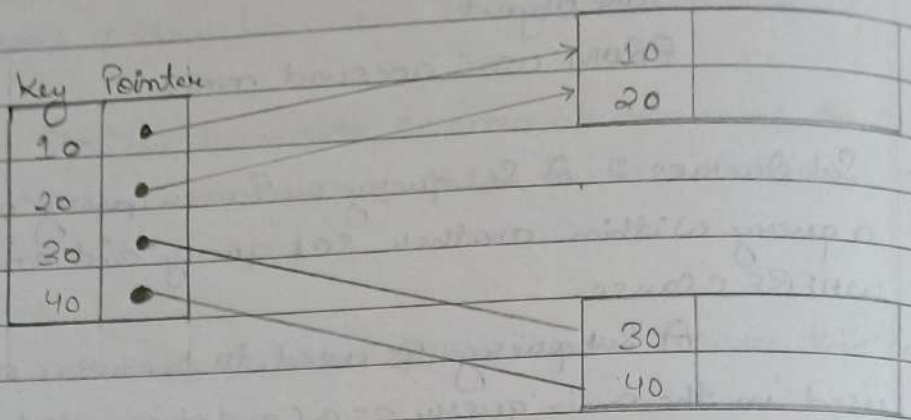
	Ordered file	Primary Index	Clu
	Unordered file	Secondary Index	Sec

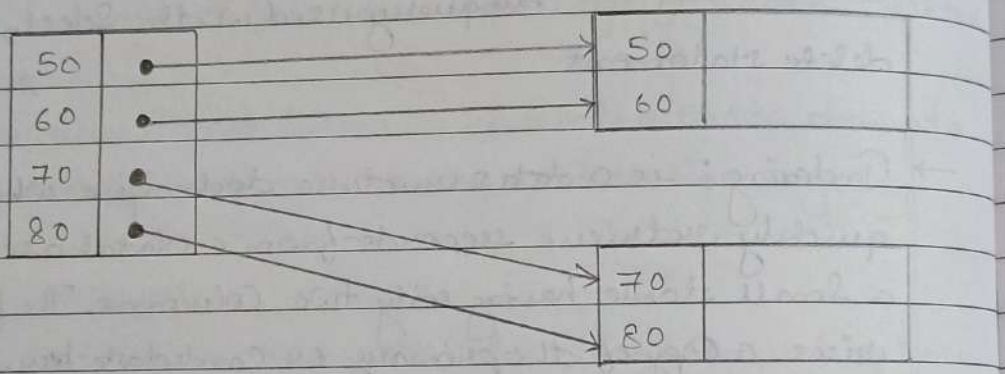
Key

i) Primary Index: Primary index is an ordered file whic fixed length size with two fields. The first field is the sa primary key and second, ~~filed~~ field is pointed to that speci data blocks. There is one to one relationship betwee entries in the index table.

Dense index: a record is created for every search key in the database. This helps you to search faster but re more space to store index records. In this, method

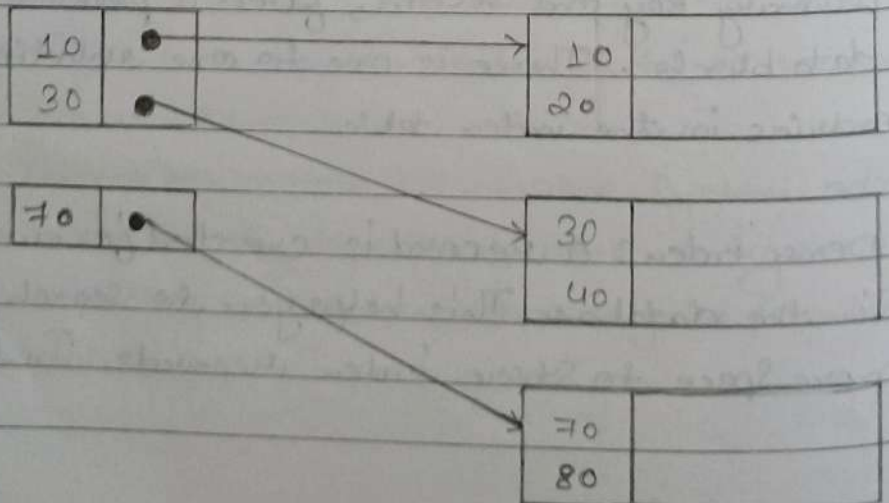
Contain search key value and points to the real records on the disks.





Sparse index: It is an index record that appears for only some of the values in the file. Sparse index helps you to resolve the issue of dense Indexing in DBMS. In this, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.

Sparse index stores index records for only some search key values. It needs less space, less maintenance overhead, for insertion, and deletion but It is slower compared to the dense index for locating records.

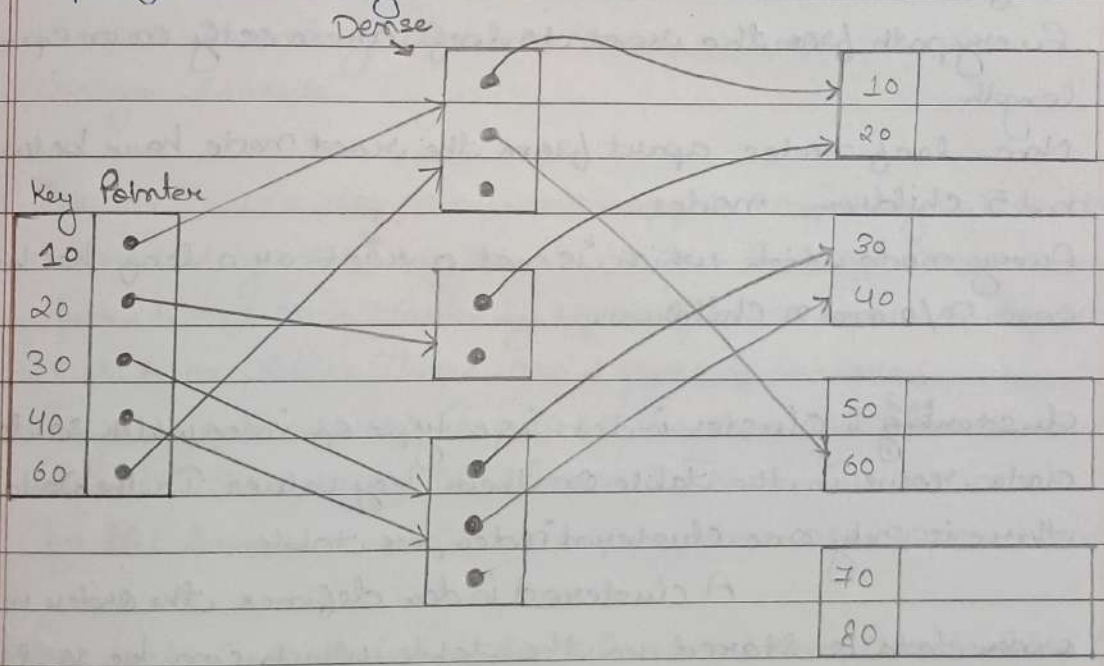


→ Secondary Index: It can be generated by field which has a unique value for each record, and it should be a candidate key. It is also known as non-clustering index.

This two level database indexing technique is used to reduce the mapping size of the first level. For the first level, a large range of numbers is selected because of this; the mapping size always remains small.

In a bank account database, data is stored sequentially by acc-no; you may want to find all accounts in of a specific branch of ~~AB~~ ABC bank.

Here, you can have a secondary index in DBMS for ~~e~~ every search-key. Index record is a record point to a bucket that contain pointers to all the records with their specific search key value.



→ Clustering Index: In a clustered index, records themselves are stored in the index and not pointers. Sometimes the index is created on non-primary key columns which might not be unique for each record. In that situation, you can group two or more columns to get the unique values and create an index which is called clustered index.

In this Block hanger concept is also used for searching records.

→ Multilevel Index : Multilevel Indexing in Database is created when a primary key index does not fit in memory. In this type of indexing method, you can reduce the number of disk accesses to short any record and kept on a disk as a Sequential file and create a Sparse base on that file.

→ B-Tree Index : B-tree Index is widely used data structure for tree based indexing in DBMS. It is a multilevel format of tree based indexing in DBMS technique which has balanced binary search trees. All leaf nodes of the B tree signify actual data pointers.

Moreover, all leaf nodes are interlinked with a linked list, which allows a B tree to support both random and Sequential access.

- Leaf nodes must have between 2 and 4 values.
- Every path from the root to leaf are mostly on an equal length.
- Non-leaf nodes apart from the root node have between 3 and 5 children nodes.
- Every node which is not a root or a leaf has between n/2 and n children.

→ Clustered : Cluster index is a type of index which sorts the data rows in the table on their key values. In the Database there is only one clustered index per table.

A clustered index defines the order in which data is stored in the table which can be sorted in only one way.

- Allows you to stores data pages in the leaf nodes of the index.
- The size of the clustered index is quite large.
- Faster Data access.
- Not Required additional disk Space.

→ Non-clustered index: A non-clustered index stores data at one location and indices at another location. The index contains pointers to the location of that data. A single table can have many non-clustered indexes as an index in the non-clustered index is stored in different places.
Example: a book can have more than one index, one at beginning which displays the contents of a book unit wise while the second index shows the index of terms in alphabetical order.

An non clustering index is defined in the non-ordering field of the table. This type of indexing method helps you to improve the performance of queries that use keys which are not assigned as a primary key. A non-clustered index allows you to add a unique key for a table.

- Stores key values only.
- Pointers to Heap/ Clustered rows.
- Allows secondary data access.
- Bridge to data.

→ Cursor: is a temporary memory or temporary work station. It is allocated by database server at the time of performing DML operations on a table by user. Cursors are used to store database tables. There are 2 types of cursors:-

- Implicit Cursors: Implicit cursors are also known as default cursors of SQL server. These cursor are allocated by SQL server when the user performs DML operation.

- Explicit Cursor: Explicit cursors are created by users whenever the user requires them. Explicit cursor are used for fetching data from table in Row-by-Row manner.

→ Relational Database features:
i) Tabular Structure of indexed file data.
ii) System Catalog
iii) SQL DDL Support
iv) SQL DML Enhancements
v) Multi-level Security

vi) Enhanced Performance

→ Functional Dependency : is a constraints that determines the relation of one attribute to another attribute in a DBMS. Functional Dependency helps to maintain the quality of data in the database. It plays a vital role to find the difference between good and bad database design.

A functional dependency is denoted by an arrow "→". The functional dependency of X on Y is represented by X → Y.

Key terms :

- Axiom : Axioms is a set of ~~int~~ inference rules used to infer all the functional dependencies on a relational database.
- Decomposition : It is a rule that suggests if you have a ~~tabl~~ table that appears to contain two entities which are determined by the same primary key then you ~~should~~ should consider breaking them up into two different tables.
- Dependent : It is displayed on the right side of the functional dependency diagram.
- Determinant : It is displayed on the left side of the functional dependency diagram.
- Union : It suggest that if two tables are separate, and the PK is same, you should consider putting them together.

Rules :

- Reflexive rule : If x is a set of attributes and Y is subset of x, then x holds a value of y.
- Augmentation rule : When x → y holds and c is attribute set, then ac → bc also holds. That is adding attributes which do not change the basic dependencies.
- Transitivity rule : This rule is very much similar to the transitive rule in algebra if x → y holds and y → z holds, then x → z also holds. x → y is called a functionally that determines y.

Types:

i) Multivalued :- It occurs in the situation where there are multiple independent multivalued attributes in a single table. A multivalued dependency is a complete constraints between two sets of attributes in a relation. It requires that certain tuples be present in a relation.

Ex: Car_model → maf_year
Car_model → Colour

ii) Trivial :- The trivial dependency ~~of~~ is a ~~subset~~ of ~~x~~ attributes which are called trivial if the set of attributes are included in that attribute.
So $x \to y$ is a trivial functional dependency if y is a subset of x. It is valid and always true.

Ex: E_id → E_id $x \to y$ and y is a subset of x.
x y

iii) Non-Trivial :- Functional dependency which also known as a non trivial dependency occurs when $A \to B$ holds true where B is not a subset of A. In a relationship, if attribute B is not a subset of attribute A, then it is considered as a non-trivial dependency.

Ex: $x \to y$ $x \cap y = \phi$
Sid → Sname
Sid → Semester
Sid → Phoneno.

iv) Transitive : A transitive Dependency is a type of functional dependency which happens when 't' is indirectly formed by two functional dependencies.
Note: You need to remember that transitive dependency can only occur in a relation of three or more attributes.

v) Fully functional dependency : Fully functional dependent on another attribute, if it is functionally dependent on that attribute and not any of its proper subset.

Ex: $ABC \rightarrow D$ { D is fully FD on ABC }
{ D cannot depend on any subset of AB

$BC \rightarrow D$ } not possible because
$C \rightarrow D$ } BC cannot determine D
$A \rightarrow D$ } C cannot determine D
A cannot determine D.

vi) Partial Dependency : It occurs when a non-prime attribute is functionally dependent on part of a candidate key.

→ Normalization : is a database design technique that reduces data redundancy and eliminates undesirable characteristics like insertion, Update and Deletion. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of normalisation in SQL is to eliminate redundant data and ensure data is stored logically.

Purpose of Normalization :

i) Prevent the same data from being stored in many places
ii) Prevent Updates made to some Data and not others
iii) Prevent deleting unrelated Data.
iv) Ensure queries are more efficient

3 anomalies solved by normalization are :

i) Insert Anomaly : An insert anomaly occurs when in related database when some attributes or data items are to be inserted into the database without existence of other attributes. For example, In Student table, if we want to insert a new Course ID, we need to wait until the student enrolled in a Course. In this way, it is difficult to insert new record in the table. Hence it is called insertion anomalies.

ii) Update Anomalies : The anomaly occurs when duplicate data is updated only in one place and not in all instances. Hence, it makes our data or table inconsistent state. For example, Suppose there is a student 'James' who belongs to Student table. If we want to update the course in the Student, we need to update the same in the Course table ; otherwise, the data can be inconsistent. And it reflects the changes in a table with updated values where some of them will not.

iii) Delete Anomalies : An anomaly occurs in a database table when some records are lost or deleted from the database table due to the deletion of other records. For example, if we want want to remove Trent Boult from Student table, it also removes his address, course and other details from the Student tables. Therefore, we can say that deleting some attributes can remove other attributes of the databases table.

Types of Normalization :

i) First Normal form (1NF) : The table will be 1NF if all the attributes of the table contain only atomic values. We can also say that if a table holds the multivalued data items in attributes or Composite values, the relation cannot be in the first normal form. So we need to make it first normal form by making the entries of the table atomic.

Table should not contain any multivalued attributes.

Roll	Name	Course
1	Gaurav	C/C++
2	Tripti	Java
3	Kajal	C/DBMS

Not in 1NF

=

Roll	Name
1	Gaurav
2	Tripti
3	Kajal

Roll	Course
1	C
1	C++
2	Java
3	C
3	DBMS

In 1NF

ii) Second Normal Form (2NF):
- The table or relation should be in 1NF or first Normal form.
- All the non-prime attributes should be fully functionally dependent on the Candidate key.
- The table should not contain any partial dependency.

iii) Third Normal Form (3NF):
- The table or relation should be in 2NF.
- It should not contain any transitive dependency. A transitive dependency is that any non-prime attribute determines or depends on the other non-prime attribute.

A relation is in 3NF if FD x determines Y ('X' -> 'Y') Satisfies one of following condition:
1. If $x \rightarrow Y$ is a trivial FD, ie, Y is a subset of x.
2. If $x \rightarrow Y$, where x is a Super Key.
3. If $x \rightarrow Y$, $(Y - x)$ is a prime attribute.

iv) BCNF: It stands for Boyce Codd Normal form, which is the next version of 3NF. Sometimes, it is also pronounced as 3.5NF. A normal form is said to be in BCNF if it follows the given conditions:
A table or relation must be in 3NF.
If a relation R has functional dependencies and if A determines B, where A is a Super Key, the relation is in BCNF.

→ Storage System: Databases are stored in file formats, which contains records. At physical level, the actual data is stored in electromagnetic format on some device. These storage devices can be broadly categorized into three types:-
- Primary Memory
- Secondary Memory
- Tertiary Memory.

- Primary Storage : The memory storage that is directly accessible to the CPU comes under this category. CPU internal memory, fast memory and main memory are directly accessible to the CPU as they are placed on the motherboard.

- Secondary Storage : Secondary storage devices are used to store data for future use or as backup. Secondary storage includes memory devices that are not a part of the CPU chipset.

- Tertiary Storage : Tertiary storage is ~~used~~ used to store ~~huge volumes of data~~ huge volumes of data. Since storage devices are external to the Computer System, they are the slowest in speed. These storage devices are mostly used to take the back up of entire System.

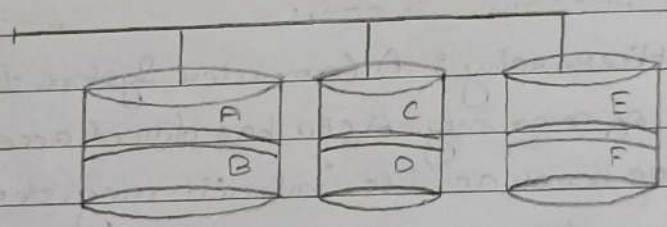
→ Memory Hierarchy : A Computer System has a well-defined hierarchy of memory. A CPU has direct access to main ~~Computer~~ memory as well as its inbuilt registers. To minimize this speed mismatch, a cache memory is ~~int~~ introduced. Cache memory provides the fastest access time and it contains data that is most frequently accessed by CPU.

→ Magnetic Disks : Hard drive are the most common Secondary storage devices in present Computer System. These are called magnetic disks because they use the concept of magnetization to store ~~inf~~ information. Hard disk consists of metal disks coated with magnetizable material. These disks are placed vertically on spindle. A read/write head moves in between the disks and used to magnetize or de-magnetize the spot under it.

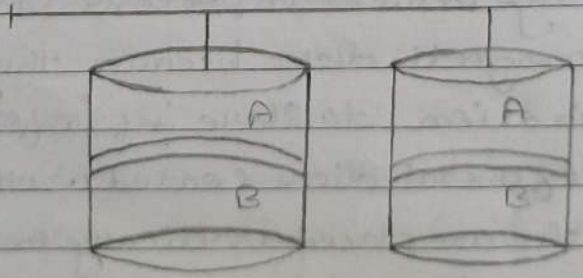
Hard disks are formatted in well defined order to store data ~~efficiently~~ efficiently. A hard ~~disk~~ disk plate ~~has many~~ has many concentric circles on it, called tracks. Every tracks are further divided into Sectors.

→ Redundant Array of Independent Disks :
Raid or Redundant Array of Independent Disks, is a tec
hnology to connect multiple Secondary Storage devices
and use them as a Single Storage media.
Raid Consists of an array of disks in which multi
iple disks are connected together to achieve different goals
Raid defines the use of disk arrays.

Raid 0 : In this level, a striped array of disks is implemen
ented. The data is broken down into blocks and the
blocks are distributed among disks. Each disk receives a block
of data to write/read in parallel. It enhances the Speed and
performance of the storage device. There is no parity and
backup in level 0. Data ko todna. It gives performance.

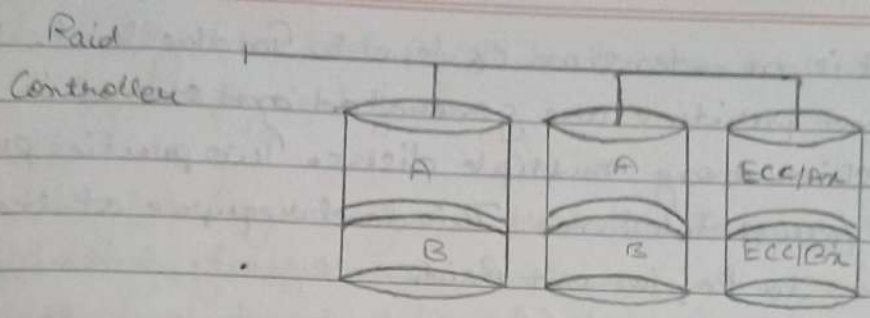


Raid 1 : Raid 1 uses as mirroring technique. When data is sent
to a Raid Controller, it sends a copy of data to all disks in
the array. Raid level 1 is also called mirroring and provi-
des 100% redundancy in case of a failure.

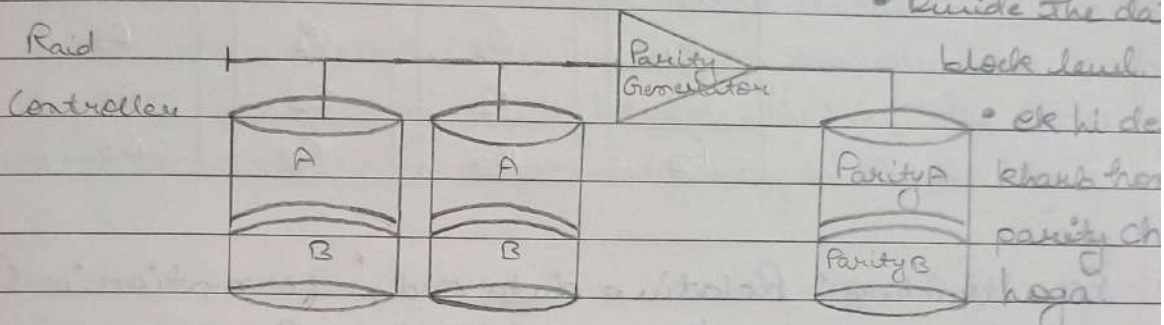


• Mirroring
• Copy banaiye
• It gives availabi
lity

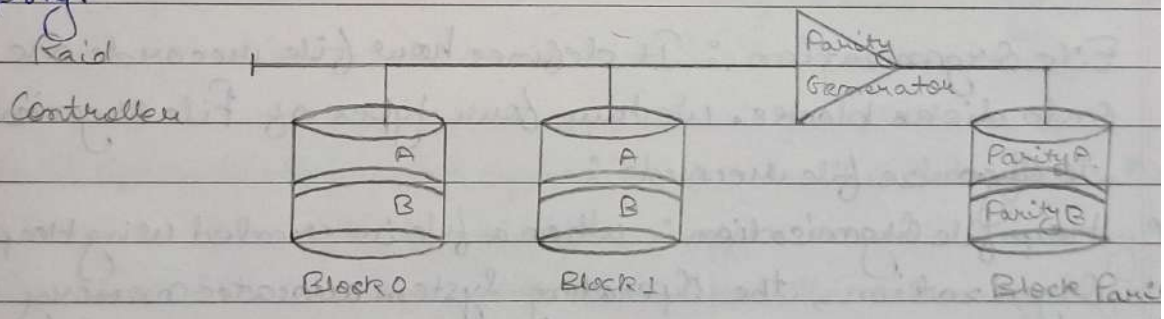
x Raid 2 : Raid 2 records error correction code using Hamming
distance for its data, Striped on different disks. like
each data bit in a word is recorded on a separate disk
and ECC codes of the data words are stored on a different
set disks. Due to its complex structure and high cost, it is
not commercially available.



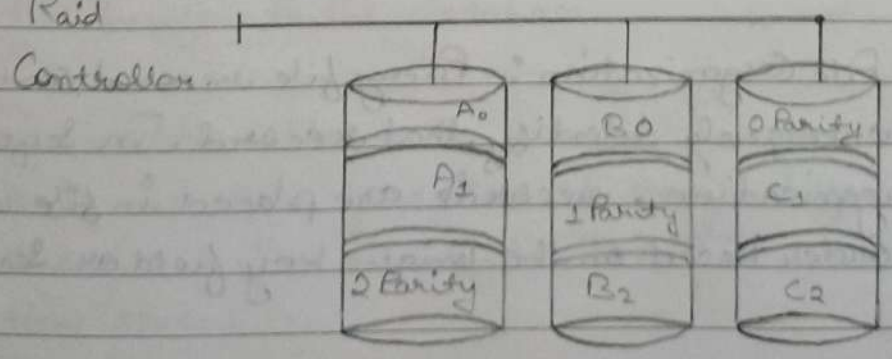
Raid 3 : Raid 3 Stripes the data onto multiple disks. parity bit generated for data word is stored on a differ This technique makes it to overcome single disk



Raid 4 : An entire block of data is written onto the disk the parity generated and stored on a different disk. No level 3 uses byte Striping whereas level 4 uses block le ping.



Raid 5 : Raid 5 writes whole data blocks onto different but the parity bits generated for data block Stripe ar ributed among all the data blocks disks rather than S them on a different dedicated disk.



Raid 6 : It is an extension of level 5. In this level, two independent parities are generated and stored in distributed fashion among multiple disks. Two parities provide additional fault tolerance. This level requires at least four disk drives to implement Raid. 2 parity, 2 disk hai aur 2 parity kyuki agar ek bhi kharab hogya toh dusra se chala jaye

Raid ← Ecc code (Q)
Controller ← X-OR Parity (P)

A_0	B_0	Q_0	P_0
A_1	Q_1	P_1	D_1
Q_2	P_2	C_2	D_2
P_3	B_3	C_3	Q_3

File Structure : Relative data and information is stored collectively in file formats. A file is a sequence of records stored in binary format. A disk drive is formatted into several blocks that can store records. File records are mapped onto those disk blocks.

File Organization : It defines how file records are mapped onto disks blocks. We have four types of File Organisation to organize file records :

Heap file Organisation : When a file is created using Heap file Organization, the Operating System allocates memory area to that file without any further accounting details. File records can be placed anywhere in that memory area. It is the responsibility of the software to manage records. Heap file does not support any ordering, sequencing, or indexing on its own.

Sequential File Organization : Every file record contains a data field to uniquely identify that record. In sequential file organisation, records are placed in file in some sequential order based on the unique key field or search key.

Particaly, it is not possible to store all the records tially in physical form.

iii) Hash file Organisation : Hash file Organisation uses H function Computation on some fields of the records. Output of the hash function determines the location disk block where the records are to be placed.

iv) Clustered file Organization : Clustered file Organisat is not considered good for large databases. In th mechanism, related records from one or more rel are kept in the same disk blocks, that is, the ordering records is not based on primary key or Search key.

→ Transaction : A transaction can be defined as a grou tasks. A single task is the minimum processing unit w cannot be divided further. or It is a set of operations to perform a logical unit of work.

Operations in Transaction :- A transaction gener

1. Read operation — represent change in d
2. Write operation

1. Read Operation : Read operation reads the data from the base and then stores it in the buffer in main memory. Ex : Read(A) instruction will read the value of A from database and will store it in the buffer in the main m

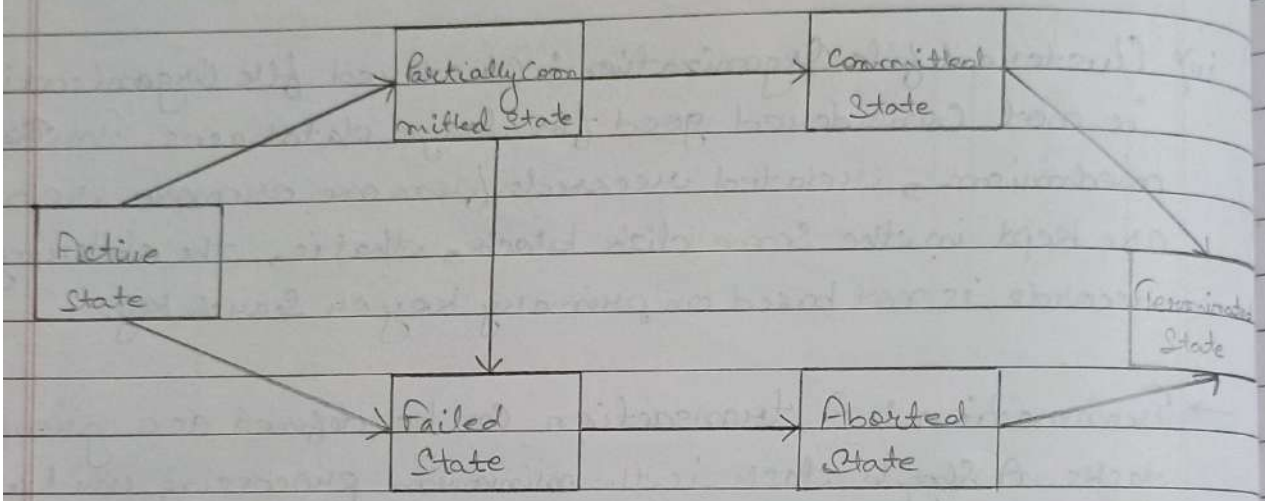
2. Write Operation : Write Operation writes the updated data value back to the database from the buffer. Ex : Write(A) will write the updated updated value of A the buffer to the database.

→ Transaction States :

A transaction goes through many different states thr throughout its life cycle. These states are called tran tion states.

Transaction are as follows:

1. Active State
2. Partially Committed State.
3. Committed State
4. Failed State
5. Aborted State
6. Terminated State.



Transaction States in DBMS

1. Active State :- • This is the first State in the life cycle of a transaction.

A transaction is called in an active state as long as its instruction are getting executed.

All the changes made by transaction now are stored in the buffer in main memory.

2. Partially Committed State :- • After the last instruction of transaction has executed, it enters into a partially committed State.

After entering this state, the transaction is considered to be partially committed.

It is not considered fully committed because all the changes made by the transaction are still stored in the buffer in the ~~main~~ main memory.

3. Committed State :- • After all the changes made by t transaction have been successfully stored in the databa it enters into a Committed State.
• Now, transaction is considered to be fully committed.

4. Failed State :- • When a transaction is getting executed the active state or partially committed state and some fa occurs due to which it becomes impossible to continue t execution, it enters into a failed state.

5. Aborted State :- • After the transaction has failed and ed into a failed state, all the changes made by it have to be undone.
• To undo the changes made by a transaction, it becomes scary to roll back the transaction.
• After the transaction has rolled back completely, it en into an aborted state.

Terminated State

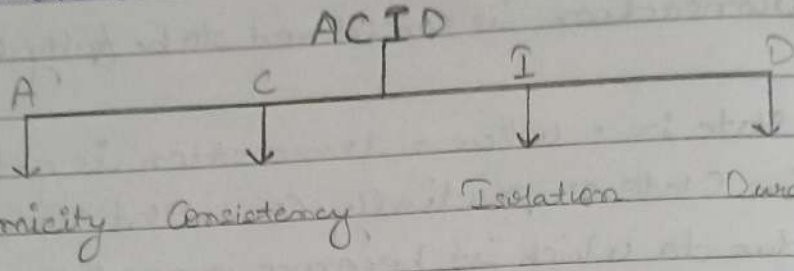
6. Terminated State :- • This is the last state in the life of a transaction.
• After entering the committed state or aborted state, t transaction finally enters into a terminated state where its life cycle finally comes to an end.

→ Commit Command :- is used to permanently save save transaction into the database.

→ Rollback Command :- This command will restores the d ase to last committed State. It is also used with Save Command to jump to a savepoint in an ongoing transa

→ Savepoint Command :- Savepoint Command is used t temporarily save a transaction so that you can r back to that point whenever required.

→ Acid properties in DBMS :- The Acid properties are meant for transaction that goes through a different group of tasks and there we come to see the role of Acid properties.



- Atomicity : We mean that either transaction takes place at once or doesn't happen at all. There is no midway i.e transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or it is not executed at all.
  - Abort : If a transaction aborts, changes made to the database are not visible
  - Commit : If a transaction commits, changes are made visible.

  It is also called 'All or nothing rule'.

Ans :

- Consistency : This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.
  Ex : Total amount : 1400
  Total before T occurs : 500+200 = 700
  Total after T occurs : 400+300 = 700
  Therefore the database is consistent.

- Isolation : This property ensures that multiple transaction can occur concurrently without leading to the inconsistency of the database state. Transaction occurs independently without interference. Changes occuring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in

a state that is equivalent to a ~~ed~~ state achieved these were executed Serially in some order.

- Durability : This property ensures that once the transaction has completed execution, the updates and modification to the database are stored in and written in disk and they persist even if a System failure occurs. These updates now becomes permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

Property	Responsibility for maintaining properties
• Atomicity	Transaction Manager.
• Consistency	Application programmer.
• Isolation	Concurrency Control Manager.
• Durability	Recovery Manager.

→ How to implement Atomicity in Transaction ?

Ans: Shadow Copy :- In shadow Copy Scheme, a transaction that wants to update the database first creates a complete copy of the database. All updates are done on the new database Copy, leaving the Original Copy, the Shadow Copy, untouched. If any point the transaction has to be aborted, the System merely deletes the new Copy. The old Copy of the database has not been affected.

This Scheme is based on making Copies of database, Called Shadow Copies, assumes that only one transaction is active at a time. This Scheme also assumes that the database is Simply a file on disk. A pointer Called dp-pointer is maintained on disks ; it points to the current Copy of the database.

→ Concurrent transaction : When multiple transaction ~~Ex~~ execute Concurrently in an uncontrolled or unrestricted manner, then it might lead to several problems. The Problems are Commonly reffered to as Concurrency problems in a database environment.

→ Types of Concurrency Problem :-

i) Lost Update Problem :- In lost update problem, an update done to a data by a transaction is lost as it is overwritten by the update done by another transaction.

ii) Dirty Read Problem :- Occurs when one transaction updates an item ~~in~~ is changed or reverted back to its last value and fails. But the updated item is used by another transaction before the item is changed or reverted back to its last value.

iii) Unrepeatable Read problem :- The unrepeatable problem occurs when two or more read operations of the same transaction read different values of the same variable.

iv) Incorrect Summary Problem :- Consider a situation, where one transaction is applying the aggregate function on some records while another transaction is updating these records. The aggregate function may calculate some values before the values ~~from~~ have been updated and others after they updated.

Advantages of Concurrency :

i) Waiting time will be decreased.

ii) Response time will decrease

iii) Resource utilization will increase.

iv) System performance & Efficiency is increased.

→ Schedule :- A series of operation from one transaction to another transaction is known as Schedule. It is used to preserve the order of the operation in each of the individual transaction. or it is chronological execution sequence of multiple transaction.

Schedule :- i) Serial Schedule

ii) Non-Serial Schedule →

iii) Serialization Schedule.

iv) Conflict Serializable

vi) View Serializable
vii) Recoverable Schedule
viii) Non-Recoverable Schedule
viii) Cascadless Schedule
ix) Cascading Schedule
x) Strict Schedule.

i) Serial Schedule : Schedules in which the transactions are executed non- interleaved, i.e, a Serial schedule is one in which no transaction starts until a running transaction has ended are called Serial Schedules.

ii) Non-Serial Schedule : This is type of Scheduling where the operations of multiple transaction are interleaved. This might lead to rise in the Concurrency problem. The transactions are executed in non- Serial manner, keeping the end result correct and same as the Serial Schedule. Unlike Serial Schedule where one transaction must wait for another to complete all its operation, in the non- Serial Schedule, the other transaction proceeds without waiting for the previous transaction to complete.

a) Serializable : This is used to maintain the consistency of the database. It is mainly used in the non- Serial Scheduling to verify whether the Scheduling will lead to any Consistency or not. a Serial Schedule does not need the Serializability bec because it follows a transaction only when the previous transaction is Complete.

i) Conflict Serializable : A schedule is called Conflict Serializable if it Can be transformed into a Serial Schedule by Swapping non-conflicting operation. Two operation are said to be conflicting if all Conditions Satisfy:

- They belong to different transaction
- They operate on the same data item
- At least one of them is a write operation

ii) View Serializable : A Schedule is called view Serializable if it is view equal to a serial Schedule. A Conflict Schedule is a view Serializable but if the Serializability contain blind writes, then the view Serializable does not Conflict Serializable.

b) Non-Serializable : It is of two types :- Recoverable & Non recoverable.

i) Recoverable Schedule : Schedules in which transactions commit only after all transactions whose changes they read Commit are called recoverable Schedules. In other words, If some transaction T_j is reading value updated or written by some other transaction T_i, then commit of T_j, must occur after the commit T_i.

It is of 3 types :-

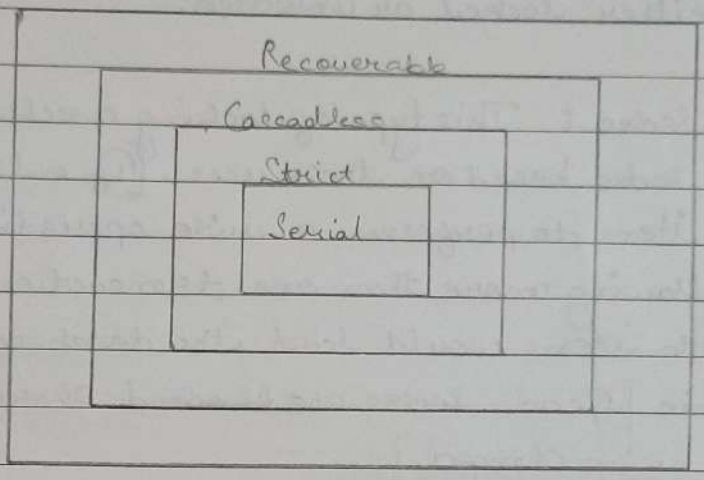
a) Cascading Schedules : Also called avoids cascading abort/rollback. When there is a failure in one transaction and this leads to the rolling back or aborting other dependent transaction, then such Scheduling is reffered to as Cascading rollback.

Cascadeless Schedule : Schedules in which transaction read value only after all transactions whose changes they are going to read Commit are called Cascadeless Schedules. Avoids the at a single transaction abort leads to a series of transaction rollback. A strategy to prevent Cascading abort is to disallow a transaction from reading uncommitted changes from another transaction in the same Schedule.

c) Strict Schedule : A Schedule is strict if for any two transaction T_i, T_j, if a write operation of T_i precedes a Conflicting operation of T_j, then the commit or abort event of T_i also precedes that conflicting operation of T_j. In other words T_j can read or write updated or written value of T_i only after T_i commits / aborts.

iii) Non-Recoverable Schedule : Ex : Consider the following Schedule involving two transaction T_1 and T_2.

T_2 read the value of A written by T_1 and committed. T_1 later aborted, therefore the value read by T_2 is wrong but since T_2 committed, this Schedule is non-recoverable.



→ Conflict Serializable ~~Conflict~~ Schedule :

- A schedule is called Conflict Serializability if after swapping of non-conflicting operations, it can transform into a Serial Schedule.
- The Schedule will be a Conflict Serializable if it is Conflict equivalent to a Serial Schedule.

→ Conflicting Operations :-

The two operations become conflicting if all conditions only :

1. Both belong to Separate transaction.
2. They have the same data item
3. They contain at least one write operation

→ Concurrency Control :- In a multiprogramming environment where multiple transaction can be executed simultaneously, it is highly important to control the concurrency of transaction. We have concurrency control protocols to ensure atomicity, isolation and Serializability of concurrent transaction.

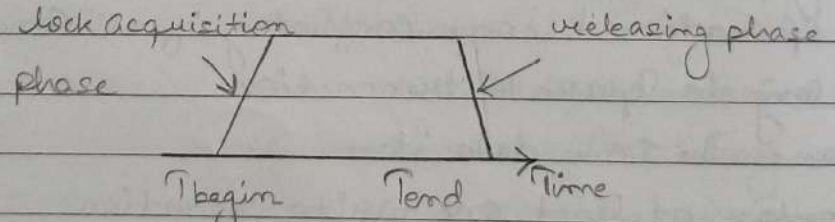
a) Lock based protocols
b) Time stamp based protocols.

a) Lock-based Protocols : Database Systems equipped with lock based protocols use a mechanism by which [any transaction cannot read or write data until it acquires an appropriate lock on it]

i) Binary locks : A lock on a data item can be in two states; it is either locked or unlocked.

ii) Shared locks : This type of locking mechanism differentiates the locks based on their uses. [If a lock is acquired on a data item to perform a write operation, it is an exclusive lock.] Allowing more than one transaction to write on the same data item would lead the databases into an inconsistent state. [Read locks are ~~blocked~~ shared because no data value is being changed.]

→ 2-Phase Locking Protocols : This locking protocol divides the execution phase of a transaction starts executing, it seeks permission for the locks it requires. The second part is where the transaction acquires all the locks. As soon as the transaction releases its first lock, the third phase starts. In this phase the transaction cannot demand any new locks; it only releases the acquired locks.



Two phase locking has two phases, one is ~~growing~~ growing, where all the locks are being acquired by the transaction; and the second phase is shrinking, where the locks held by the transaction are being released.

Growing phase → locks are acquired and no locks are released

Shrinking phase → locks are released and no locks are acquired

b) Time stamp Ordering Protocol :- Time Stamp means it is a unique value assign to every transaction.