IARE '24 DISCORD TEAM

II-II

DBMS MODULE 1 SOLUTIONS

PRIYANANDINI • LALITHA • UJJWAL • IKRAM

CONCEPTUAL MODELING
INTRODUCTION

DBMS MODULE 1

PART-B

1) Compare and Contrast file Systems with database systems.

BASIS	FILE SYSTEM	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	It provides backup and recovery of data even if it is lost.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.

2) Define Data Abstraction and discuss levels of Abstraction.

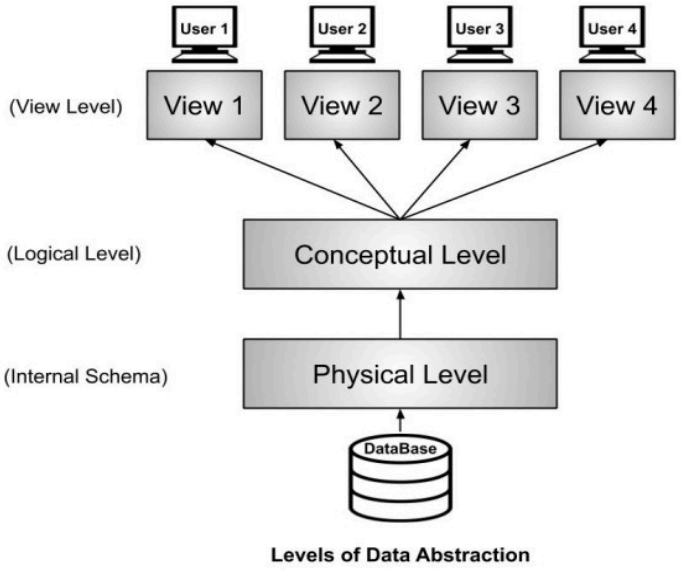
Data Abstraction is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data.

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

Levels of abstraction for DBMS

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

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



Levels of Data Abstraction

Physical or Internal Level

It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database.

For example, customer information is stored in tables, and data is stored in the form of blocks of storage such as bytes, gigabytes, etc.

Logical or Conceptual Level

Logical level is the intermediate level or the next higher level. It describes what data is stored in the database and what relationship exists among those data. It tries to describe the entire or whole data because it describes what tables to be created and what are the links among those tables.

The logical level is used by developers or database administrators (DBA).

View or External Level

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

View level can be used by all users (all levels' users).

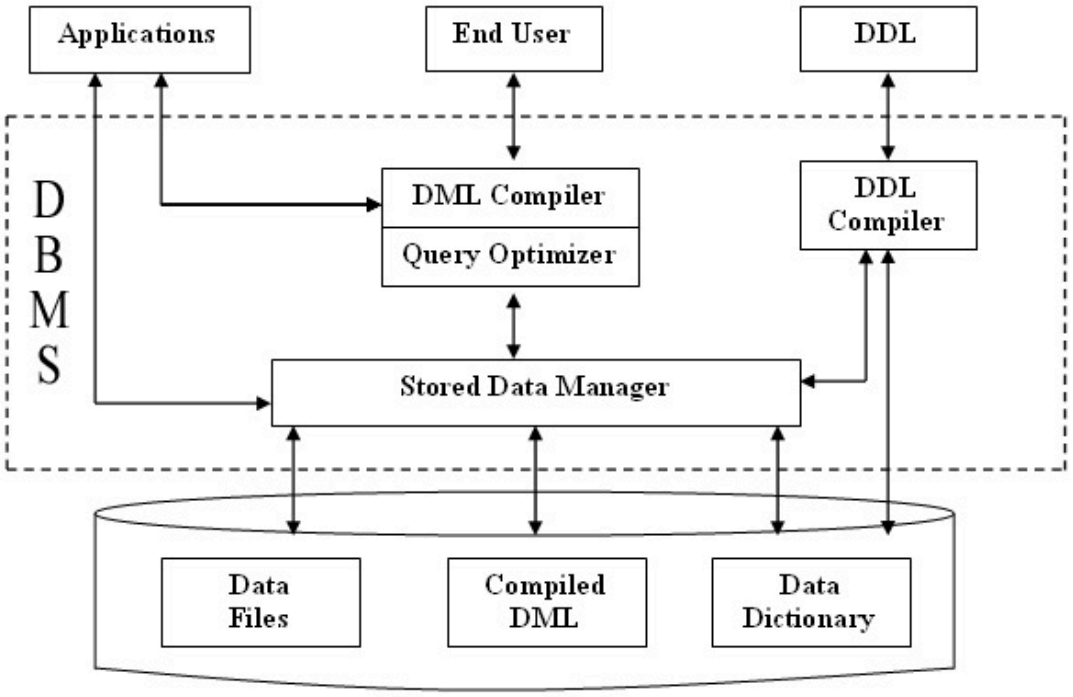
3) Discuss different types of Data models.

- Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and a relationship is an association among these entities. It was widely used in database design. A set of attributes describe the entities. For example, student_name, and student_id describe the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as a 'relationship set'.
- Relational Data Model: This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations.
- Object-based Data Model: An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in the 1980s, various database systems following the

Object-Oriented approach were developed. Here, the objects are nothing but the data-carrying its properties.

- Semi-structured Data Model: The semi-structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.

4) Describe the Structure of DBMS



The database system is divided into three components:

1. Query Processor
2. Storage Manager
3. Disk Storage

Query Processor:

It interprets the requests (queries) received from the end-user via an application program into instructions. It also executes the user request which is received from the DML compiler.

Query Processor contains the following components –

- DML Compiler – It processes the DML statements into low-level instruction (machine language), so that they can be executed.
- DDL Interpreter – It processes the DDL statements into a set of tables containing meta data (data about data).
- Embedded DML Pre-compiler – It processes DML statements embedded in an application program into procedural calls.
- Query Optimizer – It executes the instruction generated by DML Compiler.

Storage Manager:

Storage Manager is a program that provides an interface between the data stored in the database and the queries received. It is also known as Database Control System. It maintains the consistency and integrity of the database by applying the constraints and executing the DCL statements. It is responsible for updating, storing, deleting, and retrieving data in the database.

It contains the following components –

- Authorization Manager – It ensures role-based access control, i.e,. checks whether the particular person is privileged to perform the requested operation or not.
- Integrity Manager – It checks the integrity constraints when the database is modified
- Transaction Manager – It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus, it ensures that the database remains in a consistent state before and after the execution of a transaction.
- File Manager – It manages the file space and the data structure used to represent information in the database.
- Buffer Manager – It is responsible for cache memory and the transfer of data between the secondary storage and main memory.

Disk Storage:

It contains the following components –

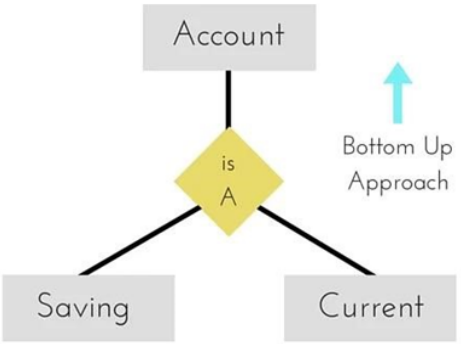
- Data Files – It stores the data.

- Data Dictionary – It contains information about the structure of any database object. It is the repository of information that governs the metadata.
- Indices – It provides faster retrieval of data items.

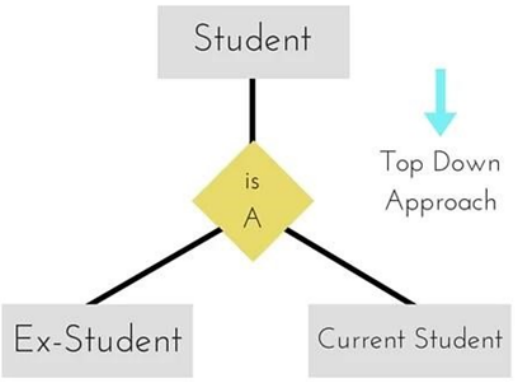
5) Discuss additional features of the ER-Models.

ER model stands for an Entity-Relationship model. ER Models use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.
additional features of the ER-Models:

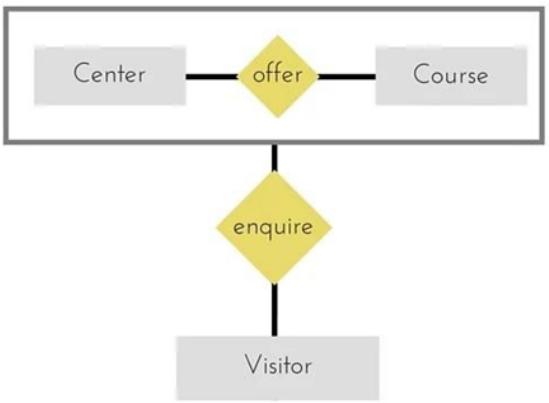
1. Generalisation
2. Specialisation
3. Aggregation



Generalisation is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalisation, the higher level entity can also combine with other lower level entities to make further higher level entities. It's more like a Superclass and Subclass system, but the only difference is the approach, which is bottom-up. Hence, entities are combined to form a more generalised entity, in other words, sub-classes are combined to form a super-class.



Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entities. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.



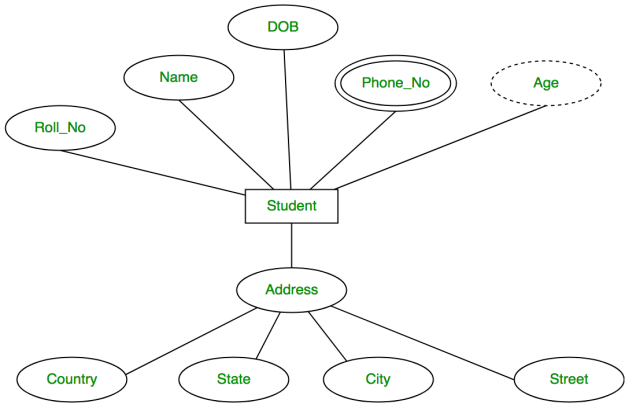
Aggregation is a process when relation between two entities is treated as a single entity. In the diagram above, the relationship between Center and Course together, is acting as an Entity, which is in relationship with another entity Visitor. Now in the real world, if a Visitor or a Student visits a Coaching Center, he/she will never enquire about the center only or just about the course, rather he/she will enquire about both.

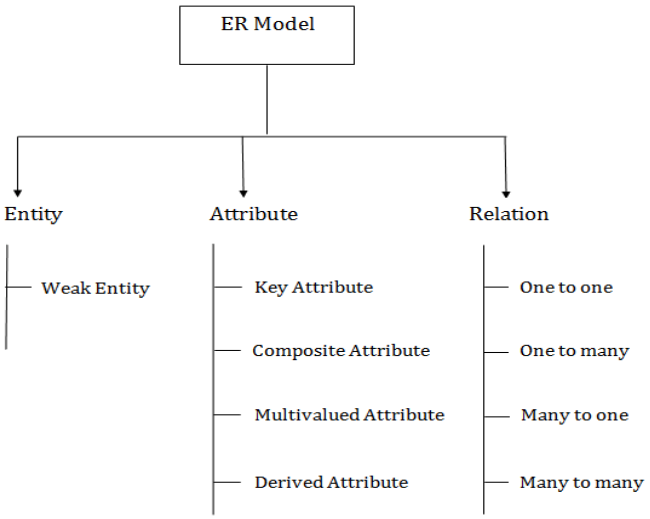
6) Discuss about the Concept Design with the ER Model.

Refer here

ER model-

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.
- For example, Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.





Include part B 15th answer!

7) Explain in detail Different types of Data Independence with examples.

Refer here

Data Independence

- Data independence can be explained using the three-schema architecture.
- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level

There are two types of data independence:

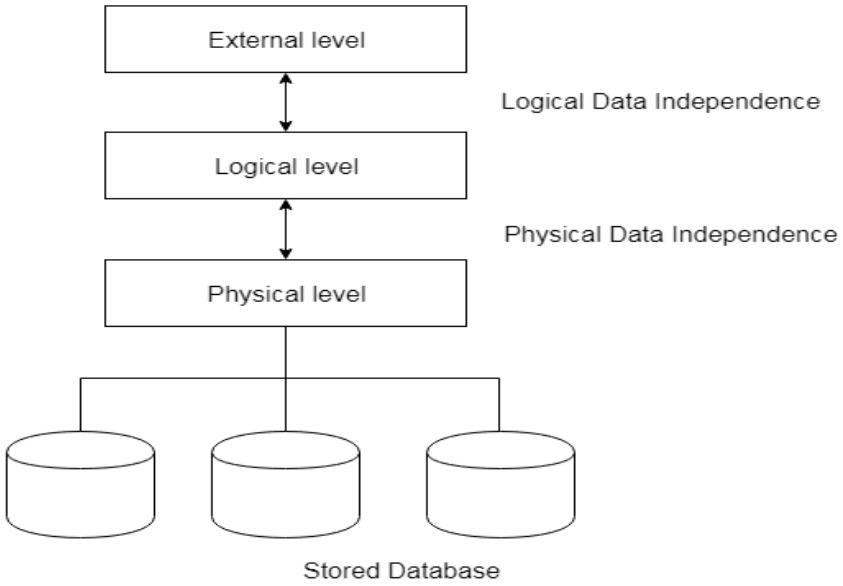
1. Logical Data Independence

- Logical data independence refers to the characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.

- Physical data independence occurs at the logical interface level.



8) Explain different types of database users and write the functions of DBA.

There are four different types of database system users. Different types of user interfaces have been designed for the different types of users.

- Naive users:Naive users are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously. For example, a clerk in the university who needs to add a new instructor to Users is differentiated by the way they expect to interact with the system department A invokes a program called New - hire. This program asks the clerk for the name of the new instructor, her new ID, the name of the department (that is, A), and the salary
- Application programmers: Application programmers are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces. Rapid application development (RAD) tools are tools that enable an application

programmer to construct forms and reports with minimal programming effort.

- Sophisticated users: Sophisticated users interact with the system without writing programs. In- stead, they form their requests either using a database query language or by using tools such as data analysis software. Analysts who submit queries to explore data in the database fall in this category
- Specialized users: Specialized users are sophisticated users who write specialized database applications that do not fit into the traditional data processing framework.

Functions of DBA:

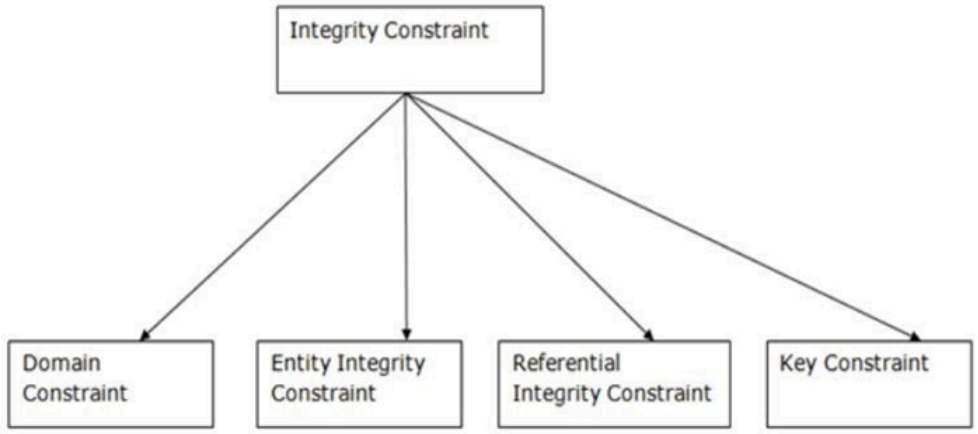
One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

- The DBA creates the original database schema by executing a set of data definition statements in the DDL.
- Storage structure and access-method definition.
- Schema and physical-organization modification.
- Routine maintenance.
- Periodically backing up the database.
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required. – Monitoring jobs running on the Database.

9) List out different types of integrity constraints.

- Integrity constraints are a set of rules. It is used to maintain the quality of information.

- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to guard against accidental damage to the database.



1. Domain constraints:Domain constraints can be defined as the definition of a valid set of values for an attribute.
2. Entity integrity constraints:The entity integrity constraint states that primary key value can't be null.
3. Referential Integrity Constraints:A referential integrity constraint is specified between two tables.
4. Key constraints: Keys are the entity set that is used to identify an entity within its entity set uniquely.

10) Discuss about Different keys used in database design with examples.

KEYS in DBMS is an attribute or set of attributes which helps you to identify a row(tuple) in a relation(table). They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table.

Key is also helpful for finding unique records or rows from the table. Database key is also helpful for finding unique records or rows from the table.

Types of Keys in DBMS (Database Management System):

1. Primary Key 2. Candidate Key 3. Foreign Key 4.Super Key

1) Primary Key:

PRIMARY KEY in DBMS is a column or group of columns in a table that uniquely identifies every row in that table. The Primary Key can't be a duplicate meaning the same value can't appear more than once in the table. A table cannot have more than one primary key. Example:

In the following example, StudID is a Primary Key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Candidate Key:

CANDIDATE KEY in SQL is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes. The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key. A table can have multiple candidate keys but only a single primary key.

Candidate key Example:

In the given table Stud ID, Roll No, and email are candidate keys that help us to uniquely identify the student record in the table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com

3	13	Dana	Natan	mno@yahoo.co m

Foreign key:

FOREIGN KEY is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.It acts as a cross-reference between two tables as it.

Example:

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	Fname	Lname
B002	David	Warner
B017	Sara	Joseph
B009	Mike	Brunton

In this key DBMS example, we have two tables, a teacher and department in a school. However, there is no way to see which search works in which department.

In this table, adding the foreign key in Deptcode to the Teacher name, we can create a relationship between the two tables.

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner

B017	002	Sara	Joseph
B009	001	Mike	Brunton

Super key:

A superkey is a group of single or multiple keys which identifies rows in a table.

A Super key may have additional attributes that are not needed for unique identification. Example:

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

In the above-given example, EmpSSN and EmpNo names are superkeys.

11) Distinguish between a strong entity set and a weak entity set?

Strong Entity Set	Weak Entity Set
it has its own primary key.	It does not save sufficient attributes to form a primary Key on its own.
It is represented by a rectangle.	It is represented by a double rectangle.
It contains a primary key represented by an underline.	It contains a Partial Key or discriminator represented by a dashed underline.
The member of strong entity set is called as dominant entity set.	The member of weak entity set is called as subordinate entity set.
The Primary Key is one of its attributes which uniquely Identifies its member.	The Primary Key of weak entity set is a combination of partial Key and Primary Key of the strong entity set.
The relationship between two strong entity set is represent by a diamond symbol.	The relationship between one strong and a weak entity set is represented by a double diamond sign. It is known as identifying relationship.
The line connecting strong entity set with the relationship is single	The line connecting weak entity set with the identifying relationship is double.
Total participation in the relationship may or may not exist.	Total participation in the identifying relationship always exists.

12) Differentiate relation schema and relational instance?

1. A schema is the design representation of a database whereas an instance is the snapshot of a database at a particular moment.
2. Instance changes very frequently, whenever data is removed or added to the database. As against, the changes in schema occur rarely.
3. For example, schema and instance can be easily perceived by analogy to a program. At the time of writing a program in a programming language, the variables of that program are declared at first, this is analogous to the schema definition. Additionally, each variable in a program must have some values associated at a particular time; this is similar to an instance.

Definition of instance: The data stored in a database at a particular moment of time is called an instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.

Definition of schema: Design of a database is called the schema. Schema is of three types:

- Physical schema
- logical schema
- view schema.

—----------doubt—----------

13) List and explain the design issues of entity relationship ER Design Issues.

Normally, users often mislead the concept of the elements and the design process of the ER diagram. Thus, it leads to a complex structure of the ER diagram and certain issues that do not meet the characteristics of the real-world enterprise model. The basic design issues of an ER database schema are:

1. Use of Entity Set vs Attributes :

   The use of an entity set or attribute depends on the structure of the real-world enterprise that is being modeled and the semantics associated with its attributes. It leads to a mistake when the user uses

the primary key of an entity set as an attribute of another entity set. Instead, he should use the relationship to do so. Also, the primary key attributes are implicit in the relationship set, but we designate it in the relationship sets.

2. Use of Entity Set vs. Relationship Sets:

   It is difficult to examine if an object can be best expressed by an entity set or relationship set. To understand and determine the right use, the user needs to designate a relationship set for describing an action that occurs in-between the entities. If there is a requirement of representing the object as a relationship set, then it's better not to mix it with the entity set.

3. Use of Binary vs n-ary Relationship Sets:

   Generally, the relationships described in the databases are binary relationships. However, non-binary relationships can be represented by several binary relationships.

   For example, we can create and represent a ternary relationship 'parent' that may relate to a child, his father, as well as his mother. Such relationships can also be represented by two binary relationships i.e, mother and father, that may relate to their child.

   Thus, it is possible to represent a non-binary relationship by a set of distinct binary relationships.

4. Placing Relationship Attributes:

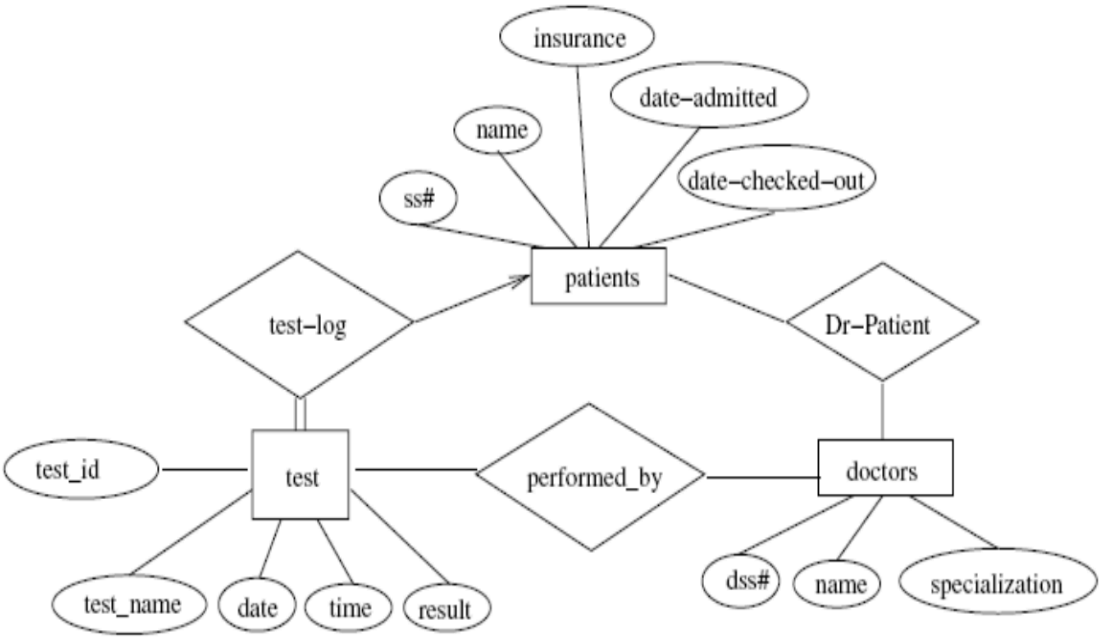
   The cardinality ratios can become an effective measure in the placement of the relationship attributes. So, it is better to associate the attributes of one-to-one or one-to-many relationship sets with any participating entity sets, instead of any relationship set. The decision of placing the specified attribute as a relationship or entity attribute should possess the characteristics of the real world enterprise that is being modeled.

   For example, if there is an entity which can be determined by the combination of participating entity sets, instead of determining it as a

separate entity. Such type of attribute must be associated with the many-to-many relationship sets.

Thus, it requires the overall knowledge of each part that is involved in designing and modeling an ER diagram. The basic requirement is to analyze the real-world enterprise and the connectivity of one entity or attribute with another.

14) Construct an ER-Diagram for a hospital with a set of patients and a set of medical doctors. Associated with each patient a log of the various tests and examinations conducted.



15) Describe about Basic Concepts of ER Model in DBMS.

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

Entity:

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attributes sharing similar values.

For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

Attributes:

Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.

There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

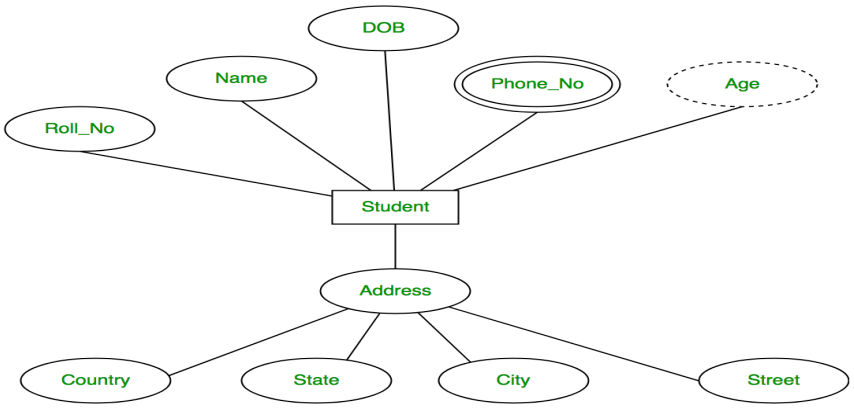
Keys:

Key is an attribute or collection of attributes that uniquely identifies an entity among an entity set.

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

- Super Key – A set of attributes (one or more) that collectively identifies an entity in an entity set.
- Candidate Key – A minimal super key is called a candidate key. An entity set may have more than one candidate key.

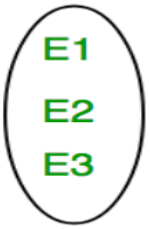
- Primary Key – A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.
- Relationship:
  The association among entities is called a relationship. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and Enrolls are called relationships.
- Cardinalities:
  Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of another set via the relationship set.



16) Explain ER Model, with its Entity and Entity Set?

ER Model is used to model the logical view of the system from data perspective which consists of these components:
An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.
An Entity is an object of Entity Type and the set of all entities is called an entity set. e.g.; E1 is an entity having an Entity Type Student and the set of all students is called Entity Set. In ER diagram, Entity Type is represented as:

Entity Type



Entity Set

17)Discuss about ER Model and its Relationships?

ER model

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.
- Relationship:
  The association among entities is called a relationship. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and Enrolls are called relationships.
- Include diagram from 6Q

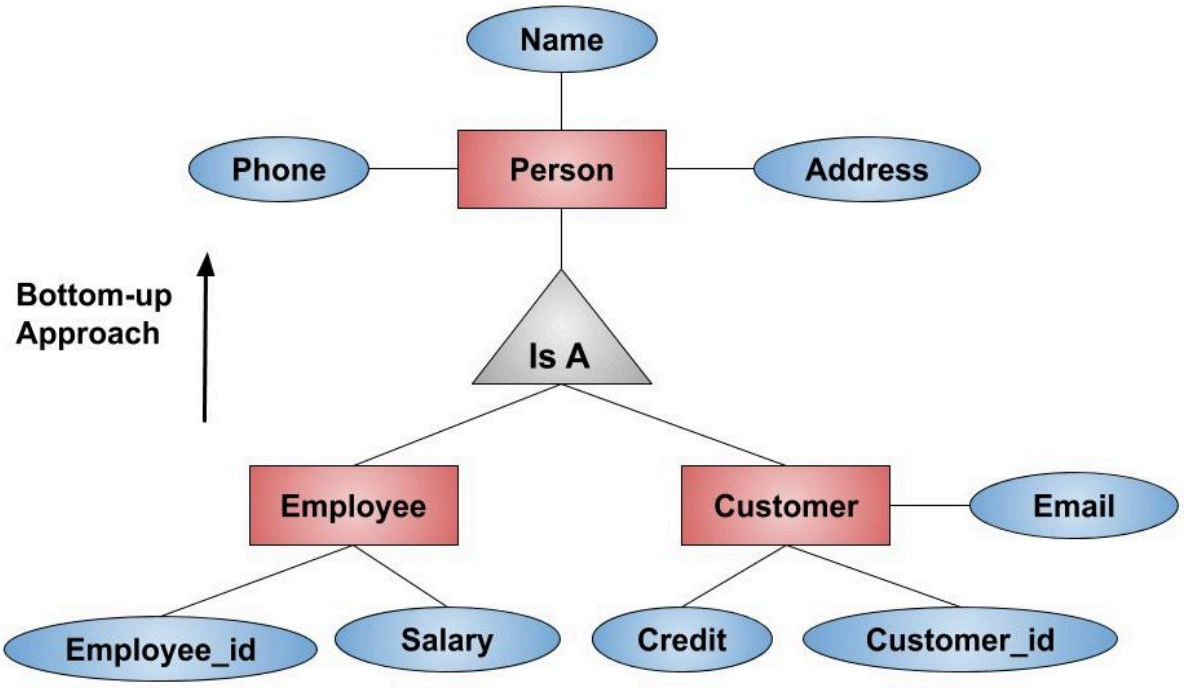
18)Discuss about generalization with a neat diagram?

Generalization is a bottom-up approach in which multiple lower-level entities are combined to form a single higher-level entity. Generalization is usually used

to find common attributes among entities to form a generalized entity. It can also be thought of as the opposite of specialization.

The following enhanced entity relationship diagram expresses entities in a hierarchical database to demonstrate generalization:

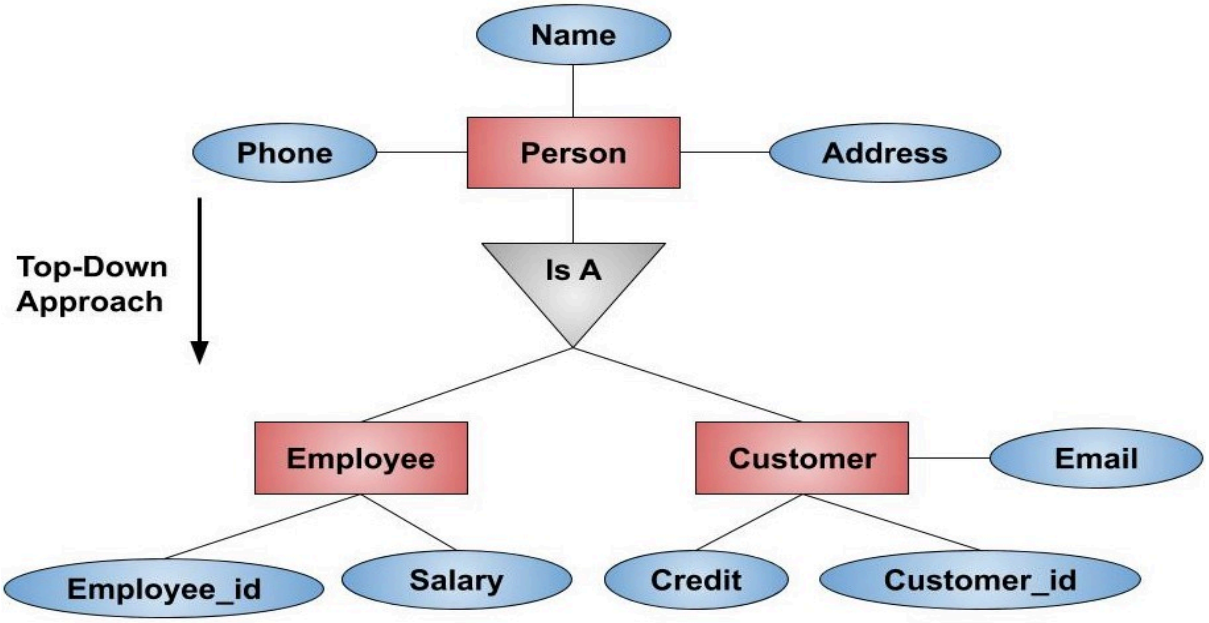
Explain in your own words



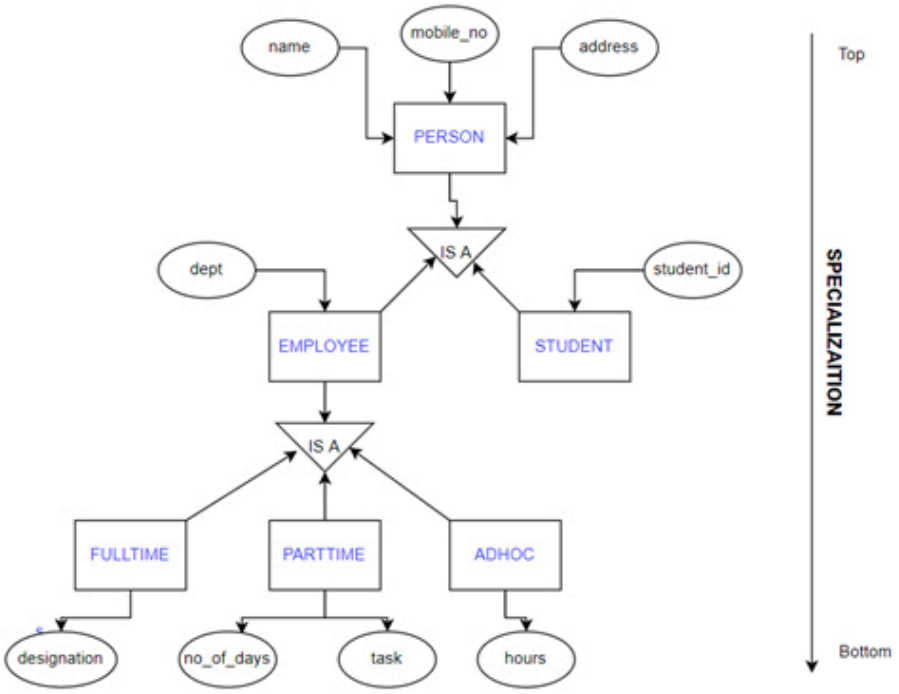
Generalization

19)Explain specialization with a neat diagram?

Specialization is a top-down approach in which a higher-level entity is divided into multiple specialized lower-level entities. In addition to sharing the attributes of the higher-level entity, these lower-level entities have specific attributes of their own. Specialization is usually used to find subsets of an entity that has a few different or additional attributes.
The following enhanced entity relationship diagram expresses the entities in a hierarchical database to demonstrate specialization:



Specialization

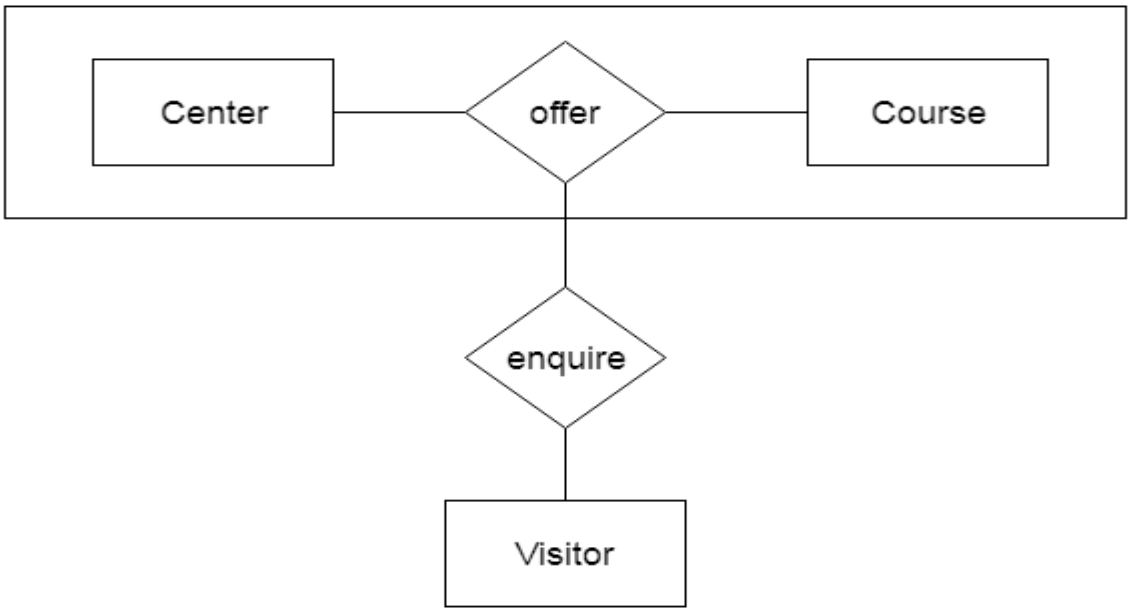


20)Describe aggregation with a neat diagram?

For 18,19,20 refer PART-B question no 5

Aggregation refers to the process by which entities are combined to form a single meaningful entity. The specific entities are combined because they do not make sense on their own. To establish a single entity, aggregation creates a relationship that combines these entities. The resulting entity makes sense because it enables the system to function well.

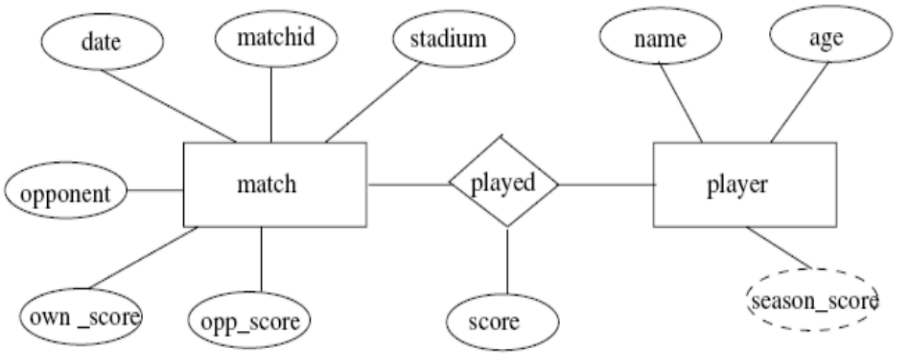
For example: Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.



PART-A

1 Construct an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match,

the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes.



2) Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to- many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. Calculate the minimum number of tables required to represent this situation in the relational model.

We require 3 tables.

3) Analyze and find whether modifications made at conceptual level makes application programs written by users at view level to be modified in a database. Analyze your answer with illustration.

- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.

- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.

Not complete!

4) We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Analyze why, then, do we have weak entity sets?

We want to avoid the data duplication and consequent possible inconsistencies caused by duplicating the key of the strong entity.

- Weak entities reflect the logical structure of an entity being dependent on
  another entity.
- Weak entities can be deleted automatically when their strong entity is deleted.
- Weak entities can be stored physically with their strong entities.

5) What are the responsibilities of a DBA? If we assume that the DBA is never interested in running his or her own queries; does the DBA still need to understand query optimization? Why?

The functions of a DBA include:

• Schema definition. The DBA creates the original database schema by executing a set of data
definition statements in the DDL.

• Storage structure and access-method definition.

• Schema and physical-organization modification.

1 Routine maintenance.

2 Periodically backing up the database.

3 Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.

– Monitoring jobs running on the Database.

6) Describe the structure of a DBMS. If your operating system is upgraded to support some new functions on OS files (e.g., the ability to force some sequence of bytes to disk), which layer(s) of the DBMS would you have to rewrite to take advantage of these new functions.

7) Why did relational models become more popular compared with other record based models?

In the early 1980s microcomputer-based implementations of database management systems, overwhelmingly based on the relational model, became available. Since that time, the relational model has gained wide acceptance and has continued to be the subject of research studies to extend its capabilities.

8) Describe the process to convert the ER model into relational schema.

ER diagrams can be mapped to relational schema, that is, it is possible to create relational schema using ER diagram. We cannot import all the ER constraints into a relational model, but an approximate schema can be generated. There are several processes and algorithms available to convert ER Diagrams into Relational Schema.
They are:

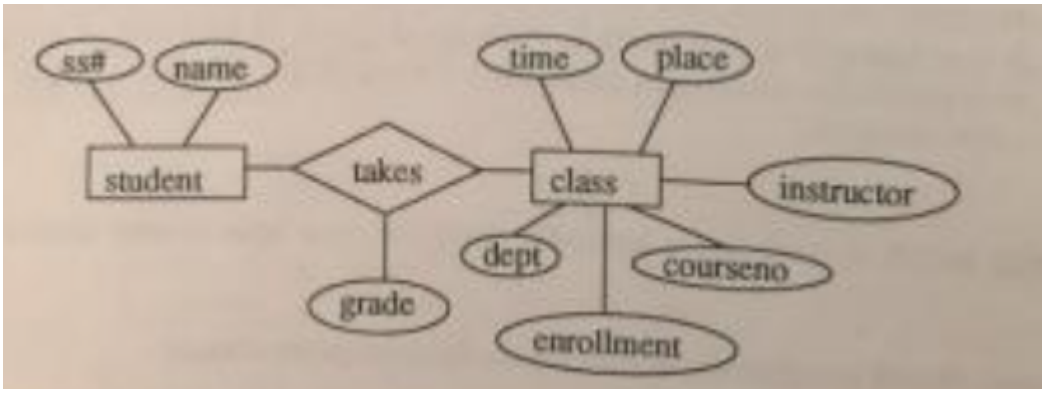
1. Mapping Entity
2. Mapping Relationship
3. Mapping Weak Entity Sets
4. Mapping Hierarchical Entities

{ ___DOUBT___ }

9) Discuss the disadvantages of the file processing system, and explain how these disadvantages are avoided in DBMS?

REFER PART B Q1

10) Design a relational database for a university registrar's office. The office maintains data about each class, including the instructor, the number of students enrolled, and time and place of the class Meetings. For each student - class pair, a grade is recorded.



PART C

1)List the advantages of DBMS.

Advantage of Database Management System (DBMS):

- Better Data Transferring
- Better Data Security
- Better data integration
- Minimized Data Inconsistency
- Faster data Access

- Better decision making
- Increased end-user productivity
- Simple.

2) List the database Applications.

- Accounting: payments, receipts, account balance, assets.
- Human Resources: employee records, salaries, tax deductions
- Manufacturing: production, inventory, orders, supply chain
- Online Retails: order tracking, customized recommendations
- Banking and Finance: all transactions
- Credit card Transaction: generation of monthly statements.
- Finance: storing information about holdings and sales, 3. Universities: registration, grades
- Airlines: reservations, schedules
- Telecommunications: keeping records of calls made, generating monthly bills.

3) Define instances and schemas of the database.

Definition of instance: The data stored in a database at a particular moment of time is called an instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.
Definition of schema: Design of a database is called the schema. Schema is of three types:

- Physical schema
- logical schema
- view schema.

4) Discuss Data Independence.

Data Independence

- Data independence can be explained using the three-schema architecture.
- Data independence refers to the characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

There are two types of data independence:

1. Logical Data Independence
   - Logical data independence refers to the characteristic of being able to change the conceptual schema without having to change the external schema.
2. Physical Data Independence
   - Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.

5)How application programs access databases?

6) Define (i) Database (ii) DBMS.

• Database: Database is a collection of interrelated data.

• DBMS: database is a collection of interrelated data and a set of programs can access that system.

7) List out main components of Database storage structure?

There are four main components on which the working of a DBMS depends. This includes:

- Data: The main component is the data. The entire database is set based on the data and the information processed based on it. This data acts as

a bridge between the software and hardware components of DBMS. This can further be divided into three varieties:

- User Data – The actual data based on which the work is done
- Metadata – This is the data of the data, i.e., managing the data required to enter the information
- Application MetaData – This is the structure and format of the queries

8) What are the main responsibilities of the Transaction management component?

Transactions are a set of operations used to perform a logical set of work. A transaction usually means that the data in the database has changed. One of the major uses of DBMS is to protect the user's data from system failures.
It is used to solve Read/Write Conflict. It is used to implement Recoverability, Serializability, and Cascading. Transaction Management is also used for Concurrency Control Protocols and Locking of data. Transactions can be implemented using SQL queries and Server.

9) Outline main functions of Query Processor.

Query Processing is the activity performed in extracting data from the database. In query processing, it takes various steps for fetching the data from the database. The steps involved are:

1. Parsing and translation
2. Optimization
3. Evaluation

10)Define (i) Entity (ii) Attribute

Entity:An entity is an object that exists and is distinguishable from other objects . Example : specific person , company , event , plant.
Attributes:Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, age as attributes. There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

11)Define Relationship and Relationship set.

Relationship:The association among entities is called a relationship. For example, employee entities have relation work.

Relationship Set: Relationships of similar type are called relationship sets. Like entities, a relationship too can have attributes. These attributes are called descriptive attributes. Degree of relationship The number of participating entities in an relationship defines the degree of the relationship

- Binary = degree 2
- Ternary = degree 3
- n-ary = degree

12)Discuss about Data Definition language.

- It is used to define database structure or pattern.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

- Create: It is used to create objects in the database.
- Alter: It is used to alter the structure of the database.

- Drop: It is used to delete objects from the database.
- Truncate: It is used to remove all records from a table.
- Rename: It is used to rename an object.
- Comment: It is used to comment on the data dictionary.

13)Discuss about Data Manipulation language.

It is used for accessing and manipulating data in a database. It handles user requests.

- Select: It is used to retrieve data from a database.
- Insert: It is used to insert data into a table.
- Update: It is used to update existing data within a table.
- Delete: It is used to delete all records from a table.
- Merge: It performs UPSERT operation, i.e., insert or update operations.
- Call: It is used to call a structured query language or a Java subprogram.

14) List responsibilities of a DBA.

The functions of a DBA include:

•The DBA creates the original database schema by executing a set of data definition statements in the DDL.

• Storage structure and access-method definition.

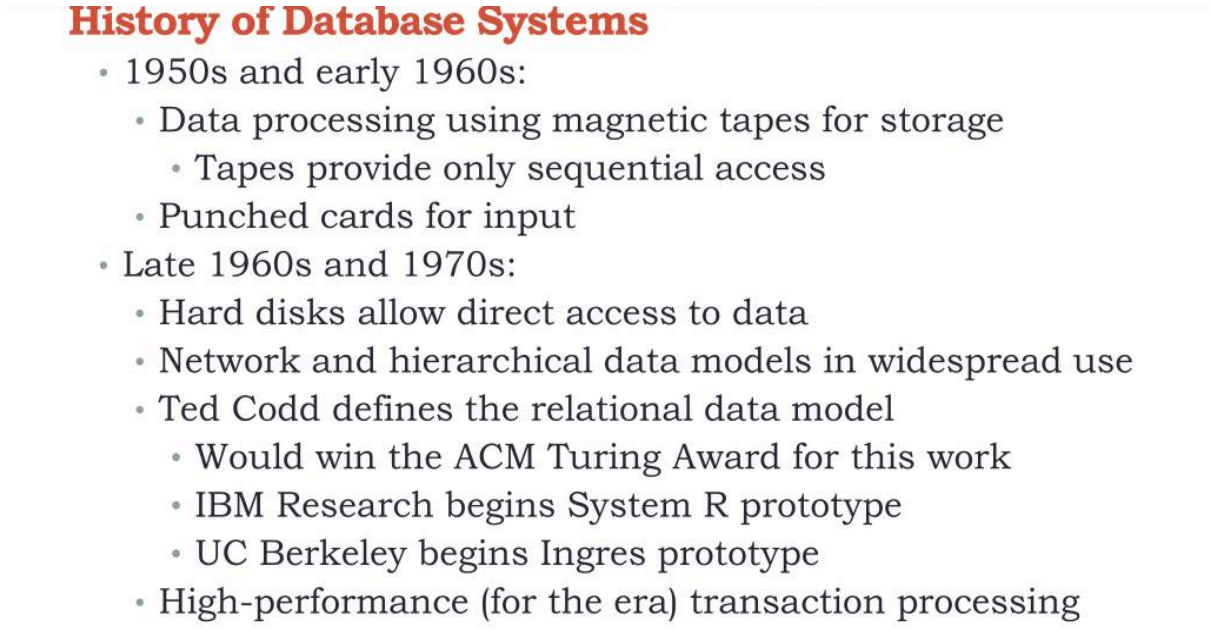
• Schema and physical-organization modification.

1. Routine maintenance.

2.Periodically backing up the database.

3.Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required. – Monitoring jobs running on the Database.

15) Outline the History of Database Systems.



16) Discuss how you can change the data in the table.

The UPDATE statement changes existing data in one or more rows in a table.

17)List various types of attributes.

In ER diagram, attributes associated with an entity set may be of the following types-

1. Simple attributes
2. Composite attributes
3. Single valued attributes
4. Multi valued attributes
5. Derived attributes
6. Key attributes

18) Discuss How you can alter and destroy tables?

The SQL ALTER TABLE command is used to add, delete or modify columns in an existing table. You should also use the ALTER TABLE command to add and drop various constraints on an existing table.

Syntax

The basic syntax of an ALTER TABLE command to add a New Column in an existing table is as follows.

```
ALTER TABLE table_name ADD column_name datatype;
```

DROP TABLE:

The DROP TABLE command deletes a table in the database.
The following SQL deletes the table "Shippers":

Example:

```
DROP TABLE Shippers;
```

19) Define a data model? List the types of data model used.

REFER PART B Q 3

20)List the levels of data abstraction.

REFER PART B Q 2

IARE '24 DISCORD TEAM

II-II

DBMS MODULE 2 SOLUTIONS

VISHAL • UJJWAL • IKRAM

RELATIONAL APPROACH

DBMS MODULE 2

Part - A

1Q) For the following relational database, give the expressions in RA (Relational Algebra). Student (stuno, stuname, major, level, age), Class(Class name, meets at room, fid), faculty(fid, fname, deptID).

1. **Find the names of all juniors (level = JR) who are enrolled in a class taught by I.teach.**
2. **Find the age of the oldest student who is either a history major or is enrolled in a course taught by I.Teach.**
3. **Find the names of all classes that either meet in room R128 or have five or more students enrolled**

1.
```
SELECT DISTINCT S.Sname
FROM   Student S, Class C, Enrolled E, Faculty F
WHERE  S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND
       F.fname = 'I.Teach' AND S.level = 'JR'
```

2.
```
SELECT MAX(S.age)
FROM   Student S
WHERE  (S.major = 'History')
       OR S.snum IN (SELECT E.snum
                     FROM   Class C, Enrolled E, Faculty F
                     WHERE  E.cname = C.name AND C.fid = F.fid
                            AND F.fname = 'I.Teach' )
```

3.
```
SELECT   C.name
FROM     Class C
WHERE    C.room = 'R128'
         OR C.name IN (SELECT   E.cname
                       FROM     Enrolled E
                       GROUP BY E.cname
                       HAVING   COUNT (*) >= 5)
```

2Q) Illustrate the relational employee (name, salary, deptno), department (deptno, deptname, address). Solve which query cannot be expressed using the basic relational algebra operations.

The sum of all employees' salaries

The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product (x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations

3Q) Discuss Query in relational algebra to find second highest salary of employee from employee relation.

Consider below simple table:

Name	Salary
abc	100000
bcd	1000000
efg	40000
ghi	500000

How to find the employee whose salary is second highest. For example, in above table, "ghi" has the second highest salary as 500000.

Simple query:

```
select * from employee where salary=(select Max(salary) from
employee);
```

We can nest the above query to find the second largest salary.

```
select * from employee
group by salary
order by  salary desc limit 1,1;
```

Note that instead of nesting for second, third, etc largest salary, we can find nth salary using general query like in MySQL:

```
SELECT salary
FROM employee
ORDER BY salary desc limit n-1,1
```

4Q) Consider the following schema given. The primary keys are underlined. Sailors (Sailor-ID, sailor-name, sailor-rating, sailor-age) Boats (boat-ID, boat-name, boat color), Reserves (Sailor ID, boat-ID, day) Write queries in relational algebra.

1. **Find the names of sailors who have reserved boat number 120**
2. **Find the names of sailors who have reserved a green boat**
3. **Find the names of sailors who have not reserved a green boat.**
4. **Find the names of sailors with the highest rating**

Sailors(sid: integer, sname: string, rating: integer, age: real);
Boats(bid: integer, bname: string, color: string);
Reserves(sid: integer, bid: integer, day: date).

```
SQL> select * from sailors;

       SID SNAME                          RATING        AGE
---------- ------------------------------ ---------- ----------
        22 Dustin                              7         45
        29 Brutus                              1         33
        31 Lubber                              8       55.5
        32 Andy                                8       25.5
        58 Rusty                              10         35
        64 Horataio                            7         35
        71 Zorba                              10         16
        74 Horataio                            9         35
        85 Art                                 3       25.5
        95 Bob                                 3       63.5
10 rows selected.
```

```
SQL> select * from reserves;
       SID        BID DAY
---------- ---------- ---------
        22        101 10-OCT-98
        22        102 10-OCT-98
        22        103 08-OCT-98
        22        104 07-OCT-98
        31        102 10-NOV-98
        31        103 06-NOV-98
        31        104 12-NOV-98
        64        101 05-SEP-98
        64        102 08-SEP-98
        74        103 08-SEP-98
10 rows selected.
```

```
SQL> select * from boats;

       BID BNAME                COLOR
---------- -------------------- --------------------
       101 Interlake            blue
       102 Interlake            red
       103 Clipper              green
       104 Marine               red
```

If boat Number is 103.Then find the name of sailors?

```
select s.sname
from sailors s,reserves r
where s.sid=r.sid and r.bid=103;

Output:

SNAME
--------------------
Dustin
Lubber
Horataio
```

What is the color of boat reverse by Lubber?

```
select b.color
from boats b,sailors s,reserves r
where s.sid=r.sid and b.bid=r.bid and s.sname='Lubber';

COLOR
--------------------
red
green
red
```

Find the sids of sailors with age over 20 who have not reserved a red boat.

```
select s.sid,s.sname
from sailors s,boats b,reserves r
where s.sid=r.sid and b.bid=r.bid and s.age>20 and b.color!='red';

       SID SNAME
---------- --------------------
        22 Dustin
        22 Dustin
        31 Lubber
        64 Horataio
        74 orataio
```

5Q) Consider the following database. Employee(employee-name, street, city), works (employee-name, company-name, salary), company (company-name, city), Manager (Employee-name, manager-name).

1. **Given an expression in the relational algebra, the tuple relational calculus, for the following query.**
2. **Find the names of all employees who work for the estate bank.**

1. Consider the following relational database:

employee(e-name, street, city)
works(e-name, c-name, salary)
company(c-name, city)
manages(e-name, m-name)

For each of the following queries, give an expression in

i) the relational algebra,
ii) the tuple relational calculus,
iii) the domain relational calculus.

For example, the following expressions would be used to find the names of all employees who work for the First Bank Corporation:

i) $\Pi_{e\text{-}name}(\sigma_{c\text{-}name = \text{'First Bank Corporation'}}(works))$

ii) $\{t \mid \exists\ s \in works\ (t[e\text{-}name] = s[e\text{-}name] \wedge s[c\text{-}name] = \text{"First Bank Corporation"})\}$

iii) $\{<p> \mid \exists\ c, s\ (<p, c, s> \in works \wedge c = \text{"First Bank Corporation"})\}$

a) Find the names and cities of residence of all employees who work for the First Bank Corporation.

i) $\Pi_{e\text{-}name,\ city}(employee \bowtie (\sigma_{c\text{-}name = \text{'First Bank Corporation'}}(works)))$

ii) $\{t \mid \exists\ r \in employee\ \exists\ s \in works\ (t[e\text{-}name] = r[e\text{-}name] \wedge t[city] = r[city] \wedge r[e\text{-}name] = s[e\text{-}name] \wedge s[c\text{-}name] = \text{"First Bank Corporation"})\}$

6Q) Define the RA expressions for the following queries. Sailor Schema (Sailor ID, sailor name, rating, age), Reserves (Sailor ID, boat ID, day), boat schema (boat ID, boat name, color).

1. **Find the names of sailors who have reserved boat name 103.**
2. **Find the sailor ID of sailors who have reserved a red boat.**
3. **Find the colors of boats reserved by the sailor rubber.**
4. **Find the names of sailors who have reserved a red boat.**

(Q1) Find the names of sailors who have reserved boat 103.

This query can be written as follows:

$$\pi_{sname}((\sigma_{bid=103}Reserves) \bowtie Sailors)$$

We first compute the set of tuples in Reserves with $bid = 103$ and then take the natural join of this set with Sailors. This expression can be evaluated on instances of Reserves and Sailors. Evaluated on the instances $R2$ and $S3$, it yields a relation that contains just one field, called *sname*, and three tuples $\langle Dustin \rangle$, $\langle Horatio \rangle$, and $\langle Lubber \rangle$. (Observe that there are two sailors called Horatio, and only one of them has reserved a red boat.)

(Q2) Find the names of sailors who have reserved a red boat.

$$\pi_{sname}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$$

This query involves a series of two joins. First we choose (tuples describing) red boats. Then we join this set with Reserves (natural join, with equality specified on the *bid* column) to identify reservations of red boats. Next we join the resulting intermediate relation with Sailors (natural join, with equality specified on the *sid* column) to retrieve the names of sailors who have made reservations of red boats. Finally, we project the sailors' names. The answer, when evaluated on the instances $B1$, $R2$ and $S3$, contains the names Dustin, Horatio, and Lubber.

An equivalent expression is:

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves) \bowtie Sailors)$$

(Q3) Find the colors of boats reserved by Lubber.

$$\pi_{color}((\sigma_{sname='Lubber'}Sailors) \bowtie Reserves \bowtie Boats)$$

This query is very similar to the query we used to compute sailors who reserved red boats. On instances $B1$, $R2$, and $S3$, the query will return the colors gren and red.

7Q) Observe the following relational database, give the expressions in RA. Student (stuno, sstuname, major, level, age) class (classname, meets at room, fid), faculty(fid, fname, deptID)

1. **Find the names of all juniors (level = JR) who are enrolled in a class taught by I.Teach.**
2. **Find the age of the oldest student who is either a history major or is enrolled in a course taught by I>Tech.**

Student

snum	sname	major	level	age
101	Helen	CS	JR	19
102	Charles	CS	SR	21
103	Andy	CS	GR	25
104	Bob	CS	SR	23
105	Zorba	CS	GR	31

Enrolled

snum	cname
101	CSC343
101	CSC443
101	ECE300
102	CSC343
102	ECE300
103	CSC343
103	CSC443
103	ECE300
103	ECE201
105	CSC343

Class

name	meets_at	room	fid
CSC343	W1	BA1180	201
CSC443	T2	BA1170	202
ECE300	M1	BA1180	203
ECE201	F12	BA1160	203
CSC165	R3	BA1170	202

Faculty

fid	fname	deptid
201	S. Jackson	301
202	M. Shanks	301
203	I. Teach	302

Q6:

```
SELECT DISTINCT S.sname as Student_Name
FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.snum = E.snum AND E.cname = C.name
        AND C.fid = F.fid
        AND F.fname = 'I.Teach' AND S.level = 'JR'
```

Answer

Student_Name
Helen

8Q) Sailor schema (Sailor ID, sailor name, rating, Age), reserves(sailor-ID, boat ID, day) boat schema(boat ID, boatname, color)

1. **Find the age of the youngest sailor for each rating level.**
2. **Find the age of the youngest sailor who is eligible to vote for each rating level with at least two such sailors?**
3. **Find the no. of reservations for each red boat?**

4. **Find the average age of sailors for each rating level that is at least 2 sailors.**

9Q) How the statement "the SID's of suppliers who supply some red or green parts" can be represented in the form of relational algebra and tuple relational algebra and tuple relational calculus from the above relations.

Suppliers scheme:

Suppliers (SID: INTEGER,
Sname: STRING,
Address: STRING)
Parts (PID: INTEGER,
Pname : STRING
Color: STRING)
Catalog (SID: INTEGER,
PID: INTEGER,
Cost: REAL)

Catalog

SID	PID	Cost
1	1	$10.00
1	2	$20.00
1	3	$30.00
1	4	$40.00
1	5	$50.00
2	1	$9.00
2	3	$34.00
2	5	$48.00

Parts

PID	Pname	Color
1	Red1	Red
2	Red2	Red
3	Green1	Green
4	Blue1	Blue
5	Red3	Red

SID	Sname	Address
1	Yosemite Sham	Devil's canyon, AZ
2	Wiley E. Coyote	RR Asylum, NV
3	Elmer Fudd	Carrot Patch, MN

2. Find the *sid*s of suppliers who supply some red or green part.

$$\pi_{sid}(\pi_{pid}(\sigma_{color='red' \vee color='green'} Parts) \bowtie catalog)$$

10Q) Given the relations Employee(name, salary, dept no), department(deptno, deptname, address) Solve which query cannot be expressed using the basic SQL operations.

***employee (name, salary, deptno)* and**
department (deptno, deptname, address)

Which of the following queries cannot be expressed using the basic relational algebra operations (U, -, x, π, σ, p)? (GATE CS 2000)

(a) Department address of every employee
(b) Employees whose name is the same as their department name
(c) The sum of all employees' salaries
(d) All employees of a given department

Answer: (c)

Explanation:
The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product (x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations.

Part - B

1Q) Illustrate different set operations in Relational algebra with an example.

The set operation are mainly categorized into the following:

1. Union operation
2. Intersection operation
3. Set difference or Minus operation

UNION Operation:

Notation:
A ∪ S
where, A and S are the relations,

symbol '∪' is used to denote the Union operator.

The result of Union operation, which is denoted by A ∪ S, is a relation that basically includes all the tuples that are present in A or in S, or in both, eliminating the duplicate tuples.

INTERSECTION Operation:

Notations:
A ∩ S
where, A and S are the relations,

symbol '∩' is used to denote the Intersection operator.

The result of Intersection operation, which is denoted by A ∩ S, is a relation that basically includes all the tuples that are present in both A and S.

MINUS (or SET DIFFERENCE) Operation:

Notations:
A - S

where, A and S are the relations,

symbol ' – ' is used to denote the Minus operator.

The result of Intersection operation, which is denoted by A – S, is a relation that basically includes all the tuples that are present in A but not in S.

Consider a relation Student(FIRST, LAST) and Faculty(FIRSTN, LASTN) given below :

First	Last
Aisha	Arora
Bikash	Dutta
Makku	Singh
Raju	Chopra

FirstN	LastN
Raj	Kumar
Honey	Chand
Makku	Singh
Karan	Rao

1. Student UNION Faculty :

Student ∪ Faculty

First	Last
Aisha	Arora
Bikash	Dutta
Makku	Singh
Raju	Chopra
Raj	Kumar
Honey	Chand
Karan	Rao

2. Student INTERSECTION Faculty :

Student ∩ Faculty

First	Last
Makku	Singh

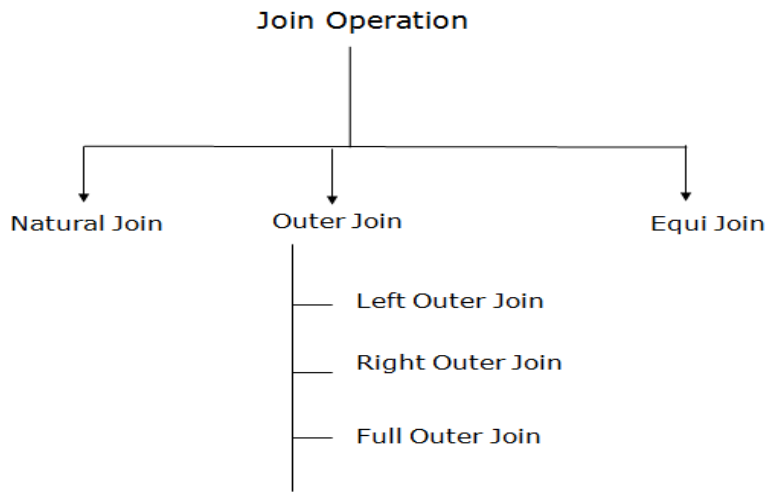
3. Student MINUS Faculty :

Student - Faculty

First	Last
Aisha	Arora
Bikash	Dutta
Raju	Chopra
Raj	Kumar
Honey	Chand
Karan	Rao

2Q) Define Join. Explain different types of joins in relational algebra.

In DBMS, a join statement is mainly used to combine two tables based on a specified common field between them. In terms of Relational algebra, it is the cartesian product of two tables followed by the selection operation. Thus, we can execute the product and selection process on two tables using a single join statement.



Refer all operations from DBMS Join Operation - javatpoint

3Q) Discuss Tuple Relational calculus in detail.

Tuple Relational Calculus is a non-procedural query language unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do it. In Tuple Calculus, a query is expressed as

`{t| P(t)}`

where `t = resulting tuples`,

`P(t) = known as Predicate` and these are the conditions that are used to fetch t .

Thus, it generates a set of all tuples t, such that Predicate P(t) is true for t.

P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).

It also uses quantifiers:

∃ t ∈ r (Q(t)) = ”there exists” a tuple in t in relation r such that predicate Q(t) is true.

∀ t ∈ r (Q(t)) = Q(t) is true “for all” tuples in relation r.

For example:

```
{ T.name | Author(T) AND T.article = 'database' }
```

OUTPUT: This query selects the tuples from the AUTHOR relation. It returns a tuple with 'name' from the Author who has written an article on 'database'.

TRC (tuple relational calculus) can be quantified. In TRC, we can use Existential (∃) and Universal Quantifiers (∀).

4Q) Define the difference between Relational Algebra and Relational Calculus

Relational Algebra	Relational Calculus
It is a Procedural Language.	While Relational Calculus is Declarative Language
Relational Algebra means how to	While Relational Calculus means what

obtain the results.	results we have to obtain
The order is specified in which the operations have to be performed.	The order is not specified
Relational Algebra is independent of the domain	Can be domain dependent.
This is a language in which queries can be expressed but the queries should also be expressed in relational calculus to be relationally complete	Database language to be relationally complete., the query written in ikt must be expressed in relational calculus

5Q) Describe Extended relational operations with examples.

Extended operators are those operators which can be derived from basic operators. There are mainly three types of extended operators in Relational Algebra:

- Join
- Intersection
- Divide

Let us consider two tables named as A and B:

A –

RollNo	Name	Marks
1	Aashi	98
3	Anjali	79
4	Brijesh	88

B –

RollNo	Name	Marks
1	Aashi	98
2	Abhishek	87
3	Anjali	79
4	Brijesh	88

- Intersection

  Intersection works on the relation as 'this and that'. In relational algebra, A ∩ B returns a relation instance that contains every tuple that occurs in relation to instance A and relation instance B (both together). Here, A and B need to be union-compatible, and the schema of both result and A must be identical.

Syntax:

```
SELECT * FROM A INTERSECT SELECT * FROM B;
```

RollNo	Name	Marks
1	Aashi	98
3	Anjali	79
4	Brijesh	88

- Divide

  The divide operator is used for the queries that contain the keyword ALL. For e.g. – Find all the students who have chosen additional subjects Machine Learning and Data Mining.

  Student –

Student Name	Subject
Ashish	Machine Learning
Ashish	Data Mining
Shivam	Network Security
Shivam	Data Mining
Tarun	Network Security
Tarun	Machine Learning
Yash	Machine Learning
Yash	Data Mining

Subject –

Student Name
Machine Learning
Data Mining

Output: Student ÷ Subject

Student
Ashish
Yash

Read join from here

6Q) Discuss procedural language in SQL

PL/SQL (Procedural Language/Structured Query Language) is a block-structured language extension of SQL that consists of statements and

language elements that can be used to implement procedural logic in SQL statements which enable developers to combine the power of SQL with Procedural statements. All the statements of a block are passed to the Oracle engine all at once which increases processing speed and decreases the traffic. SQL PL provides statements for declaring variables and condition handlers, assigning values to variables, and implementing procedural language.

Features of PL/SQL:

- PL/SQL is basically a procedural language, which provides the functionality of decision making, iteration and many more features of procedural programming languages.
- PL/SQL can execute a number of queries in one block using single command.
- One can create a PL/SQL unit such as procedures, functions, packages, triggers, and types, which are stored in the database for reuse by applications.
- PL/SQL provides a feature to handle the exception which occurs in PL/SQL block known as exception handling block.
- Applications written in PL/SQL are portable to computer hardware or operating system where Oracle is operational.
- PL/SQL Offers extensive error checking.

7Q) Discuss the structure of the query in TRC with an example.

Tuple Relational Calculus is a non-procedural query language unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do.

In Tuple Calculus, a query is expressed as : `{t| P(t)}`

where t = resulting tuples,

P(t) = known as Predicate and these are the conditions that are used to fetch t

Thus, it generates a set of all tuples t, such that Predicate P(t) is true for t.
P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).
It also uses quantifiers:
∃ t ∈ r (Q(t)) = ”there exists” a tuple in t in relation r such that predicate Q(t) is true.
∀ t ∈ r (Q(t)) = Q(t) is true “for all” tuples in relation r.
Example:
Find the loan number, branch, amount of loans of greater than or equal to 10000 amount.
Ans: {t| t ∈ loan ∧ t[amount]>=10000}

8Q) Illustrate a query in TRC to find the names of sailors who have reserved both a red and green boat? Write a query in TRC to find the names of sailors who have reserved all boats?

9Q) Write a query in TRC to find the names of sailors who have reserved both a red and green boat? Write a query in TRC to find the names of sailors who have not reserved a red boat?

The following is a TRC query for finding the name of sailors who have reserved both red and green boats.

10Q) r Write a TRC query to find the names of sailors who have reserved boat 103?

This query can be written as follows:

$$\pi_{sname}((\sigma_{bid=103} Reserves) \bowtie Sailors)$$

TRC query:

$$\{P \mid \exists S \in Sailors\ \exists R \in Reserves(R.sid = S.sid \wedge R.bid = 103 \wedge P.sname = S.sname)\}$$

This query can be read as follows: "Retrieve all sailor tuples for which there exists a tuple in Reserves, having the same value in the *sid* field, and with $bid = 103$." That is, for each sailor tuple, we look for a tuple in Reserves that shows that this sailor has reserved boat 103. The answer tuple P contains just one field, *sname*.

11Q) Let R = (ABC) and S = (DEF) let r(R) and s(S) both relations on schema R and S. Give an expression in the tuple relational calculus that is equivalent to each of the following.ρ B=19(r) A,F,(ρC=D(r×s)) r ⋂ s

12Q) Sketch the following schema instruction (ID, name, dept name), teaches (ID, course ID, sec ID, semester, year), section (course ID, sec ID, semester, year), student (ID, name, dept name), takes (ID, course ID, sec ID, semester, year, grade). Write the following query in RA, TRC and DRC. Find the names of the instructors not teaching any course.

13Q)r Find the names of the sailors who have reserved a green boat.

```
SELECT S.sname
FROM Sailors S, Boats B, Reserves R
WHERE B.color='green' AND B.bid=R.bid AND S.sid = R.sid
```

14Q) r Describe the sides of sailors who have reserved red and green boats. Find sides of all sailors who have reserved a red boat but not a green boat.

```
SELECT S1.sname FROM Sailors S1, Reserves R1, Boats B1 WHERE
S1.sid=R1.sid AND R1.bid=B1.bid AND B1.color=`red'
INTERSECT
SELECT S2.sname FROM Sailors S2, Reserves R2, Boats B2 WHERE
S2.sid=R2.sid AND R2.bid=B2.bid AND B2.color=`green'

SELECT S1.sid FROM Sailors S1, Reserves R1, Boats B1 WHERE
S1.sid=R1.sid AND R1.bid=B1.bid AND B1.color=`red'
EXCEPT
SELECT S2.sid FROM Sailors S2, Reserves R2, Boats B2 WHERE
S2.sid=R2.sid AND R2.bid=B2.bid AND B2.color=`green'
```

15Q)Describe sides of all sailors who have a rating of 10 or reserved boat 104 find a sailor whose rating is better than every sailor called Horatio.

```
(SELECT sid FROM s WHERE rating>=10) UNION (SELECT sid FROM r
WHERE bid=104)

SELECT sname FROM s WHERE rating > all ( SELECT rating FROM s s2
WHERE s2.sname=`Horatio')
```

16Q) Find the sailors with the highest rating. Find the names of all branches in the loan relation.

17Q) Define set operations with syntax and examples.

ans)

The SQL Set operation is used to combine the two or more SQL SELECT statements.

Types of Set Operation

1. Union
2. UnionAll
3. Intersect
4. Minus

1. Union

- The SQL Union operation is used to combine the result of two or more SQL SELECT queries.
- In the union operation, all the number of data type and columns must be same in both the tables on which the UNION operation is being applied.
- The union operation eliminates the duplicate rows from its resultset.

Syntax:

```
SELECT column_name FROM table1
UNION
SELECT column_name FROM table2;
```

2. Union All

Union All operation is equal to the Union operation. It returns the set without removing duplication and sorting the data.

Syntax:

```
SELECT column_name FROM table1
UNION ALL
SELECT column_name FROM table2;
```

3. Intersect

- It is used to combine two SELECT statements. The Intersect operation returns the common rows from both the SELECT statements.

- In the Intersect operation, the number of data type and columns must be the same.
- It has no duplicates and it arranges the data in ascending order by default.

Syntax:

```
SELECT column_name FROM table1
INTERSECT
SELECT column_name FROM table2;
```

4. Minus

- It combines the result of two SELECT statements. Minus operator is used to display the rows which are present in the first query but absent in the second query.
- It has no duplicates and data arranged in ascending order by default.

Syntax:

```
SELECT column_name FROM table1
MINUS
SELECT column_name FROM table2;
```

For examples please refer to :
DBMS SQL Set Operation - javatpoint

18Q) Write about Division operation in relational algebra with an example.

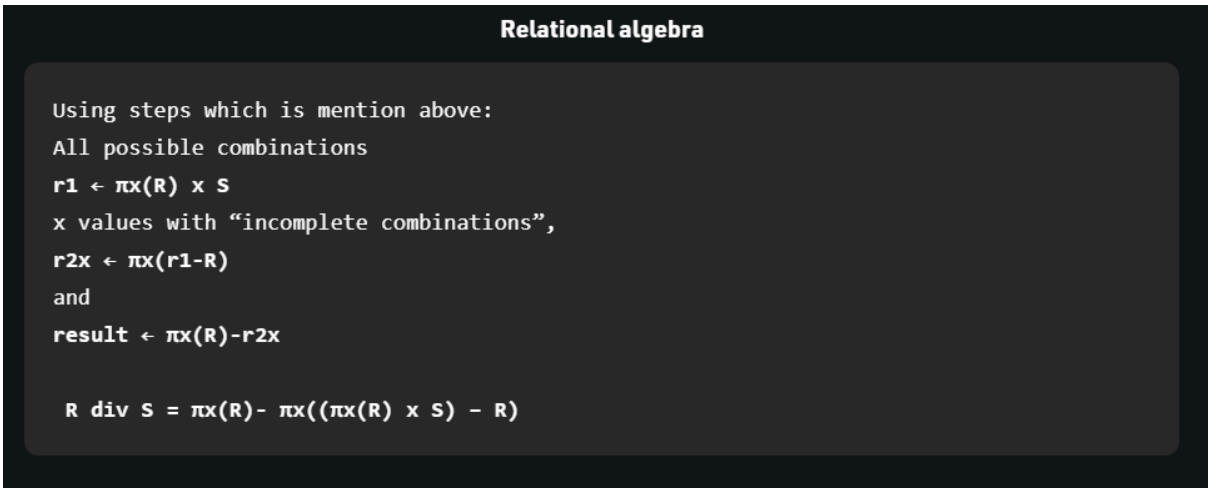
Division is typically required when you want to find out entities that are interacting with all entities of a set of different type entities.

The division operator is used when we have to evaluate queries which contain the keyword ‘all’.

Some instances where division operator is used are:

- Which person has an account in all the banks of a particular city?
- Which students have taken all the courses required to graduate?

In all these queries, the description after the keyword ‘all’ defines a set which contains some elements and the final result contains those units who satisfy these requirements.



Another way you can identify the usage of division operator is by using the logical implication of if...then. In context of the above two examples, we can see that the queries mean that,

1. If there is a bank in that particular city, that person must have an account in that bank.
2. If there is a course in the list of courses required to be graduated, that person must have taken that course.

19Q) Write about join operations with syntax and examples

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

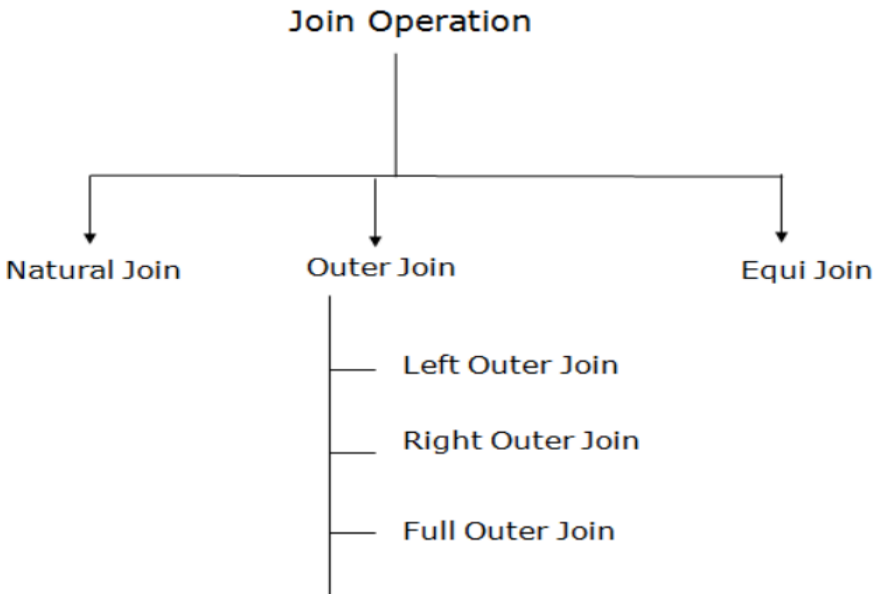
INNER JOIN

LEFT JOIN

RIGHT JOIN

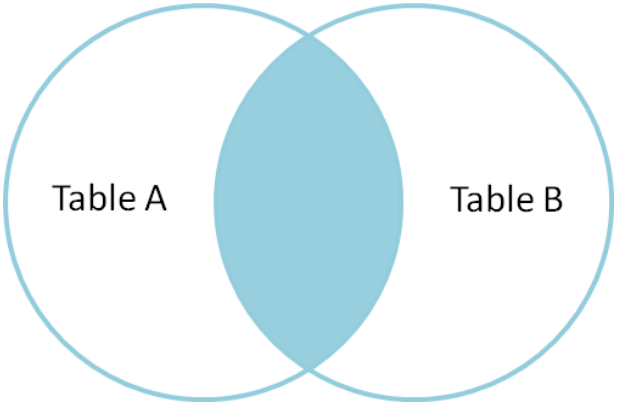
FULL JOIN

A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied. It is denoted by ⋈.



A. INNER JOIN

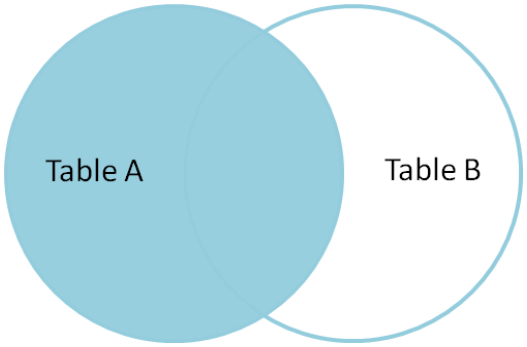
The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



B. 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. LEFT JOIN is also known as LEFT OUTER JOIN.



SQL | Join (Inner, Left, Right and Full Joins) - GeeksforGeeks

Refer to this: DBMS SQL Joins - javatpoint

20Q) Differentiate natural join and inner join operations with examples.

1. Natural Join :

Natural Join joins two tables based on the same attribute name and datatypes. The resulting table will contain all the attributes of both the tables but keep only one copy of each common column.

2. Inner Join :

Inner Join joins two tables on the basis of the column which is explicitly specified in the ON clause. The resulting table will contain all the attributes from both the tables including the common column also.

SR.NO.	NATURAL JOIN	INNER JOIN
1.	Natural Join joins two tables based on same attribute name and datatypes.	Inner Join joins two table on the basis of the column which is explicitly specified in the ON clause.
2.	In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column	In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also
3.	In Natural Join, If there is no condition specifies then it returns the rows based on the common column	In Inner Join, only those records will return which exists in both the tables
4.	SYNTAX: SELECT * FROM table1 NATURAL JOIN table2;	SYNTAX: SELECT * FROM table1 INNER JOIN table2 ON table1.Column_Name = table2.Column_Name;

Part - C

1Q) Describe a create statement with an example.

2Q) Define DML statements in SQL. Give an example.

It is used for accessing and manipulating data in a database. It handles user requests.

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Delete: It is used to delete all records from a table.

Merge: It performs UPSERT operation, i.e., insert or update operations.

Call: It is used to call a structured query language or a Java subprogram.

3Q) Discuss various Aggregate functions used in SQL.

An aggregate function in SQL returns one value after calculating multiple values of a column. We often use aggregate functions with the GROUP BY and HAVING clauses of the SELECT statement.

Various types of SQL aggregate functions are:
Count()
Sum()
Avg()
Min()
Max()

4Q) Define the primary key.

5Q) State the syntax of foreign key constraint.

6Q) What are the data types in SQL?

7Q) Write a SQL statement to find employees whose commission is greater than their salaries.

8Q) Write a SQL statement to find the employees who are not clerks, analysts or salesmen.

9Q) Write a SQL statement to display the names of all the employees and positions where the string AR occurs in the name.

10Q) List out all classes in the select statement.

11Q) Define redundancy and its problems.

12Q) Define functional dependency. Why are some functional dependencies trivial?

13Q) Discuss normalization.

14Q) Differentiate between trivial and non-trivial dependencies.

15Q) If relation R consists of only simple candidate keys then R should be in which normal form?

16Q) Define the First Normal form

17Q) Define the Second Normal form

18Q) Define the Third Normal form

19Q) Define the Fourth Normal form

20Q) Identify the normal forms of the relation R. R(ABCD) FD: {A → B, B → C}

IARE '24 DISCORD TEAM

II-II

DBMS MODULE 3 SOLUTIONS

KAVYA SREE • UJJWAL

SQL QUERY - BASICS , RDBMS - NORMALIZATION

Many Pending

DBMS MODULE 3

PART A

1 Write the SQL expression for the following Queries. Sailor Schema (sailor id, Sailorname, Rating.Age) Reserves (Sailor id, Boat id, Day) Boat Schema (Boat id, Boatname.color) 1. Find the names of sailors who have reserved boat name 103;

SELECT S.sname

FROM Sailors S, Reserves R

WHERE S.sid = R.sid AND R.bid = 103

2. Find the sailor id of sailors who have reserved a red boat;

SELECT S.sid,

FROM Sailors S, Reserves R, Boats B

WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’

3. Find the colors of boats reserved by the sailor rubber?

4. Find the names of sailors who have reserved a red boat?

SELECT S.sname,

FROM Sailors S, Reserves R, Boats B

WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’

2 For the following relational database, give the expressions in SQL. student(stuno, stuname, major,level,age) Class(Classname, meets at, Room, fid) Faculty(fid,fname,deptid) 1. Find the names of all uniors (level = JR) Who are enrolled in a class taught by I.Teach? 2. Find the age of the oldest student who is either a history major or is enrolled in a course taught by I.Tech? 3. Find the names of all classes that either meet in room R128 or have five or more students enrolled?

1.
```
SELECT DISTINCT S.Sname
FROM   Student S, Class C, Enrolled E, Faculty F
WHERE  S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND
       F.fname = 'I.Teach' AND S.level = 'JR'
```

2.
```
SELECT MAX(S.age)
FROM   Student S
WHERE  (S.major = 'History')
       OR S.snum IN (SELECT E.snum
                     FROM   Class C, Enrolled E, Faculty F
                     WHERE  E.cname = C.name AND C.fid = F.fid
                            AND F.fname = 'I.Teach' )
```

3.
```
SELECT   C.name
FROM     Class C
WHERE    C.room = 'R128'
         OR C.name IN (SELECT   E.cname
                       FROM     Enrolled E
                       GROUP BY E.cname
                       HAVING   COUNT (*) >= 5)
```

3 Write the SQL expressions for the following relational database. sailor schema (sailor id, Boat id, sailorname, rating, age) Recerves (Sailor id, Boat id, Day) Boat Schema (boat id, Boatname, color)

1. Find the age of the youngest sailor for each rating level?

SELECT S.sname, S.age

FROM Sailors S

WHERE S.age <= ALL (SELECT age FROM Sailors)

2. Find the age of the youngest sailor who is eligible to vote for each rating level with at lead two such sailors? 3. Find the No.of reservations for each red boat?

For all sailors queries refer Tutorial_6.doc (toronto.edu)

4 Consider the following schema: Suppliers(sid: integer, sname: string, address: string) Parts(pid: integer, pname: string, color: string) Catalog(sid: integer, pid: integer, cost: real) The Catalog relation lists the prices charged for parts by Suppliers. Answer the following questions: Give an example of an updatable view involving one relation. 1. Give an example of an updatable view involving two relations. 2. Give an example of an insertable-into view that is updatable. 3. Give an example of an insertable-into view that is not updatable.

5 Consider following relations in DB and solve the queries: Student (GR NO, name, gender, address, city, class) Marks (GR NO, sub1, sub2, sub3, total, per) 1. Display the student of 'FYBCA' and 'TYBCA'. (2 mark each) 2. Display the marks of student whose gr no> 100. 3. Count the no of girls in FYBCA. 4. count the no: of first class students in TYBCA

6. Consider a relation scheme R = (A, B, C, D, E, H) on which the following functional dependencies hold: A→ B, BC→ D, E→ C, D→ A. Write the candidate keys of R.

$R = (A, B, C, D, E, H)$

$A \longrightarrow B$
$BC \longrightarrow D$
$E \longrightarrow C$
$D \longrightarrow A$

} Consider RHS
Missing elements are
possible Candidate keys

E, H are missing. Try combinations that make R.

$(EA)^+ \longrightarrow E, A, B, C, D$ (Not possible)

$(EAH)^+ \longrightarrow E, A, H, B, C, D$ (Candidate key)

$(EBH)^+ \longrightarrow E, B, H, C, D, A$ (Candidate key)

$(EDH)^+ \longrightarrow E, D, H, C, A, B$ (Candidate key

* AEH, BEH, DEH are Candidate keys

7.Consider the following relational schemes for a library database: Book (Title, Author, Catalog no, Publisher, Year, Price) Collection (Title, Author, Catalog no) the following are functional dependencies: Title Author → Catalog no Catalog no → Title Author Publisher Year Publisher Title Year → Price Assume Author, Title is the key for both schemes. Apply the appropriate normal form for Book Cancellation.

Book - bcnf (so it is 3nf ,2nf,1nf as well)
Collection - 3nf
Explanation :

- The table “Collection” is in BCNF as there is only one functional dependency “Title Author ⟶ Catalog_no” and {Author, Title} is key for collection.
- Book is not in BCNF because Catalog_no is not a key and there is a functional dependency “Catalog_no ⟶ Title Author Publisher Year”.
- Book is not in 3NF because non-prime attributes (Publisher Year) are transitively dependent on key [Title, Author].
- Book is in 2NF because every non-prime attribute of the table is either dependent on the whole of a candidate key [Title, Author], or on another non-prime attribute.

In the table book, candidate keys are {Title, Author} and {Catalog_no}. In table Book, non-prime attributes (attributes that do not occur in any candidate key) are Publisher, Year and Place

8. Consider a schema R (A, B, C, D) and functional dependencies A → B and C →D. Solve and find whether the decomposition of R into R1 (A, B) and R2(C, D) belongs to which one or both (dependency preserving and lossless join)?

Dependency preserving but not lossless join

Explanation: Dependency Preserving Decomposition**:**
Decomposition of R into R1 and R2 is a dependency preserving decomposition if closure of functional dependencies after decomposition is same as closure of FDs before decomposition. A simple way is to just check whether we can derive all the original FDs from the FDs present after decomposition.
In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D) and there are only two FDs A → B and C → D. So, the decomposition is dependency preserving.

Lossless-Join Decomposition:

Decomposition of R into R1 and R2 is a lossless-join decomposition if at least one of the following functional dependencies are in F+ (Closure of functional dependencies)

In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D), and R1 ∩ R2 is empty. So, the decomposition is not lossless.

9. Consider the relation R(A,B,C,D,E,F) and FDs A →BC, F→ A, C→ AD → E, E → D AD is the decomposition of R into R1(A,C,D) R2 (B,C,D) and R3 (E,F,D) loss less? Explain the requirement of Lossless decomposition.

FDs are not proper in the question.

The decomposition is lossless when it satisfies the following statement –

- If we union the sub Relation R1 and R2 then it must contain all the attributes that are available in the original relation R before decomposition.
- Intersections of R1 and R2 cannot be Null. The sub relation must contain a common attribute. The common attribute must contain unique data.

10. Suppose the schema R(A,B,C,D,E) is decomposed into (A,B,C) and (A,D,E)show that the decomposition is not a dependency preserving decomposition if the following set of FD hold A→ BC, CD→ E, B→ D, E →A

R1 ∩ R2 = A; (A → BC) (A → ABC) (R1 ∩ R2 → R1) this is a lossless-join decomposition.

PART B

1 **Define a View in SQL. Write about updates on views.**

Views in SQL

- Views in SQL are considered as a virtual table. A view also contains rows and columns.
- To create the view, we can select the fields from one or more tables present in the database.
- A view can either have specific rows based on certain condition or all the rows of a table.

1. Creating view

A view can be created using the CREATE VIEW statement. We can create a view from a single table or multiple tables.

Syntax:

CREATE VIEW view_name AS

SELECT column1, column2.....

FROM table_name

WHERE condition;

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

4. Deleting View

A view can be deleted using the Drop View statement.

Syntax

DROP VIEW view_name;

For more information refer [DBMS SQL View - javatpoint]

2 Illustrate Group by and Having clauses with examples.

1. Having Clause :

Having Clause is basically like the aggregate function with the GROUP BY clause. The HAVING clause is used instead of WHERE with aggregate functions. While the GROUP BY Clause groups rows that have the same values into summary rows. The having clause is used with the where clause in order to find rows with certain conditions. The having clause is always used after the group By clause.

```
SELECT COUNT (SALARIES) AS COUNT_SALARIES, EMPLOYEES
FROM EMPLOYEES
GROUP BY SALARIES
HAVING COUNT(SALARIES) > 1;
```

2. Group By Clause :

The GROUP BY clause is often used with aggregate functions (MAX, SUM, AVG) to group the results by one or more columns or In simple words we can say that The GROUP BY clause is used in collaboration with the SELECT statement to arrange required data into groups.

The GROUP BY statement groups rows that have the same values. This Statement is used after the where clause. This statement is often used with some aggregate function like SUM, AVG, COUNT atc. to group the results by one or more columns.

```
SELECT COUNT (SALARIES) AS COUNT_SALARIES, EMPLOYEES
FROM EMPLOYEES
GROUP BY SALARIES;
```

3 Discuss about Complex integrity constraints in SQL

4 Write a nested query to find the names of sailors who have reserved both a red and green boat. Write a nested query to find the names of sailors who have reserved all boats.

5 Discuss various DML statements in SQL and explain with Examples.

It is used for accessing and manipulating data in a database. It handles user requests.

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Delete: It is used to delete all records from a table.

Merge: It performs UPSERT operation, i.e., insert or update operations.

Call: It is used to call a structured query language or a Java subprogram

6 define referential integrity constraint, unique key. Is unique +not null is same as primary key

7 What are nested queries? What is correlation in nested queries? Explain

8 Consider the following schema instructor (ID, name, dept name), teaches (ID, course id, sec id, semester, year), section (course id, sec id, semester, year), student (ID, name, dept name), takes (ID, course id, sec id, semester, year, grade) Write the following queries in SQL 1. Find the names of the students not registered in any section 2. Find the total number of courses taught department wise 3. Find the total number of courses registered department wise

9

10 Define functional dependencies. How are primary keys related to FDs?

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

X → Y

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

For example:

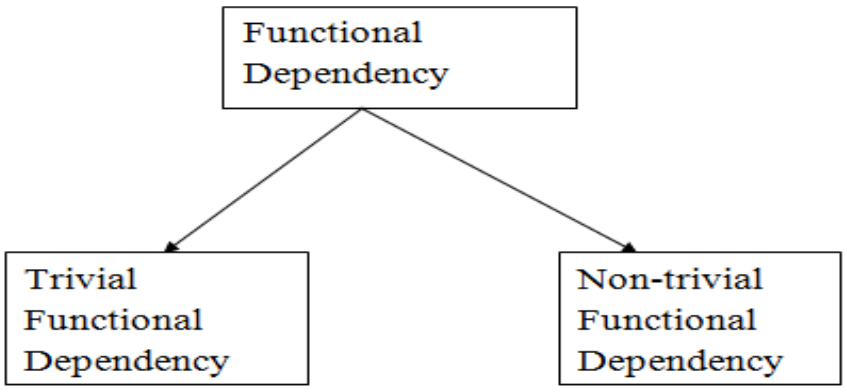
Assume we have an employee table with attributes: Emp_Id, Emp_Name, Emp_Address.

Here Emp_Id attribute can uniquely identify the Emp_Name attribute of employee table because if we know the Emp_Id, we can tell that employee name associated with it.

Functional dependency can be written as:

1. Emp_Id → Emp_Name

We can say that Emp_Name is functionally dependent on Emp_Id.



1. Trivial functional dependency

- A → B has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: A → A, B → B

2. Non-trivial functional dependency

- A → B has a non-trivial functional dependency if B is not a subset of A.
- When A intersection B is NULL, then A → B is called as complete non-trivial.

Refer DBMS Functional Dependency - javatpoint

11. Define normalization? Explain 1NF, 2NF, 3NF Normal forms

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.
- The main reason for normalizing the relations is removing these anomalies

1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.

Extra info visit : [DBMS Normalization: 1NF, 2NF, 3NF and BCNF with Examples - javatpoint]

12. Describe properties of decompositions ?

Properties of Relational Decomposition

1. **Attribute Preservation**
   The attributes in R will appear in at least one relation schema Ri in the decomposition, i.e., no attribute is lost. This is called the Attribute Preservation condition of decomposition.

2. **Dependency Preservation**
   If each functional dependency X→Y specified in F appears directly in one of the relation schemas Ri in the decomposition D or could be inferred from the dependencies that appear in some Ri. This is the Dependency Preservation.

3. **Non Additive Join Property**
   Another property of decomposition that D should possess is the Non Additive Join Property, which ensures that no spurious tuples are generated when a NATURAL JOIN operation is applied to the relations resulting from the decomposition.

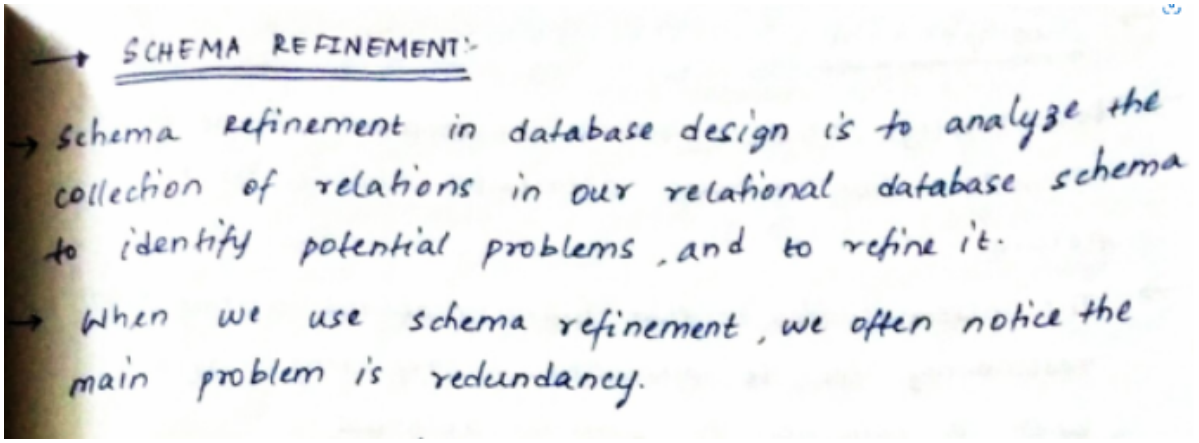
4. **No redundancy**
   Decomposition is used to eliminate some of the problems of bad design like anomalies, inconsistencies, and redundancy.If the relation has no proper decomposition, then it may lead to problems like loss of information.

5. **Lossless Join**

Lossless join property is a feature of decomposition supported by normalisation. It is the ability to ensure that any instance of the original relation can be identified from corresponding instances in the smaller relations.

13. Explain about Schema refinement in Database design ?

Normalisation or Schema Refinement is a technique of organising the data in the database. It is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies.



Problems Caused by Redundancy

Storing the same information redundantly, that is, in more than one place within a database, can lead to several problems:

Redundant storage: Some information is stored repeatedly.

Update anomalies: If one copy of such repeated data is updated, an inconsistency is created unless all copies are similarly updated.

Insertion anomalies: It may not be possible to store some information unless some other information is stored as well.

Deletion anomalies: It may not be possible to delete some information without losing some other information as well

14. Illustrate multivalued dependencies and Fourth normal form with example ?

For a table to satisfy the Fourth Normal Form, it should satisfy the following two conditions:

1. It should be in the Boyce-Codd Normal Form.
2. And, the table should not have any Multi-valued Dependency.

A table is said to have multi-valued dependency, if the following conditions are true,

1. For a dependency A → B, if for a single value of A, multiple value of B exists, then the table may have multi-valued dependency.
2. Also, a table should have at-least 3 columns for it to have a multivalued dependency.
3. And, for a relation R(A,B,C), if there is a multi-valued dependency between A and B, then B and C should be independent of each other.

If all these conditions are true for any relation(table), it is said to have multi-valued dependency

Example :

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
2	C#	Cricket
2	Php	Hockey

As you can see in the table above, a student with s_id 1 has opted for two courses, Science and Maths, and has two hobbies, Cricket and Hockey.

You must be thinking what problem this can lead to, right?

Well the two records for students with s_id 1, will give rise to two more records, as shown below, because for one student, two hobbies exist, hence along with both the courses, these hobbies should be specified.

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
1	Science	Hockey
1	Maths	Cricket

And, in the table above, there is no relationship between the columns course and hobby. They are independent of each other.

So there is multivalued dependency, which leads to unnecessary repetition of data and other anomalies as well. To make the above relation satisfy the 4th normal form, we can decompose the table into 2 tables.

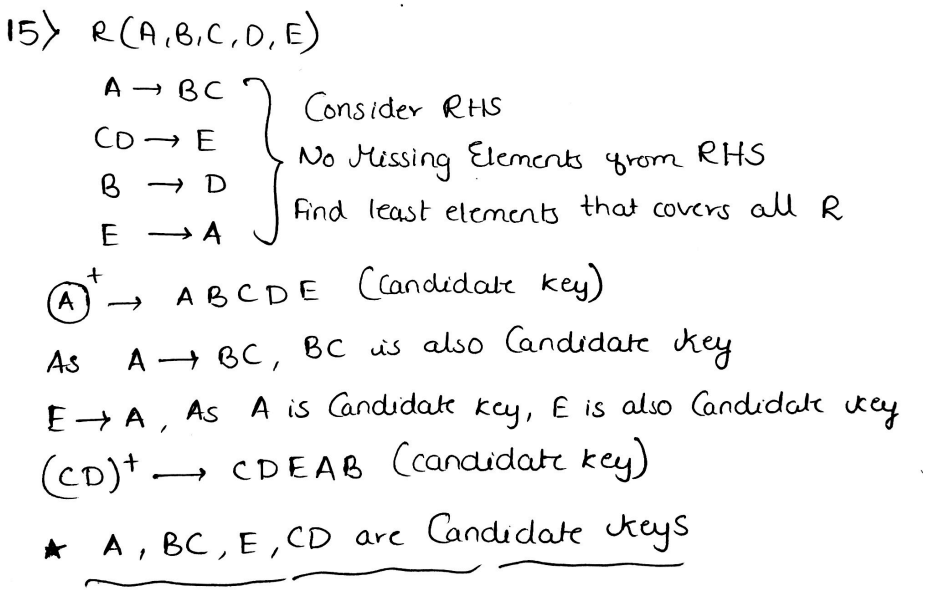
CourseOpted Table

s_id	course
1	Science
1	Maths
2	C#
2	Php

And, **Hobbies Table**,

s_id	hobby
1	Cricket
1	Hockey
2	Cricket
2	Hockey

15. Compute the closure of the following set of functional dependencies for a relation scheme. R(A,B,C,D,E) F= A→BC,CD→E,B→D, E→ A List out the candidate keys of R.



16. Write a note on INSERT, DELETE, UPDATE commands in SQL

INSERT , UPDATE , and DELETE are all functions in SQL that help you ensure your data is up-to-date and kept clear of unnecessary or outdated information. INSERT , UPDATE , and DELETE , as well as SELECT and MERGE, are known as Data Manipulation Language (DML) statements, which let SQL users view and manage data.

The INSERT INTO command is used to insert new rows in a table.

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO Customers (CustomerName, ContactName, Address, City,
PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger',
'4006', 'Norway');
```

The DELETE statement is used to delete existing records in a table.

```
DELETE FROM table_name WHERE condition;
```

The UPDATE statement is used to modify the existing records in a table.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

17. R(ABCD) is related to the FD set C→D, C→A, B→C. Find

i. Candidate Key

ii. Normal form that can be existed

iii. Decompose in BCNF relations

18. Explain the key constraints Primary key and Foreign key with examples?

Comparison Basis	Primary Key	Foreign Key
Basic	It is used to identify each record into the database table uniquely.	It is used to links two tables together. It means the foreign key in one table refers to the primary key of another table.
NULL	The primary key column value can never be NULL.	The foreign key column can accept a NULL value.
Count	A table can have only one primary key.	A table can have more than one foreign key.
Duplication	The primary key is a unique attribute; therefore, it cannot stores duplicate values in relation.	We can store duplicate values in the foreign key column.
Indexing	The primary key is a clustered index by default, which means it is indexed automatically.	A foreign key is not a clustered index by default. We can make clustered indexes manually.
Deletion	The primary key value can't be removed from the table. If you want to delete it, then make sure the referencing foreign key does not contain its value.	The foreign key value can be removed from the table without bothering that it refers to the primary key of another table.
Insertion	We can insert the values into the primary key column without any limitation, either it present in a foreign key or not.	The value that is not present in the column of a primary key cannot be inserted into the referencing foreign key.
Temporary table	The primary key constraint can be defined on the temporary tables.	A foreign key constraint cannot be defined on the temporary tables.
Relationship	It cannot create a parent-child relationship in a table.	It can make a parent-child relationship in a table.

19. Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

$$\rho(R1, Catalog)$$
$$\rho(R2, Catalog)$$
$$\pi_{R1.sid,R2.sid}(\sigma_{R1.pid=R2.pid \wedge R1.sid != R2.sid \wedge R1.cost > R2.cost}(R1 \times R2))$$

20. Find the sids of suppliers who supply some red part and some green part

$$\rho(R1, \pi_{sid}((\pi_{pid}\ \sigma_{color='red'} Parts) \bowtie Catalog))$$
$$\rho(R2, \pi_{sid}((\pi_{pid}\ \sigma_{color='green'} Parts) \bowtie Catalog))$$
$$R1 \cap R2$$

IARE '24 DISCORD TEAM

II-II

DBMS MODULE 4 SOLUTIONS

IKRAM • VISHAL • VISHNU • UJJWAL

TRANSACTION MANAGEMENT

Few Pending

DBMS MODULE 4

PART A

1. Consider the following transactions with data items P and Q initialised to zero:
T1: read(P); read(Q); If P = 0 then Q := Q + 1; write(Q);
T2: read(Q); read(P); If Q = 0 then P := P + 1; write(P);
Solve and find non-serial interleaving of T1 and T2 for concurrent execution leads to a serializable schedule or non serializable schedule. Explain.

Two or more actions are said to be in conflict if:

1. The actions belong to different transactions.
2. At least one of the actions is a write operation.
3. The actions access the same object (read or write).

The schedules S1 and S2 are said to be conflict-equivalent if the following conditions are satisfied:

1. Both schedules S1 and S2 involve the same set of transactions (including ordering of actions within each transaction).
2. The order of each pair of conflicting actions in S1 and S2 are the same.

A schedule is said to be conflict-serializable when the schedule is conflict-equivalent to one or more serial schedules.
In the given scenario, there are two possible serial schedules:

1. T1 followed by T2
2. T2 followed by T1

In both of the serial schedules, one of the transactions reads the value written by the other transaction as a first step. Therefore, any non-serial interleaving of T1 and T2 will not be conflict serializable.

2. Analize which of the following concurrency control protocols ensure both conflict serializability and freedom from deadlock?

(a)z – phase Locking

(b)Timestamp – ordering

(a) z – phase Locking is a concurrency control method that guarantees serializability. The protocol utilizes locks, applied by a transaction to data, which may block (interpreted as signals to stop) other transactions from accessing the same data. During the transaction's life. 2PL may lead to deadlocks that result from the mutual blocking of two or more transactions.

(b) Timestamp – ordering concurrency protocol ensures both conflict serializability and freedom from deadlock.

- Timestamp based Protocol in DBMS is an algorithm which uses the System Time or Logical Counter as a timestamp to serialise the execution of concurrent transactions. The Timestamp-based protocol ensures that every conflicting read and write operations are executed in a timestamp order.
- The older transaction is always given priority in this method. It uses system time to determine the time stamp of the transaction. This is the most commonly used concurrency protocol.

Only (b) is correct.

3. Suppose that we have only two types of transactions, T1 and T2. Transactions preserve database consistency when run individually. We have defined several integrity constraints such that the DBMS never executes any SQL statements that leads the database into an inconsistent state. Assume that the DBMS does not perform any concurrency control. Give an example schedule of two transactions T1 and T2 that satisfies all these conditions, yet produces a database instance that is not the result of any serial execution of T1 and T2.

Example Schedule (Cont.)

- Serial Schedule and *equivalent non*-serial Schedule:
  (Start: A = \$100, B = \$100, A+B = \$200)

T_1	T_2
read(A)	
$A := A - 50$	
write (A)	
read(B)	
$B := B + 50$	
write(B)	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
	read(B)
	$B := B + temp$
	write(B)

T_1	T_2
read(A)	
$A := A - 50$	
write(A)	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
read(B)	
$B := B + 50$	
write(B)	
	read(B)
	$B := B + temp$
	write(B)

A = \$45, B = \$155, A+B = \$200

4. Suppose that there is a database system that never fails. Analyse whether a recovery manager is required for this system.

14.1 Suppose that there is a database system that never fails. Is a recovery manager required for this system?

Answer:

Even in this case the recovery manager is needed to perform roll-back of aborted transactions.

5. Explain the immediate database modification technique for using the log to ensure transaction atomicity despite failures?

- The log is a sequence of records. Log of each transaction is maintained in some stable storage so that if any failure occurs, then it can be recovered from there.
- If any operation is performed on the database, then it will be recorded in the log.
- But the process of storing the logs should be done before the actual transaction is applied in the database.

There are two approaches to modify the database:

Deferred database modification:

- The deferred modification technique occurs if the transaction does not modify the database until it has committed.
- In this method, all the logs are created and stored in the stable storage, and the database is updated when a transaction commits.

Immediate database modification:

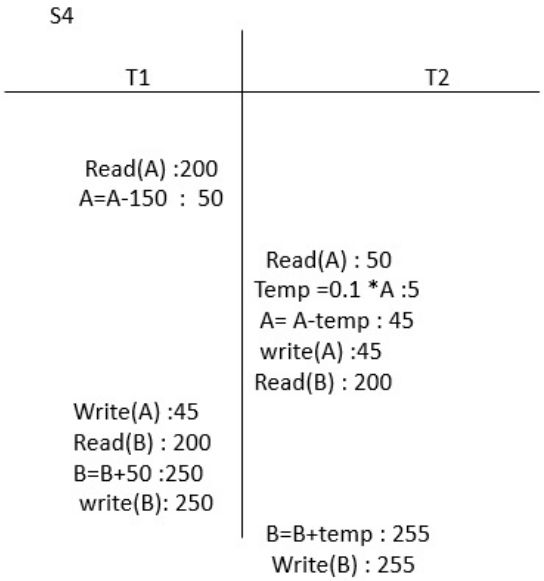
- The Immediate modification technique occurs if database modification occurs while the transaction is still active.
- In this technique, the database is modified immediately after every operation. It follows an actual database modification.

- When a system crashes or a transaction fails, the old value of the data item should be used for bringing the database into the consistent state. This can be done by undoing operation.

6. Consider the following actions taken by transaction T1 on database objects X and Y: R(X), W(X), R(Y), W(Y). Give an example of another transaction T2 that, if run concurrently, could interfere with T1.

1. **Explain how the use of strict 2PL would prevent interference between the two transactions.**
2. **Strict 2PL is used in many database systems. Give two reasons for its popularity.**

Example



Let T1 and T2 are two transactions.

T1=A+B and T2=B+A

T1	T2
Lock-X(A)	Lock-X(B)
Read A;	Read B;
Lock-X(B)	Lock-X(A)

Here,

Lock-X(B) : Cannot execute Lock-X(B) since B is locked by T2.

Lock-X(A) : Cannot execute Lock-X(A) since A is locked by T1.

In the above situation T1 waits for B and T2 waits for A. The waiting time never ends. Both the transaction cannot proceed further at least any one releases the lock voluntarily. This situation is called deadlock.

Strict 2PL is popular for many reasons. One reason is that it ensures only 'safe' interleaving of transactions so that transactions are recoverable, avoid cascading aborts, etc. Another reason is that strict 2PL is very simple and easy to implement.

7. Suppliers (sid: integer, sname: string address: string), Parts(pid: integer, pname: string, colour: string), Catalog (sid: integer, pid: integer, cost: real). The catalog relation lists the prices charged for parts by suppliers. For each

of the following transactions, state the SQL isolation level that you would use and explain why you chose it.

1. **A transaction that adds a new part to a supplier's catalog.**
2. **A transaction that increases the price that a supplier charges for a part.**

8. Answer each of the following questions briefly. The questions are based on the following relational schema: Emp(eid: integer, ename: string, age: integer, salary: real, did: integer), Dept(did: integer, dname: string, floor: integer) and on the following update command:

- **Replace (salary = 1.1 * EMP salary) where EMP.name = Santa**

1. **Give an example of a query that would conflict with this command (in a concurrency control sense) if both were run at the same time.**
2. **Explain what could go wrong, and how locking tuples would solve the problem.**
3. **Give an example of a query or a command that would conflict with this command, such that the conflict could not be resolved by just locking individual tuples or pages but requires index locking.**

9. Suppose that we have only two types of transactions, T1 and T2. Transactions preserve database consistency when run individually. We have defined several integrity constraints such that the DBMS never brings the database into an inconsistent state. Assume that the DBMS does not perform any concurrency control. Give an example schedule of two transactions T1 and T2 that satisfies all these conditions, yet produces a database instance that is not the result of any serial execution of T1 and T2.

10. What are the roles of the analysis, redo, and undo phases in ARIES?

The most advanced & most difficult recovery algorithm is ARIES recovery algorithm. The advantage of this recovery technique is that it reduces recovery time as well as overhead of a log.

ARIES stands for “Algorithm for Recovery and Isolation Exploiting Semantics.”

ANALYSIS PHASE :-

- The dirty page table has been formed by analysing the pages from the buffer and also a set of active transactions has been identified. When the system encounters crash, ARIES recovery manager starts the analysis phone by finding the last checkpoint log record after that, it prepares dirty pages tables this phone mainly prepares a set of active transactions that are needed to be undo analysis phone, after getting the last checkpoint log record, log record is scanned in forward direction, & update of the set of active transactions, transaction value, & dirty page table are done in the following manner :-
- The 1st recovery manager finds any transaction which is not in the active transaction set, then adds that transaction in that set if it finds an end log record, then that record has been deleted from the transaction table.
- If it finds a log record that describes an update operation, the log record has been added to the dirty page table.

REDO PHASE:-

- Redo phase is the second phase where all the transactions that are needed to be executed again take place.
- It executes those operations whose results are not reflected in the disk.
- It can be done by finding the smallest LSN of all the dirty page in dirty page table that defines the log positions, & the Redo operation will start from this position
- This position indicates that either the changes that are made earlier are I the main memory or they have already been flaunted to the disk.
- Thus, for each change recorded in the log, the Redo phase determines whether or not the operations have been re-executed.

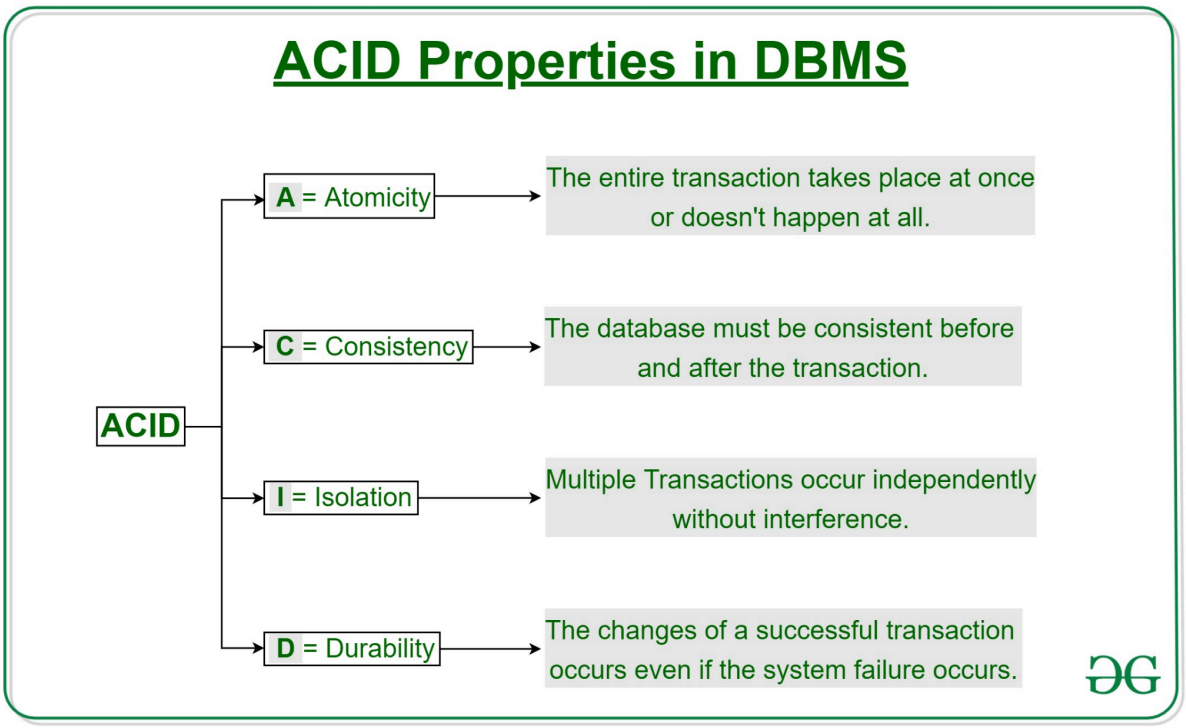
UNDO PHASE:-

- In the Undo phase, all the transactions that are listed in the active transaction are set here to be undone.
- Thus the log should be scanned background from the end & the recovery manager should Undo the necessary operations.
- Each time an operation is undone, a compensation log recorded has been written to the log.
- This process continues until there is no transaction left in the active transaction set.
- After the successful completion of this phase, the database can resume its normal operations.

PART B

Ikram(1-20)

1. Explain ACID properties and illustrate them through examples.



Atomicity By this, we mean that either the entire 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 is not executed at all. It involves the following two operations.

- Abort: If a transaction aborts, changes made to the database are not visible.
- Commit: If a transaction commits, changes made are visible.

Atomicity is also known as the 'All or nothing rule'.

Consider the following transaction T consisting of T1 and T2: Transfer of 100 from account X to account Y.

Before: X : 500	Y: 200
Transaction T	
T1	**T2**
Read (X) X: = X – 100 Write (X)	Read (Y) Y: = Y + 100 Write (Y)
After: X : 400	Y : 300

If the transaction fails after completion of T1 but before completion of T2.(say, after write(X) but before write(Y)), then the amount has been deducted from X but not added to Y. This results in an inconsistent database state. Therefore, the transaction must be executed in its entirety in order to ensure the correctness of the database state.

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. Referring to the example above,

The total amount before and after the transaction must be maintained.

Total before T occurs = 500 + 200 = 700.

Total after T occurs = 400 + 300 = 700.

Therefore, the database is consistent. Inconsistency occurs in case T1 completes but T2 fails. As a result, T is incomplete.

Isolation:

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring 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 state achieved if these were executed serially in some order.

Let X= 500, Y = 500.

Consider two transactions T and T”.

T	T''
Read (X) X: = X*100 Write (X) Read (Y) Y: = Y – 50 Write (Y)	Read (X) Read (Y) Z: = X + Y Write (Z)

This results in database inconsistency, due to a loss of 50 units. Hence, transactions must take place in isolation and changes should be visible only after they have been made to the main memory.

Durability:

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

2. Discuss how you implement Atomicity and Durability?

The recovery-management component of a database system implements the support for atomicity and durability.

Example of the shadow-database scheme: all updates are made on a shadow copy of the database db pointer is made to point to the updated shadow copy after the transaction reaches partial commit and all updated pages have been flushed to disk. Db pointer always points to the current consistent copy of the database. In case a transaction fails, old consistent copy pointed to by the db pointer can be used, and the shadow copy can be deleted. The shadow-database scheme:

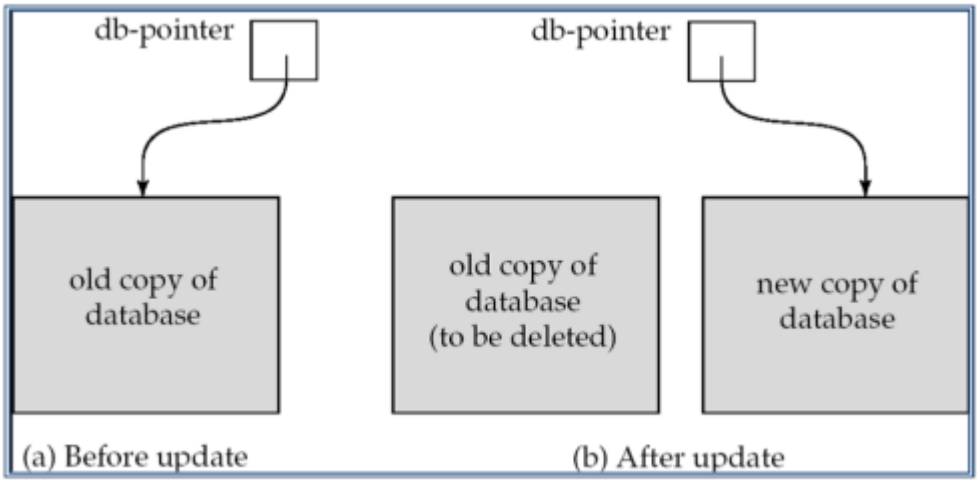


FIGURE 4.2: shadow database

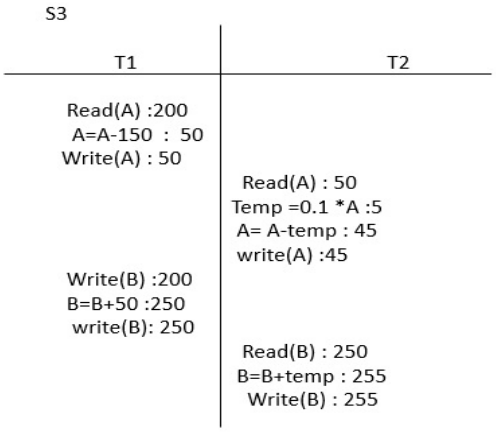
Assumes that only one transaction is active at a time. Assume disks do not fail Useful for text editors, but extremely inefficient for large databases (why?) Variant called shadow paging reduces copying of data, but is still not practical for large databases does not handle concurrent transactions

3. Illustrate concurrent execution of transactions with examples.

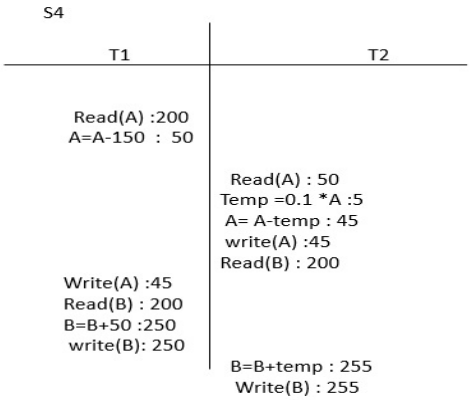
- In a multi-user system, multiple users can access and use the same database at one time, which is known as the concurrent execution of the database. It means that the same database is executed simultaneously on a multi-user system by different users.
- While working on the database transactions, there occurs the requirement of using the database by multiple users for performing different operations, and in that case, concurrent execution of the database is performed.
- The thing is that the simultaneous execution that is performed should be done in an interleaved manner, and no operation should affect the other executing operations, thus maintaining the consistency of the database. Thus, on making the concurrent execution of the transaction operations, there occur several challenging problems that need to be solved.
- In a database transaction, the two main operations are READ and WRITE operations. So, there is a need to manage these two operations in the

concurrent execution of the transactions as if these operations are not performed in an interleaved manner, the data may become inconsistent.

Example:



1)



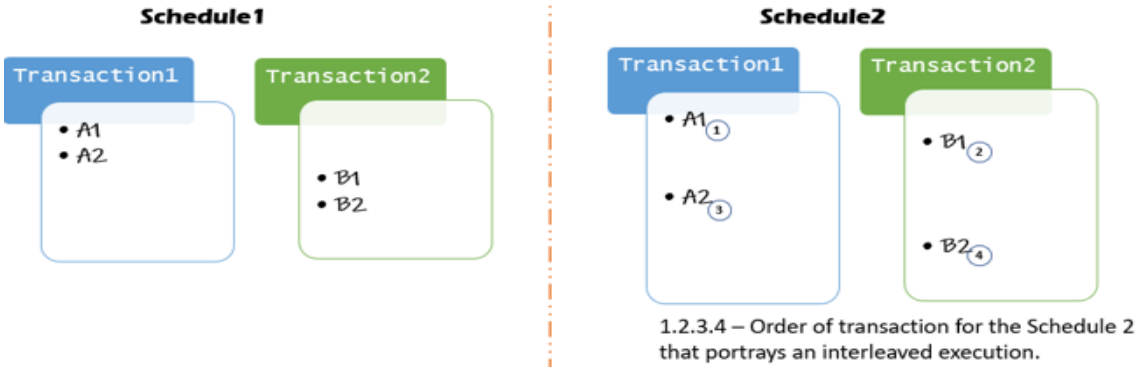
2)

4. Discuss serializability in detail with an example.

Serializability can be called a process used for finding the correct non-serial schedules in the database. It basically helps maintain the consistency in the database and often relates to the isolation features of a transaction.

Serial schedule both by definition and execution means that the transactions bestowed upon it will take place serially, that is, one after the other. This leaves no place for inconsistency within the database. But, when a set of transactions are scheduled non-serially, they are interleaved leading to the problem of concurrency within the database. Non-serial schedules do not wait for one transaction to complete for the other one to begin. Serializability in DBMS decides if an interleaved non-serial schedule is serializable or not.

EXAMPLE Consider 2 schedules, Schedule1 and Schedule2:



1.2.3.4 – Order of transaction for the Schedule 2 that portrays an interleaved execution.

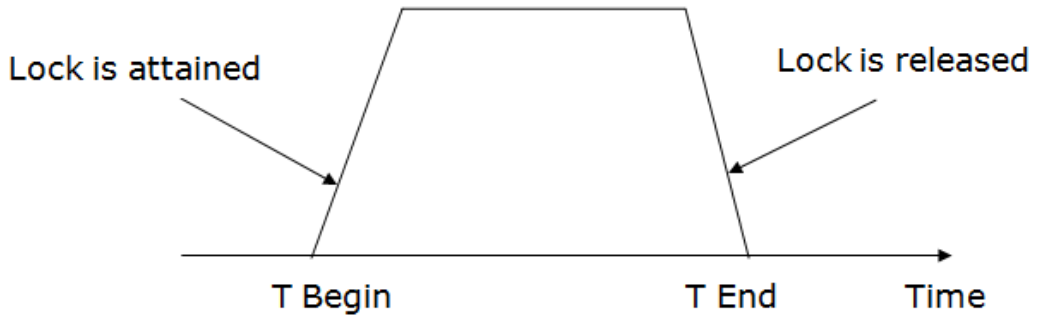
Schedule1 is a serial schedule consisting of Transaction1 and Transaction2 wherein the operations on data item A (A1 and A2) are performed first and later the operations on data item B (B1 and B2) are carried out serially.
Schedule2 is a non-serial schedule consisting of Transaction1 and Transaction2 wherein the operations are interleaved.
Explanation: In the given scenario, Schedule2 is serializable if the output obtained from both Schedule2 and Schedule1 are equivalent to one another. In a nutshell, a transaction within a given non-serial schedule is serializable if its outcome is equivalent to the outcome of the same transaction when executed serially.

5. Discuss two phase locking protocol and strict two phase locking protocols.

Two-phase locking (2PL)

- The two-phase locking protocol divides the execution phase of the transaction into three parts.
- In the first part, when the execution of the transaction starts, it seeks permission for the lock it requires.
- In the second part, the transaction acquires all the locks. The third phase is started as soon as the transaction releases its first lock.
- In the third phase, the transaction cannot demand any new locks. It only releases the acquired locks.

DBMS Lock-Based Protocol



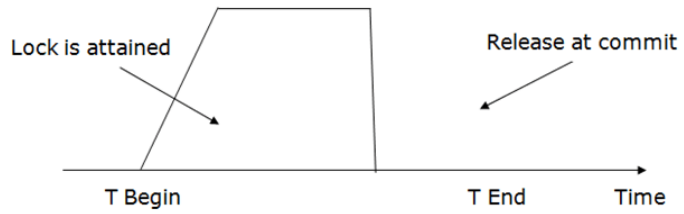
There are two phases of 2PL:

Growing phase: In the growing phase, a new lock on the data item may be acquired by the transaction, but none can be released.

Shrinking phase: In the shrinking phase, existing locks held by the transaction may be released, but no new locks can be acquired.

Strict Two-phase locking (Strict-2PL)

- The first phase of Strict-2PL is similar to 2PL. In the first phase, after acquiring all the locks, the transaction continues to execute normally.
- The only difference between 2PL and strict 2PL is that Strict-2PL does not release a lock after using it.
- Strict-2PL waits until the whole transaction to commit, and then it releases all the locks at a time.
- Strict-2PL protocol does not have a shrinking phase of lock release.



It does not have cascading abort as 2PL does.

6. Discuss timestamp based locking protocols.

Concurrency Control Protocols

1. Lock-Based Protocols
2. Two Phase Locking Protocol
3. Timestamp-Based Protocols
4. Validation-Based Protocols

- Timestamp based Protocol in DBMS is an algorithm which uses the System Time or Logical Counter as a timestamp to serialise the execution of concurrent transactions. The Timestamp-based protocol

ensures that every conflicting read and write operations are executed in a timestamp order.

Each transaction is issued a timestamp when it enters the system. If an old transaction Ti has timestamp TS(Ti), a new transaction Tj is assigned a time-stamp TS(Tj) such that TS(Ti) < TS(Tj).

The protocol manages concurrent execution such that the time-stamps determine the serializability order.In order to assure such behavior, the protocol maintains for each data Q two timestamp values:

W-timestamp(Q) is the largest time-stamp of any transaction that executed write(Q) successfully.

R-timestamp(Q) is the largest time-stamp of any transaction that executed read(Q) successfully. The timestamp ordering protocol ensures that any conflicting read and write operations are executed in timestamp order.

- The older transaction is always given priority in this method. It uses system time to determine the time stamp of the transaction. This is the most commonly used concurrency protocol.

Example:

```
Suppose there are there transactions T1, T2, and T3.
T1 has entered the system at time 0010
T2 has entered the system at 0020
T3 has entered the system at 0030
Priority will be given to transaction T1, then transaction T2 and lastly Transaction T3.
```

7. Describe validation - based locking protocols.

Validation based Protocol in DBMS also known as Optimistic Concurrency Control Technique is a method to avoid concurrency in transactions. In this

protocol, the local copies of the transaction data are updated rather than the data itself, which results in less interference while execution of the transaction.

- It allows the parallel execution of transactions to achieve maximum concurrency.
- Its storage mechanisms and computational methods should be modest to minimize overhead.

The Validation based Protocol is performed in the following three phases:

1. Read Phase
2. Validation Phase
3. Write Phase

Read Phase

In the Read Phase, the data values from the database can be read by a transaction but the write operation or updates are only applied to the local data copies, not the actual database.

Validation Phase

In the Validation Phase, the data is checked to ensure that there is no violation of serializability while applying the transaction updates to the database.

Write Phase

In the Write Phase, the updates are applied to the database if the validation is successful, else; the updates are not applied, and the transaction is rolled back.

8. Discuss in detail Multiple Granularity.

Multiple Granularity:

- It can be defined as hierarchically breaking up the database into blocks which can be locked.

- The Multiple Granularity protocol enhances concurrency and reduces lock overhead.
- It maintains the track of what to lock and how to lock.
- It makes it easy to decide either to lock a data item or to unlock a data item. This type of hierarchy can be graphically represented as a tree.

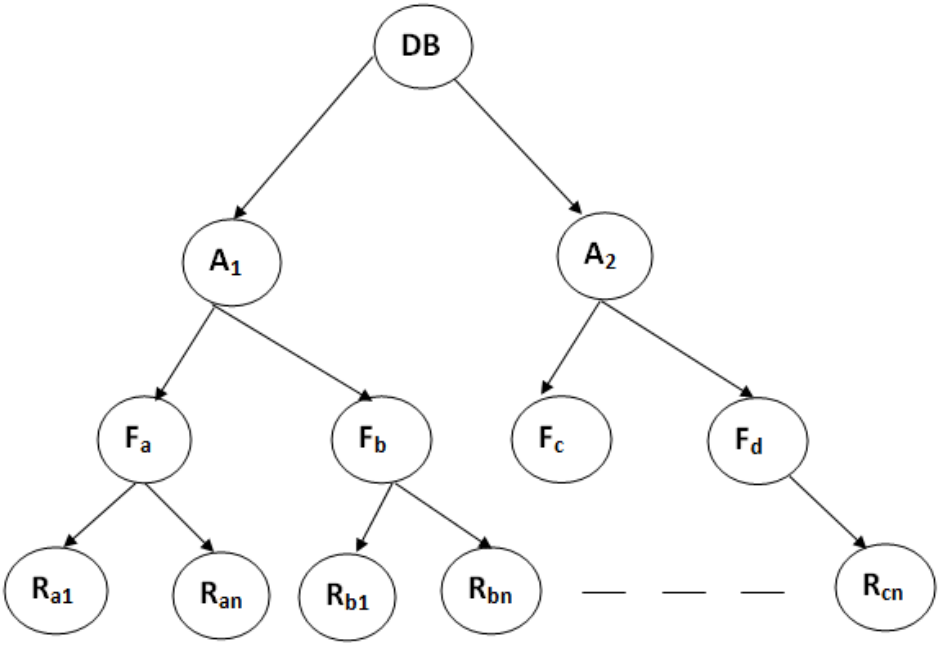


Figure: Multi Granularity tree Hierarchy

The levels of the tree starting from the top level are as follows:

1. Database
2. Area
3. File
4. Record

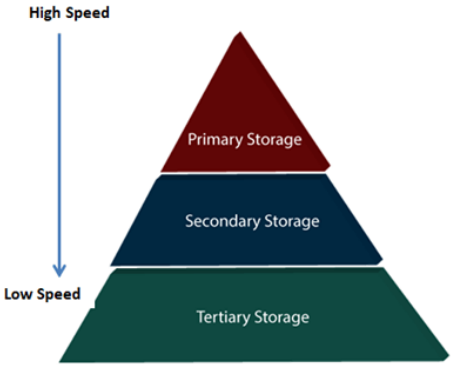
Finally, each file contains child nodes known as records. The file has exactly those records that are its child nodes. No records are represented in more than one file.

- In this example, the highest level shows the entire database. The levels below are file, record, and fields.

9. Explain in detail Storage structure.

A database system provides an ultimate view of the stored data. However, data in the form of bits, bytes get stored in different storage devices.
These storage types differ from one another as per the speed and accessibility. There are the following types of storage devices used for storing the data:

- Primary Storage
- Secondary Storage
- Tertiary Storage



Primary Storage

It is the primary area that offers quick access to the stored data. We also know the primary storage as volatile storage. It is because this type of memory does not permanently store the data.

- Main Memory: It is the one that is responsible for operating the data that is available by the storage medium. The main memory handles each instruction of a computer machine.
- Cache: It is one of the costly storage media. On the other hand, it is the fastest one. A cache is a tiny storage media which is maintained by the computer hardware usually.

Secondary Storage

Secondary storage is also called Online storage. It is the storage area that allows the user to save and store data permanently. This type of memory does

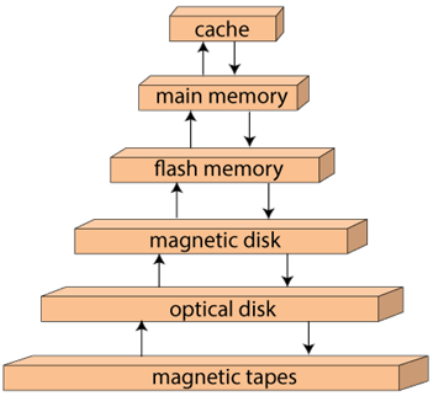
not lose the data due to any power failure or system crash. That's why we also call it non-volatile storage.

- Flash Memory: A flash memory stores data in USB (Universal Serial Bus) keys which are further plugged into the USB slots of a computer system. These USB keys help transfer data to a computer system, but it varies in size limits
- Magnetic Disk Storage: This type of storage media is also known as online storage media. It is used for storing the data for a long time. It is capable of storing an entire database. It is the responsibility of the computer system to make availability of the data from a disk to the main memory for further accessing

Tertiary Storage

It is the storage type that is external from the computer system. It has the slowest speed. But it is capable of storing a large amount of data. It is also known as Offline storage.

- Optical Storage: An optical storage can store megabytes or gigabytes of data.
- Tape Storage: It is the cheapest storage medium than disks. Generally, tapes are used for archiving or backing up the data.



Storage device hierarchy

10. Discuss Deferred database modification and Immediate database modification.

1. Deferred Update :

It is a technique for the maintenance of the transaction log files of the DBMS. It is also called the NO-UNDO/REDO technique. It is used for the recovery of the transaction failures which occur due to power, memory or OS failures. Whenever any transaction is executed, the updates are not made immediately to the database. They are first recorded on the log file and then those changes are applied once a commit is done. This is called the “Re-doing” process. Once the rollback is done none of the changes are applied to the database and the changes in the log file are also discarded. If a commit is done before crashing of the system, then after restarting of the system the changes that have been recorded in the log file are thus applied to the database.

2. Immediate Update :

It is a technique for the maintenance of the transaction log files of the DBMS. It is also called the UNDO/REDO technique. It is used for the recovery of the transaction failures which occur due to power, memory or OS failures. Whenever any transaction is executed, the updates are made directly to the database and the log file is also maintained which contains both old and new values. Once a commit is done, all the changes get stored permanently into the database and records in the log file are thus discarded. Once rollback is done the old values get restored in the database and all the changes made to the database are also discarded. This is called the “Un-doing” process. If a commit is done before crashing of the system, then after restarting of the system the changes are stored permanently in the database.

Short points for understanding-

DBMS Log-Based Recovery - javatpoint

11. Discuss how you recover from Concurrent transactions.

Concurrency control means that multiple transactions can be executed at the same time and then the interleaved logs occur. But there may be changes in transaction results so maintain the order of execution of those transactions.
During recovery, it would be very difficult for the recovery system to backtrack all the logs and then start recovering.
Recovery with concurrent transactions can be done in the following four ways:

- Interaction with concurrency control
- Transaction rollback
- Checkpoints
- Restart recovery

Interaction with concurrency control :

In this scheme, the recovery scheme depends greatly on the concurrency control scheme that is used. So, to rollback a failed transaction, we must undo the updates performed by the transaction.

Transaction rollback :

- In this scheme, we rollback a failed transaction by using the log.
- The system scans the log backward for a failed transaction, for every log record found in the log the system restores the data item.
- Whenever more than one transaction is being executed, then the interleaving of logs occurs. During recovery, it would become difficult for the recovery system to backtrack all logs and then start recovering.
- To ease this situation, 'checkpoint' concept is used by most DBMS.

Checkpoints :

- Checkpoints is a process of saving a snapshot of the applications state so that it can restart from that point in case of failure.

- Checkpoint is a point of time at which a record is written onto the database from the buffers.
- Checkpoint shortens the recovery process.
- When it reaches the checkpoint, then the transaction will be updated into the database, and till that point, the entire log file will be removed from the file. Then the log file is updated with the new step of transaction till the next checkpoint and so on.

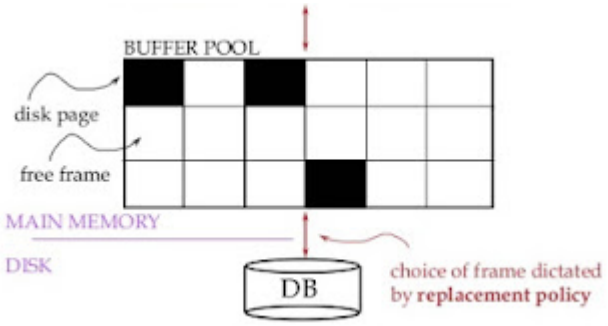
Restart recovery :

- When the system recovers from a crash, it constructs two lists.
- The undo-list consists of transactions to be undone, and the redo-list consists of transactions to be redone.
- The system constructs the two lists as follows: Initially, they are both empty. The system scans the log backward, examining each record, until it finds the first <checkpoint> record.

12. Explain Buffer Management with a neat diagram.

- A Buffer Manager is responsible for allocating space to the buffer in order to store data into the buffer.
- If a user requests a particular block and the block is available in the buffer, the buffer manager provides the block address in the main memory.
- If the block is not available in the buffer, the buffer manager allocates the block in the buffer.
- If free space is not available, it throws out some existing blocks from the buffer to allocate the required space for the new block.
- The blocks which are thrown are written back to the disk only if they are recently modified when writing on the disk.

- If the user requests such thrown-out blocks, the buffer manager reads the requested block from the disk to the buffer and then passes the address of the requested block to the user in the main memory.
- However, the internal actions of the buffer manager are not visible to the programs that may create any problem in disk-block requests. The buffer manager is just like a virtual machine.



13. Explain different types of Advanced Recovery Techniques.

- Advanced Recovery: Logical Undo Logging Operations like B+-tree insertions and deletions release locks early. They cannot be undone by restoring old values (physical undo), since once a lock is released, other transactions may have updated the B+-tree. Instead, insertions (resp. deletions) are undone by executing a deletion (resp. insertion) operation (known as logical undo).For such operations, undo log records should contain the undo operation to be executed. Such logging is called logical undo logging, in contrast to physical undo Logging Operations are called logical operations.
- Advanced Recovery:Physical Redo Redo information is logged physically (that is, new value for each write) even for Operations with logical undo Logical redo are very complicated since database state on disk may not be operation consistent when recovery starts Physical redo logging does not conflict with early lock release.
- Advanced Recovery: Operation Logging Operation logging is done as follows: When operation starts, log <Ti, on, operation begin>. Here on is a unique identifier of the operation instance. While operation is

executing, normal log records with physical redo and physical undo information are logged.When operation completes, <Ti, on, operation-end, U> is logged, where U contains information needed to perform a logical undo information.

- Advanced Recovery: Check pointing Check pointing is done as follows:
  * Output all log records in memory to stable storage
  * Output to disk all modified buffer blocks
  * Output to log on stable storage at <checkpoint L> record.

Transactions are not allowed to perform any actions while checkpointing is in progress. Some more techniques are there in lecture notes page number 69,70.

14. Write in detail about Remote Backup Systems.

Remote backup systems provide high availability by allowing transaction processing to continue even if the primary site is destroyed. Detection of failure: Backup site must detect when primary site has failed . To distinguish primary site failure from link failure, maintain several communication links between the primary and the remote backup.

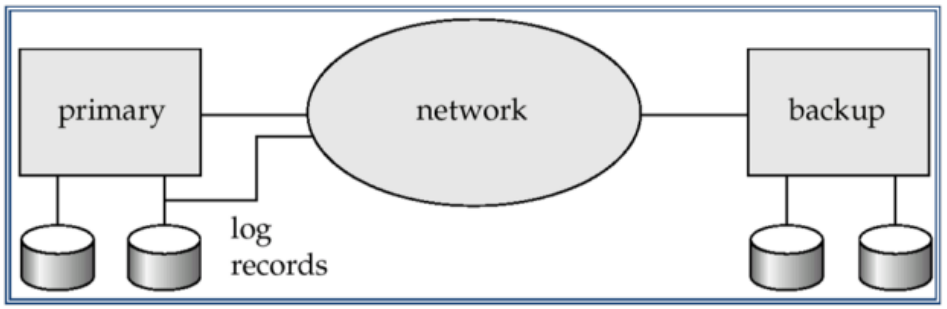


FIGURE 5.13: Remote Backup Systems

Transfer of control:

To take over control, the backup site first performs recovery using its copy of the database and all the long records it has received from the primary. Thus, completed transactions are redone and incomplete transactions are rolled back. When the backup site takes over processing it becomes the new primary

to transfer control back to the old primary. When it recovers, old primary must receive redo logs from the old backup and apply all updates locally.
Time to recover: To reduce delay in takeover, the backup site periodically processes the Redo log records (in effect, performing recovery from previous database state), performs a checkpoint, and can then delete earlier parts of the log. Hot-Spare configuration permits very fast takeover: Backup continually processes redo log records as they arrive, applying the updates locally. When failure of the primary is detected the Backup rolls back incomplete transactions, and is ready to process new transactions. Alternative to remote backup: distributed database with replicated data .Remote backup is faster and cheaper, but less tolerant to failure. Ensure durability of updates by delaying transaction commit until update is logged at backup; avoid this delay by permitting lower degrees of durability. One-safe:
commit as soon as transaction commit log record is written at primary Problem: updates may not arrive at backup before it takes over

15. Explain the Check point log based recovery scheme for recovering the database.

Log-Based Recovery

- The log is a sequence of records. Log of each transaction is maintained in some stable storage so that if any failure occurs, then it can be recovered from there.
- If any operation is performed on the database, then it will be recorded in the log.
- But the process of storing the logs should be done before the actual transaction is applied in the database.

Let's assume there is a transaction to modify the City of a student. The following logs are written for this transaction.

- When the transaction is initiated, it writes 'start' log.
  <Tn, Start>

- When the transaction modifies the City from 'Noida' to 'Bangalore', then another log is written to the file.
  <Tn, City, 'Noida', 'Bangalore' >
- When the transaction is finished, then it writes another log to indicate the end of the transaction.
  <Tn, Commit>

There are two approaches to modify the database:

1. Deferred database modification:
   The deferred modification technique occurs if the transaction does not modify the database until it has committed.
   In this method, all the logs are created and stored in the stable storage, and the database is updated when a transaction commits.
2. Immediate database modification:
   The Immediate modification technique occurs if database modification occurs while the transaction is still active.
   In this technique, the database is modified immediately after every operation. It follows an actual database modification.

Recovery using Log records

When the system crashes, then the system consults the log to find which transactions need to be undone and which need to be redone.

- If the log contains the record <Ti, Start> and <Ti, Commit> or <Ti, Commit>, then the Transaction Ti needs to be redone.
- If log contains record<Tn, Start> but does not contain the record either <Ti, commit> or <Ti, abort>, then the Transaction Ti needs to be undone.

16. When a transaction is rolled back under timestamp ordering, it is assigned a new timestamp. Why can it not simply keep its old timestamps?

A transaction has rolled back means some other transaction has made the changes in the data which it was supposed to do. Now if it returns with the

same timestamp then it will rollback again for the same previous reason and this will continue endlessly hence it is assigned a new timestamp value. (It means that to maintain sequential execution behaviour it will allocate a new timestamp.)

17. Consider the following schedule S1.

- **S1 = r3(y), r3(z), r1(x), w1(x), w3(y), w3(z), r2(z), r1(y), w1(y), r2(y), w2(y), r2(x), w2(x)**

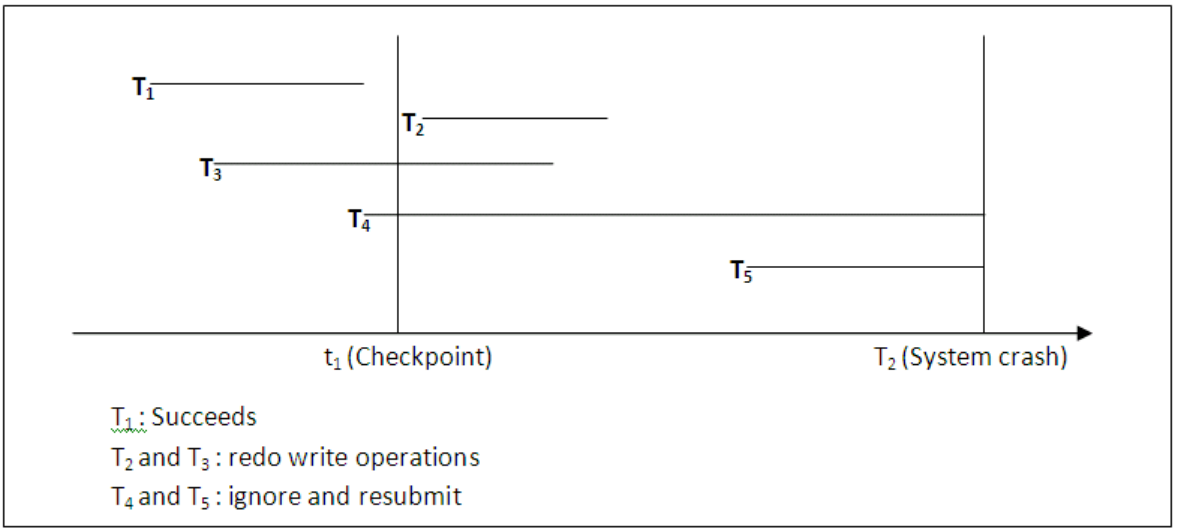
Check whether S1 is serializable or not. If it is serializable, write its equivalent serial schedule.

For conflict serializability of a schedule(which gives same effect as a serial schedule) we should check for conflict operations, which are Read-Write, Write-Read and Write-Write between each pair of transactions, and based on those conflicts we make a precedence graph, if the graph contains a cycle, it's not a conflict serializable schedule.

To make a precedence graph: if Read(X) in Ti followed by Write(X) in Tj (hence a conflict), then we draw an edge from Ti to Tj (Ti → Tj)

Refer [GATE | GATE-CS-2014-(Set-3) | Question 65 - GeeksforGeeks]

18. With a neat diagram explaining NO-UNDO / NO-REDO recovery mechanism in transaction processing.



19. Explain the serializable and non serializable schedule.

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 inconsistency or not. On the other hand, a serial schedule does not need the serializability because it follows a transaction only when the previous transaction is complete. The non-serial schedule is said to be in a serializable schedule only when it is equivalent to the serial schedules, for an n number of transactions. Since concurrency is allowed in this case thus, multiple transactions can execute concurrently. A serializable schedule helps in improving both resource utilisation and CPU throughput.

- Conflict
- view

Non-Serializable:

The non-serializable schedule is divided into two types, Recoverable and Non-recoverable Schedule.

- Recoverable Schedule
- Non-Recoverable Schedule

20. Suppose that there is a database system that never fails. Analyse whether a recovery management required for this system (REPEATED)

Refer 4th question in part A

PART - C

1. Define a Transaction. List the properties of the transaction.

A transaction is a unit of program execution that accesses and possibly updates various data items.

Example transaction to transfer $50 from account A to account B:

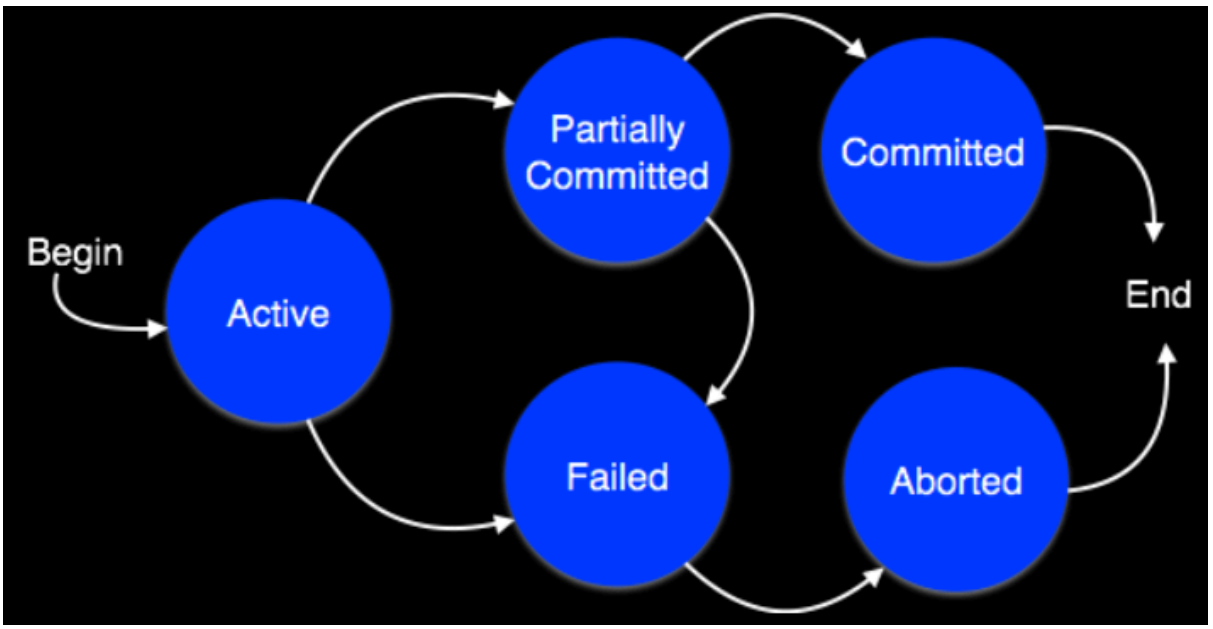
1. read(A)
2. A:=A–50
3. write(A)
4. read(B)
5. B:=B+50
6. write(B)

Properties of Transaction:

- Atomicity
- Consistency
- Durability
- Isolation

So as to ensure accuracy, completeness and data integrity.

2. Discuss different phases of the transaction.



Transaction States are as follows:

- Active: In the initial state, the transaction stays in this state while it is executing.
- Partially committed: After the execution of transaction's final operation, it is said to be in a partially committed state.

- Failed: A transaction is said to be in final state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.
- Aborted: If a transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state. The database recovery mode can select one of the two operations after abort:
    - Re-start
    - Kill

- Committed: If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanent on the database.

3. Discuss recoverable schedules.

A transaction may not execute completely due to hardware failure, system crash or software issues. In that case, we have to roll back the failed transaction. But some other transactions may also have used values produced by the failed transaction. So we have to roll back those transactions as well.

There are three types of recoverable schedules

- Cascading Scheduling
- Cascading less Scheduling
- Strict Scheduling

4. Discuss cascade less schedules.

5. Define two phase commit protocol.

6. Demonstrate the implementation of Isolation.

7. Discuss the procedure to test Serializability.

8. List different types of locks and write about compatibility among them.

9. Discuss about Failure Classification.

10. Define a checkpoint.

11. Discuss the failures that can occur with loss of Non - Volatile storage.

12. Demonstrate conflict Serializability.

13. Discuss view Serializability.

14. Explain the distinction between serializable schedule with examples

15. How is the consistency of a transaction preserved?

16. When two instructions conflict with each other?

17. Indicate the importance of Isolation property of a Transaction.

18. State the property atomicity of a Transaction.

19. Explain about transaction states with a neat diagram.

20. Discuss about Schedule and Recoverability.

IARE '24 DISCORD TEAM

II-II

DBMS MODULE 5 SOLUTIONS

SYED IKRAM • VISHNU • UJJWAL • VARUN TEJA

DATA STORAGE AND QUERY PROCESSING

Completed

DBMS MODULE 5

PART A

1. Consider a B+ tree in which the maximum number of keys in a node is 5, Calculate the minimum number of keys in any non - root node.

Since the maximum number of keys is 5, the maximum number of children a node can have is 6. Definition of B Tree, minimum children that a node can have would be 6/2 = 3. Therefore, the minimum number of keys that a node can have becomes (3-1)=2 keys.

2. In the index allocation schema of blocks to a file, Calculate on what maximum possible size of the file depends.

In the indexed scheme of blocks to a file, the maximum possible size of the file depends on the number of blocks used for index and the size of index.

3. A clustering index is defined on the fields of which type. Analyse them.

We are given that the clustering index is defined on the fields. Now, if the records of the file are physically ordered on a non-key field since it is a non-key, it will not have a distinct value for each record. Therefore, the clustering index is defined on the fields of type non-key and ordering.

4. Calculate the minimum space utilisation for a B+ tree index.

By the definition of a B+ tree, each index page, except for the root, has at least d and at most 2d key entries. Therefore—with the exception of the root—the minimum space utilisation guaranteed by a B+ tree index is 50 percent.

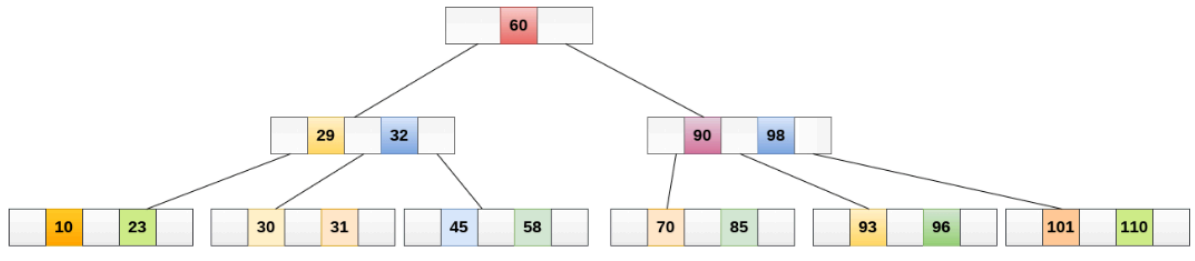
5. Explain about the B tree and the structure of B+ tree in detail with an example.

B Tree is a specialised m-way tree that can be widely used for disk access. A B Tree of order m can have at most m-1 keys and m children. One of the main reasons for using B tree is its capability to store a large number of keys in a single node and large key values by keeping the height of the tree relatively small.

A B tree of order m contains all the properties of an M way tree. In addition, it contains the following properties.

- Every node in a B Tree contains at most m children.
- Every node in a B-Tree except the root node and the leaf node contain at least m/2 children.
- The root nodes must have at least 2 nodes.
- All leaf nodes must be at the same level.

A B tree of order 4 is shown in the following image.



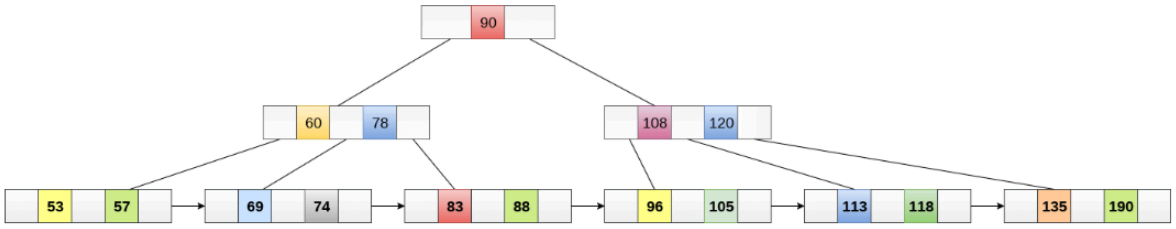
B+ Tree is an extension of B Tree which allows efficient insertion, deletion and search operations.

In B Tree, Keys and records both can be stored in the internal as well as leaf nodes. Whereas, in B+ tree, records (data) can only be stored on the leaf nodes while internal nodes can only store the key values.

The leaf nodes of a B+ tree are linked together in the form of singly linked lists to make the search queries more efficient.

B+ Tree are used to store a large amount of data which can not be stored in the main memory. Due to the fact that, size of main memory is always limited, the internal nodes (keys to access records) of the B+ tree are stored in the main memory whereas leaf nodes are stored in the secondary memory.

The internal nodes of B+ tree are often called index nodes. A B+ tree of order 3 is shown in the following figure.



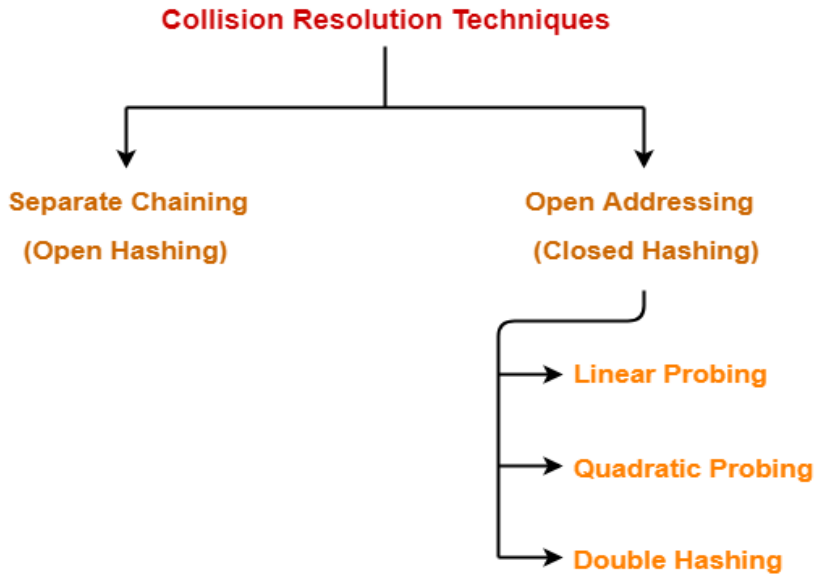
6. Explain the distinction between closed and open hashing. Discuss the relative merits of each technique in database applications

Open hashing may place keys with the same hash function value in different buckets. Closed hashing always places such keys together in the same bucket. Thus in this case, different buckets can be of different sizes, though the implementation may be by linking together fixed size buckets using overflow chains.

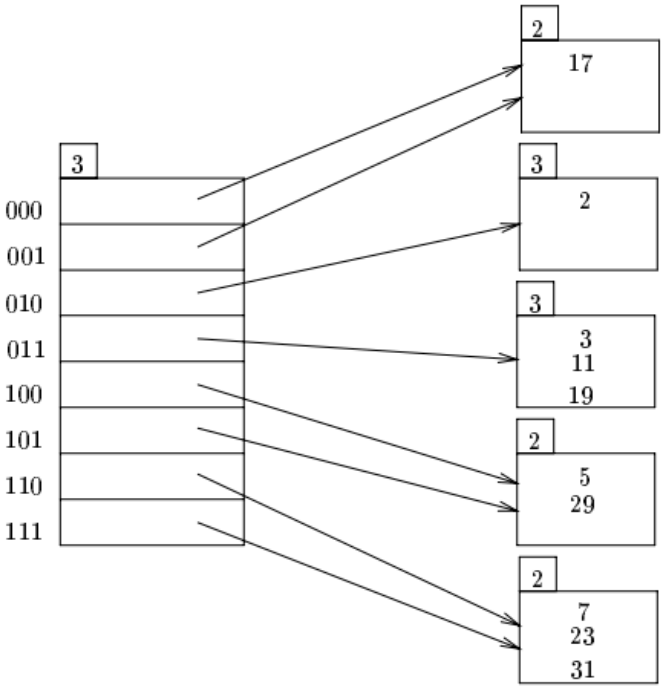
Deletion is difficult with open hashing as all the buckets may have to be inspected before we can ascertain that a key value has been deleted, Whereas in closed hashing only that bucket whose address is obtained by hashing the key value needs to be inspected. Deletions are more common in databases and

hence closed hashing is more appropriate for them. For a small, static set of data lookups may be more efficient using open hashing.

The symbol table of a compiler would be a good example.



7. Suppose that we are using extendible hashing on a file following search - key values: 2, 3, , 7 , 11, 17, 19, 23, 29, 31. Show the extendable hash structure for this file if the hash function is h(x) = x mod 8 and buckets can hold three records



8. Explain various steps in Query Processing and write any two techniques to optimise query.

Query Processing is the activity performed in extracting data from the database. In query processing, it takes various steps for fetching the data from the database. The steps involved are:

1. Parsing and translation
2. Optimization
3. Evaluation

Parsing

the parser of the query processor module checks the syntax of the query, the user's privileges to execute the query, the table names and attribute names, etc.

Parsing and Translation

As query processing includes certain activities for data retrieval. Initially, the given user queries get translated in high-level database languages such as SQL. It gets translated into expressions that can be further used at the physical level of the file system. After this, the actual evaluation of the queries and a variety of query -optimising transformations takes place. Thus before processing a query, a computer system needs to translate the query into a human-readable and understandable language.

Evaluation

- For this, with addition to the relational algebra translation, it is required to annotate the translated relational algebra expression with the instructions used for specifying and evaluating each operation. Thus, after translating the user query, the system executes a query evaluation plan.

Query Evaluation Plan

- In order to fully evaluate a query, the system needs to construct a query evaluation plan.

- The annotations in the evaluation plan may refer to the algorithms to be used for the particular index or the specific operations.
- A query execution engine is responsible for generating the output of the given query. It takes the query execution plan, executes it, and finally makes the output for the user query.

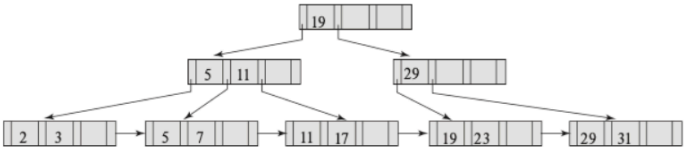
Optimization

- The cost of the query evaluation can vary for different types of queries. Although the system is responsible for constructing the evaluation plan, the user does not need to write their query efficiently.
- Usually, a database system generates an efficient query evaluation plan, which minimises its cost. This type of task is performed by the database system and is known as Query Optimization.
- For optimising a query, the query optimizer should have an estimated cost analysis of each operation. It is because the overall operation cost depends on the memory allocations to several operations, execution costs, and so on.
- Finally, after selecting an evaluation plan, the system evaluates the query and produces the output of the query.

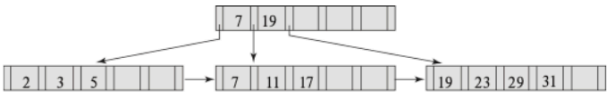
9. Construct a B+ tree for the following (2, 3, 5, 7, 11, 17, 19, 23, 29, 31). Assume that the tree is initially empty and values are added in ascending order. Construct B+ tree for the cases where the number of pointers that will fit in one node is as follows.

a. **Four**
b. **Six**
c. **Eight**

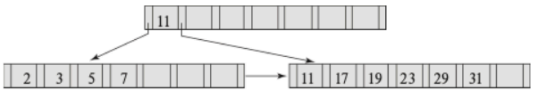
[4] $M = 4$



$M = 6$



$M = 8$



10. Consider the B+ tree index of order d = 2 shown in (fig10.1)

1. **Show the tree that would result from inserting a data entry with key 9 into this tree.**
2. **Show B+ tree that would result from inserting a data entry with key 3 into the original tree. How many page reads and page writes does the insertion require?**
3. **Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the left sibling is checked for possible distribution.**
4. **Show B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the right sibling is checked for possible redistribution.**
5. **Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 46 and then deleting the data entry with key 52.**
6. **Show the B+ tree that would result from deleting the data entry with key 91 from the original tree.**

PART B

1. Write in detail about hash based indexing and tree based indexing.

In DBMS, hashing is a technique to directly search the location of desired data on the disk without using index structure. Hashing method is used to index and retrieve items in a database as it is faster to search that specific item using the shorter hashed key instead of using its original value.

B+ tree file organisation is the advanced method of an indexed sequential access method. It uses a tree-like structure to store records in File. It uses the same concept of key-index where the primary key is used to sort the records. For each primary key, the value of the index is generated and mapped with the record.

2. Compare I/O costs for all File Organisations.

1. Sequential File Organisation
2. Heap File Organisation
3. Hash/Direct File Organisation
4. Indexed Sequential Access Method
5. B+ Tree File Organisation
6. Cluster File Organisation

- A heap file has good storage efficiency and supports fast scanning and insertion of records. However, it is slow for searches and deletions.
- A sorted file also offers good storage efficiency, but insertion and deletion of records is slow. Searches are faster than in heap files.
- A clustered file offers all the advantages of a sorted file and supports inserts and deletes efficiently. Searches are even faster than in sorted files, although a sorted file can be faster when a large number of records are retrieved sequentially, because of blocked I/O efficiencies.

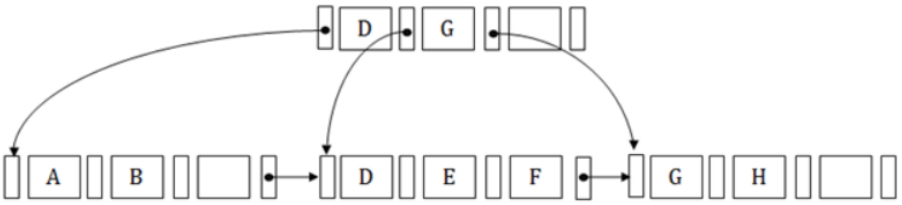
- Unclustered tree and hash indexes offer fast searches, insertion, and deletion, but scans and range searches with many matches are slow. Hash indexes are a little faster on equality searches, but they do not support range searches

3. Explain in detail about ISAM.

ISAM (Indexed Sequential Access Method) is a file management system developed at IBM which is a method for creating, maintaining, and manipulating computer files of data so that records can be retrieved sequentially or randomly by one or more keys. Indexes of key fields are maintained to achieve fast retrieval of required file records in Indexed files. IBM originally developed ISAM for mainframe computers, but implementations are available for most computer systems.

4. Explain B+ trees. Discuss about this Dynamic Index structure.

- The B+ tree is a balanced binary search tree. It follows a multi-level index format.
- In the B+ tree, leaf nodes denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height.
- In the B+ tree, the leaf nodes are linked using a link list. Therefore, a B+ tree can support random access as well as sequential access.
- In the B+ tree, every leaf node is at equal distance from the root node. The B+ tree is of the order n where n is fixed for every B+ tree.
- It contains an internal node and leaf node.



Internal node

- An internal node of the B+ tree can contain at least n/2 record pointers except the root node.
- At most, an internal node of the tree contains n pointers.

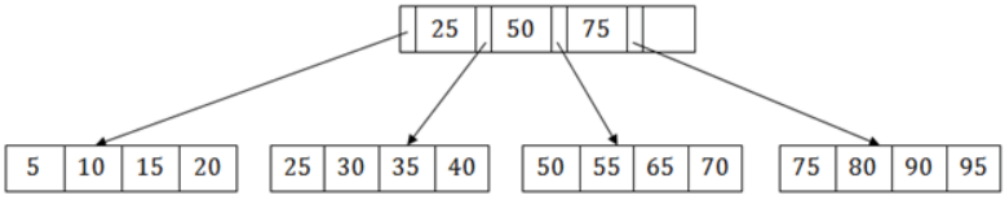
Leaf node

- The leaf node of the B+ tree can contain at least n/2 record pointers and n/2 key values.
- At most, a leaf node contains n record pointer and n key values.
- Every leaf node of the B+ tree contains one block pointer P to point to the next leaf node.

5. Demonstrate searching for a given element in B+ trees. Explain with examples.

Suppose we have to search 55 in the below B+ tree structure. First, we will fetch the intermediary node which will direct to the leaf node that can contain a record for 55.

So, in the intermediary node, we will find a branch between 50 and 75 nodes. Then at the end, we will be redirected to the third leaf node. Here DBMS will perform a sequential search to find 55.

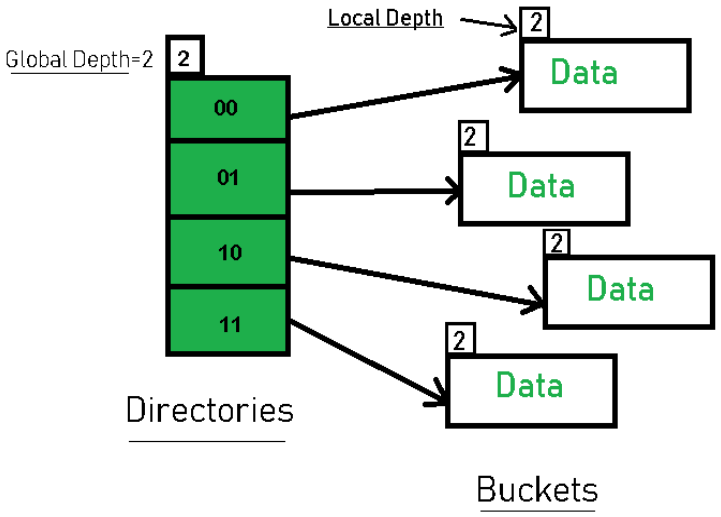


6. Compare and contrast Extendible Hashing with Linear Hashing

Extendible Hashing is a dynamic hashing method wherein directories, and buckets are used to hash data. It is an aggressively flexible method in which the hash function also experiences dynamic changes.

Main features of Extendible Hashing: The main features in this hashing technique are:

Directories: The directories store addresses of the buckets in pointers. An id is assigned to each directory which may change each time when Directory Expansion takes place.
Buckets: The buckets are used to hash the actual data.
Basic Structure of Extendible Hashing:



Extendible Hashing

- Linear hashing (LH) is a dynamic data structure which implements a hash table and grows or shrinks one bucket at a time.
- The file structure of a dynamic hashing data structure adapts itself to changes in the size of the file, so expensive periodic file reorganisation is avoided.
- A Linear Hashing file expands by splitting a predetermined bucket into two and contracts by merging two predetermined buckets into one.
- In Linear Hashing there are two types of buckets, those that are to be split and those already split. While extendible hashing splits only overflowing buckets, spiral hashing (a.k.a. spiral storage) distributes records unevenly over the buckets such that buckets with high costs of insertion, deletion, or retrieval are earliest in line for a split.

Linear Hashing vs. Extendible Hashing

- Less Code: LH
- Less Space: LH
- Higher Performance: EH potentially
- No Overflow Buckets: EH
- No Directory: LH
- Complexity of EH: B O(1), E O(1), W O(1)
- Complexity of LH: B O(1), E O(1), W O(n)

7. How does Extendible hashing use a directory of buckets? How does it handle insert and delete operations?

Extendible Hashing is a dynamic hashing method wherein directories, and buckets are used to hash data. It is an aggressively flexible method in which the hash function also experiences dynamic changes.

Main features of Extendible Hashing: The main features in this hashing technique are:

- Directories: The directories store addresses of the buckets in pointers. An id is assigned to each directory which may change each time when Directory Expansion takes place.
- Buckets: The buckets are used to hash the actual data.

Refer this link for insert and delete operations
[Extendible Hashing (Dynamic approach to DBMS) - GeeksforGeeks]

8. Explain how insert and delete operations are handled in a static hash index.

- Insert a Record

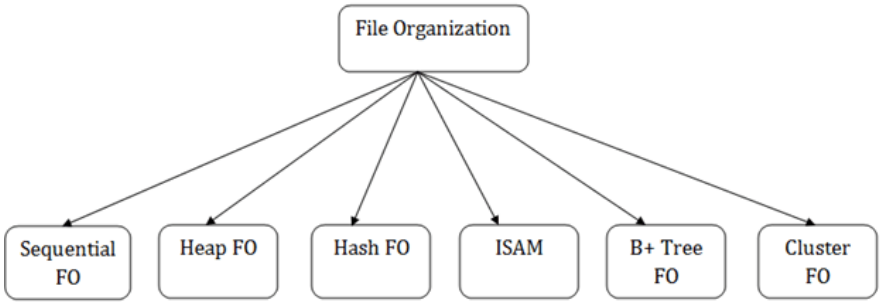
When a new record is inserted into the table, then we will generate an address for a new record based on the hash key and the record is stored in that location.

- Delete a Record

To delete a record, we will first fetch the record which is supposed to be deleted. Then we will delete the records for that address in memory.

9. Explain the organisation of records in files in detail

- The File is a collection of records. Using the primary key, we can access the records. The type and frequency of access can be determined by the type of file organisation which was used for a given set of records.
- File organisation is a logical relationship among various records. This method defines how file records are mapped onto disk blocks.
- File organisation is used to describe the way in which the records are stored in terms of blocks, and the blocks are placed on the storage medium.
- The first approach to map the database to the file is to use several files and store only one fixed length record in any given file. An alternative approach is to structure our files so that we can contain multiple lengths for records.
- Files of fixed length records are easier to implement than the files of variable length records.



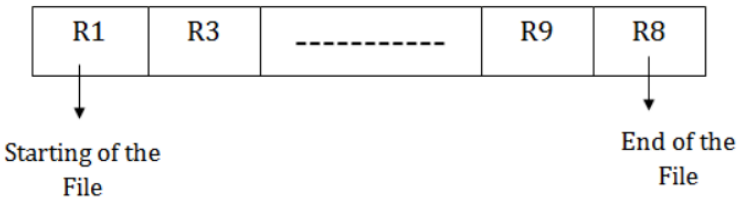
10. Explain about sequential file and heap file organisations.

Sequential

This method is the easiest method for file organisation. In this method, files are stored sequentially.This method can be implemented in two ways:

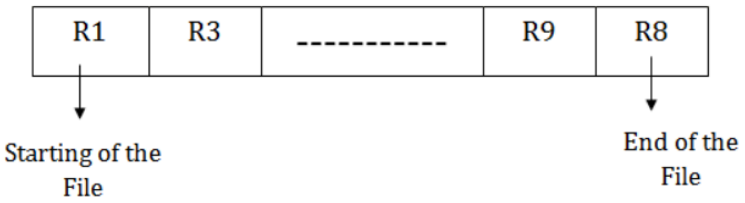
1. Pile File Method:

- It is a quite simple method. In this method, we store the record in a sequence, i.e., one after another. Here, the record will be inserted in the order in which they are inserted into tables.
- In case of updating or deleting of any record, the record will be searched in the memory blocks. When it is found, then it will be marked for deleting, and the new record is inserted.



2. Sorted File Method:

- In this method, the new record is always inserted at the file's end, and then it will sort the sequence in ascending or descending order. Sorting of records is based on any primary key or any other key.
- In the case of modification of any record, it will update the record and then sort the file, and lastly, the updated record is placed in the right place.

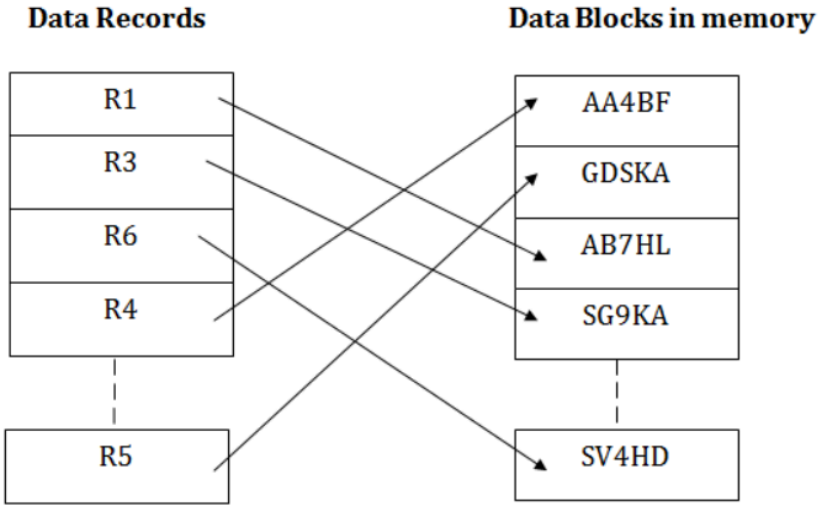


Heap File

- It is the simplest and most basic type of organisation. It works with data blocks. In heap file organisation, the records are inserted at the file's end. When the records are inserted, it doesn't require the sorting and ordering of records.
- When the data block is full, the new record is stored in some other block. This new data block need not be the very next data block, but it can select any data block in the memory to store new records. The heap file is also known as an unordered file.
- In the file, every record has a unique id, and every page in a file is of the same size. It is the DBMS responsibility to store and manage the new records.

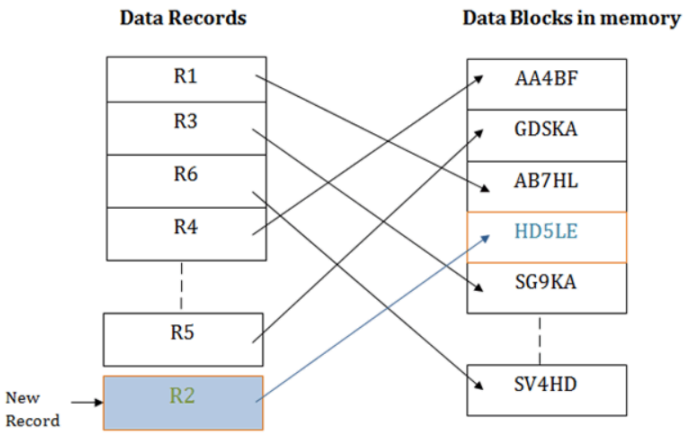
11. Explain the hash file organisation.

Hash File Organisation uses the computation of hash function on some fields of the records. The hash function's output determines the location of the disk block where the records are to be placed.



When a record has to be received using the hash key columns, then the address is generated, and the whole record is retrieved using that address. In the same way, when a new record has to be inserted, then the address is generated using the hash key and the record is directly inserted. The same process is applied in the case of delete and update.

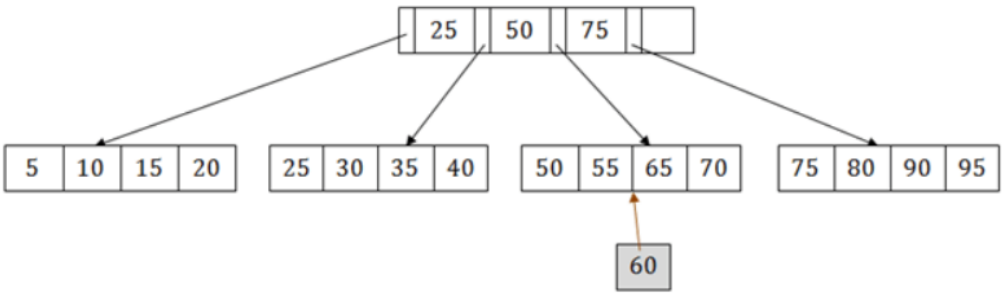
In this method, there is no effort for searching and sorting the entire file. In this method, each record will be stored randomly in the memory.



12. Illustrate insertion of an element in B+ trees with an example.

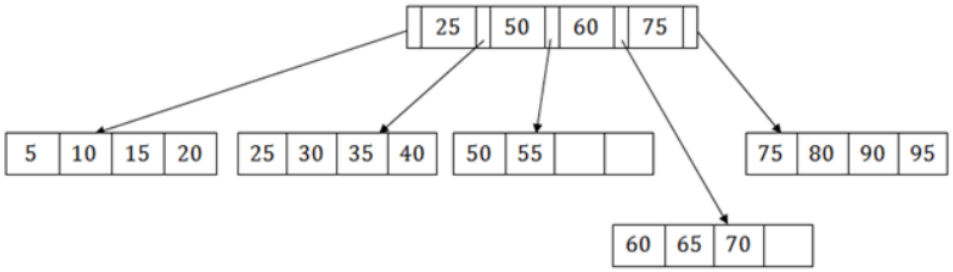
Suppose we want to insert a record 60 in the below structure. It will go to the 3rd leaf node after 55. It is a balanced tree, and a leaf node of this tree is already full, so we cannot insert 60 there.

In this case, we have to split the leaf node, so that it can be inserted into the tree without affecting the fill factor, balance and order.



The 3rd leaf node has the values (50, 55, 60, 65, 70) and its current root node is 50. We will split the leaf node of the tree in the middle so that its balance is not altered. So we can group (50, 55) and (60, 65, 70) into 2 leaf nodes.

If these two have to be leaf nodes, the intermediate node cannot branch from 50. It should have 60 added to it, and then we can have pointers to a new leaf node.

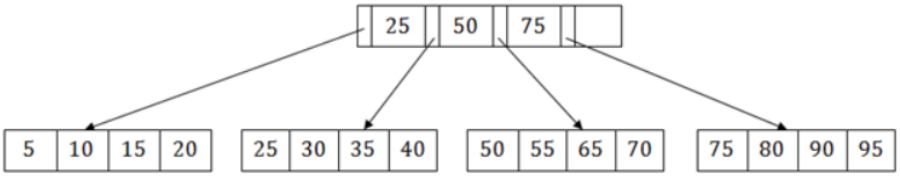


This is how we can insert an entry when there is overflow. In a normal scenario, it is very easy to find the node where it fits and then place it in that leaf node.

13. Illustrate deletion of an element in B+ trees with an example.

Suppose we want to delete 60 from the above example. In this case, we have to remove 60 from the intermediate node as well as from the 4th leaf node too. If we remove it from the intermediate node, then the tree will not satisfy the rule of the B+ tree. So we need to modify it to have a balanced tree.

After deleting node 60 from above B+ tree and re-arranging the nodes, it will show as follows:



14. Write in detail about static hashing.

Static hashing is a method of hashing, or shortening a string of characters in computer programming, in which the set of shortened characters remains the same length to improve the ease with which data can be accessed. All objects listed in an object dictionary are static and may not change when static hashing is applied. This method is often compared to the alternative, dynamic hashing.

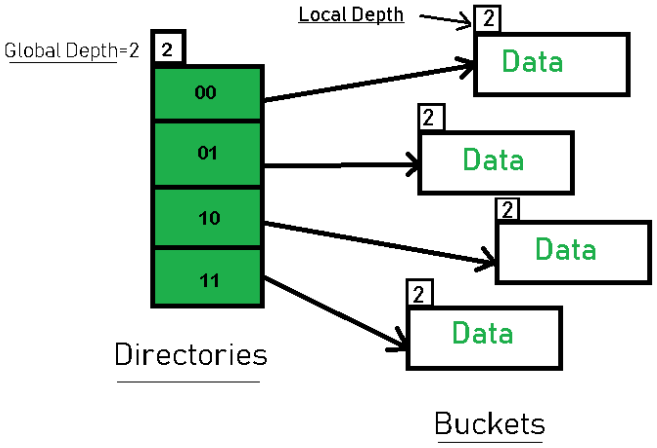
The process of static hashing creates a smaller, adaptable string of characters, making it faster and easier for users to find objects in a dictionary or groups of objects stored in a containing data structure.

15. Explain in detail about extendible hashing.

Extendible Hashing is a dynamic hashing method where directories, and buckets are used to hash data. It is an aggressively flexible method in which the hash function also experiences dynamic changes.

Main Features of Extendible hashing: The main features in this hashing techniques are:

- Directories: The directories store addresses of the buckets in pointers. An ID is assigned to each directory which may change each time when Directory Expansion takes place.
- Buckets: The buckets are used to hash the actual data.



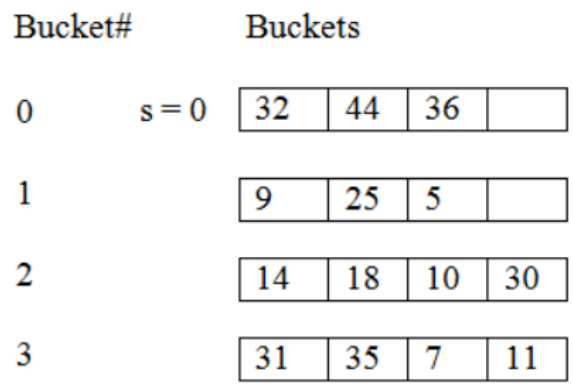
Extendible Hashing

16. Explain in detail about linear hashing.

Linear hashing is a dynamic hashing technique. It allows a file to extend or shrink its number of buckets without a directory as used in Extendible Hashing. Suppose you start with a number of buckets N to put records in the buckets 0 to N-1. Let this be round i which will be 0 initially. You start with an initial mod

hash function hi(K) = K mod 2iN. When there is a collision, the first bucket, i.e., bucket 0, is split into two buckets: bucket 0 and a new bucket N at the end of the file. The records in bucket 0 are redistributed between the two buckets using another hash function hi+1 = K mod 2(i+1)N. Important property of the hash function is that records hashed into bucket 0 based on hi will be hashed to bucket 0 or bucket N based on the new hash function hi+1.

Here is a simple example of using linear hashing to store 14 records with number of initial buckets N = 4.



17. Explain about storage devices and memory hierarchy.

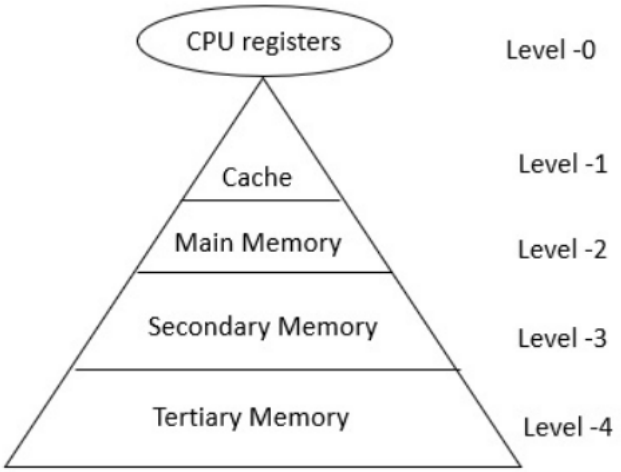
A storage device is a kind of hardware, which is also known as storage, storage medium, digital storage, or storage media that has the ability to store information either temporarily or permanently. Generally, it is used to hold, port, and extract data files.

Storage devices are available in various form factors; for example, a computer device includes different storage media such as hard disk, RAM, cache. They also have optical disk drives and externally connected USB drives.

Two types of storage devices, primary and secondary are available there to store data:

- Primary storage devices: They are fit internally to the computer and very fast in terms of accessing data files. The RAM and cache memory are the examples of the primary storage devices.
- Secondary storage devices: The hard disk, USB storage devices and optical disk drive are examples of secondary storage devices, which are designed to store data permanently. They include a large storage capacity while comparing with primary storage devices.

Memory hierarchy is arranging different kinds of storage present on a computing device based on speed of access. At the very top, the highest performing storage is CPU registers which are the fastest to read and write to.



18. Explain different RAID levels in disks.

RAID stands for a Redundant Array of Independent Disks. The technology combines two or more physical drives into a logical unit presented as a single hard drive to the operating system.

RAID levels are defined by the combination of the techniques used; they also provide varying degrees of reliability (ability to withstand drive failure) and availability (speed of I/O). There are six basic RAID levels:

- RAID Level 0 stripes data across two or more drives. No parity.
- RAID Level 1 mirrors data to two or more drives. No parity.
- RAID Level 0+1 is striped sets in a mirrored set. RAID 0+1 creates a striped set that mirrors a primary striped set.
- RAID Level 1+0 (RAID 10) is mirrored in a striped set. RAID 0+1 creates a striped set from a series of mirrored drives.
- RAID Level 3 is byte-level striping with a dedicated parity disk.
- RAID Level 4 is block-level striping with a dedicated parity disk.
- RAID Level 5 is striping with distributed (interleaved) parity. No dedicated parity disk.

19. Compare Tree indices and hash indices.

B-Tree Index Characteristics

A B-tree index can be used for column comparisons in expressions that use the =, >, >=, <, <=, or BETWEEN operators. The index also can be used for LIKE comparisons if the argument to LIKE is a constant string that does not start with a wildcard character. For example, the following SELECT statements use indexes:

```
SELECT * FROM tbl_name WHERE key_col LIKE 'Patrick%';
SELECT * FROM tbl_name WHERE key_col LIKE 'Pat%_ck%';
```

Hash Index Characteristics

Hash indexes have somewhat different characteristics from those just discussed:

They are used only for equality comparisons that use the = or ⇔ operators (but are very fast). They are not used for comparison operators such as < that find a range of values. Systems that rely on this type of single-value lookup are

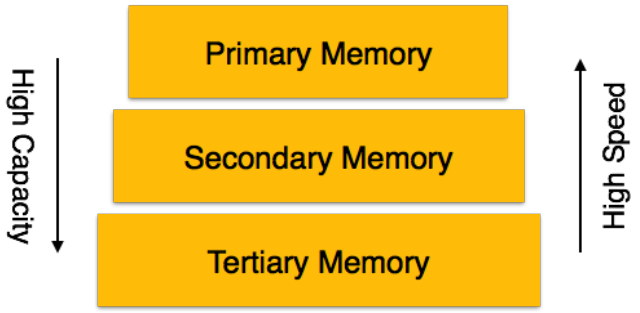
known as “key-value stores”; to use MySQL for such applications, use hash indexes wherever possible.
The optimizer cannot use a hash index to speed up ORDER BY operations. (This type of index cannot be used to search for the next entry in order.)

20. Explain the steps in query processing and write about measures of query cost.

PART C

1. Write about data on external storage.

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 or motherboard, for example, magnetic disks, optical disks (DVD, CD, etc.), hard disks, flash drives, and magnetic tapes.



2. Illustrate Clustered Indexes.

- 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 such a situation, you can group two or more columns to get the unique values and create an index which is called clustered Index. This also helps you to identify the record faster.

Example:

- Let's assume that a company recruited many employees in various departments. In this case, clustering indexing in DBMS should be created for all employees who belong to the same dept.
- It is considered in a single cluster, and index points point to the cluster as a whole. Here, Department _no is a non-unique key.

3. Discuss the primary and secondary indexes.

Indexing in Database is defined based on its indexing attributes. Two main types of indexing methods are:

- Primary Indexing
- Secondary Indexing

Primary Index in DBMS

Primary Index is an ordered file which is fixed length with two fields. The first field is the same as a primary key and the second field is pointed to that specific data block. In the primary Index, there is always one to one relationship between the entries in the index table.

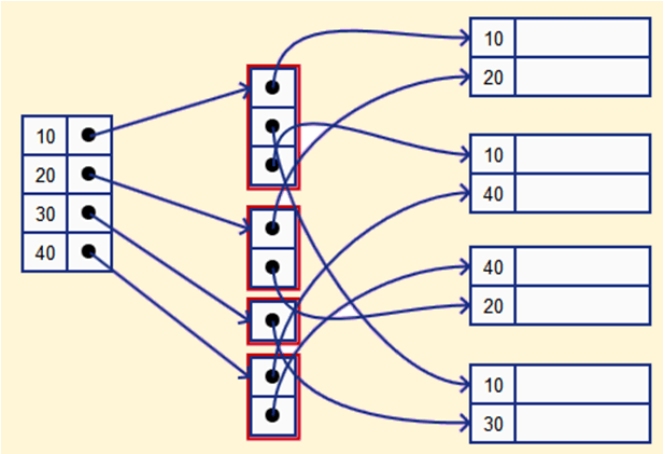
The primary Indexing in DBMS is also further divided into two types.

- Dense Index
- Sparse Index

Secondary Index in DBMS

The secondary Index in DBMS can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a 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.



Refer this link: [Indexing in DBMS: What is, Types of Indexes with EXAMPLES]

4. Define Tree hashing.

5. Describe Storage Hierarchy.

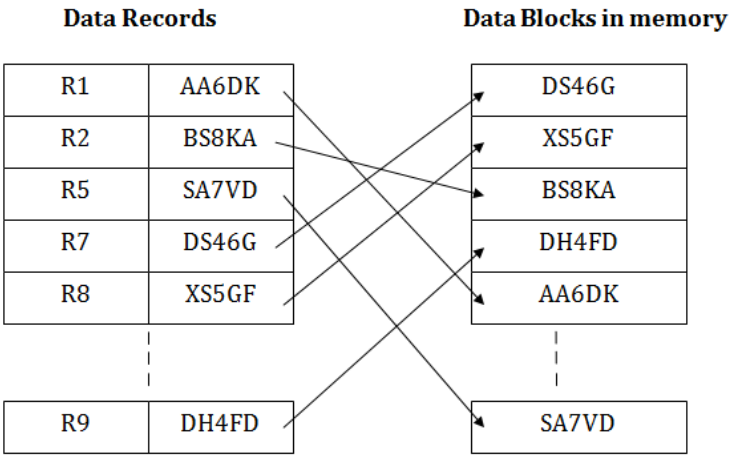
Refer question no:17(part-B)

6. Discuss the intuition for tree indexes.

7. Define Indexed Sequential Access Method.

Indexed sequential access method (ISAM):

ISAM method is an advanced sequential file organisation. In this method, records are stored in the file using the primary key. An index value is generated for each primary key and mapped with the record. This index contains the address of the record in the file.



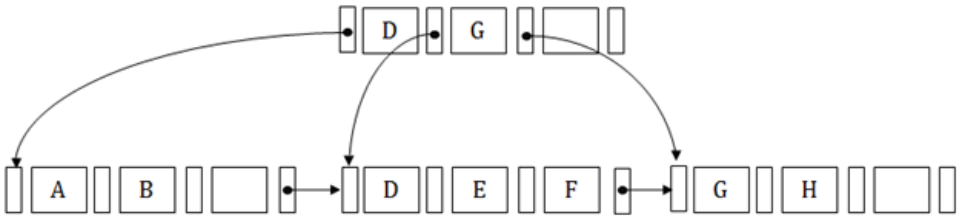
➔ If any record has to be retrieved based on its index value, then the address of the data block is fetched and the record is retrieved from the memory.

8. Discuss about Overflow pages and locking considerations of ISAM.

9. Describe structure of B+ tree nodes.

Structure of B+ Tree

- In the B+ tree, every leaf node is at equal distance from the root node. The B+ tree is of the order n where n is fixed for every B+ tree.
- It contains an internal node and leaf node.



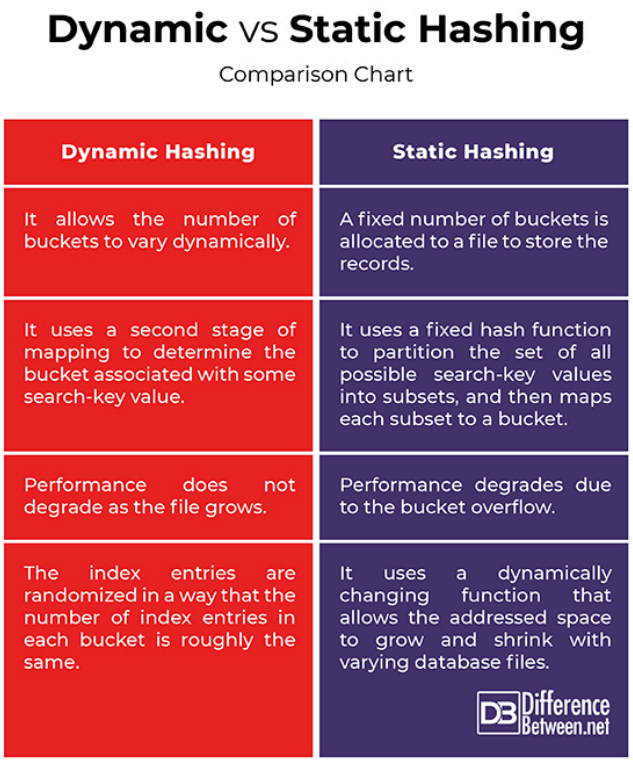
Internal node

- An internal node of the B+ tree can contain at least n/2 record pointers except the root node.
- At most, an internal node of the tree contains n pointers.

Leaf node

- The leaf node of the B+ tree can contain at least n/2 record pointers and n/2 key values.
- At most, a leaf node contains n record pointer and n key values.
- Every leaf node of the B+ tree contains one block pointer P to point to next leaf node.

10. Compare dynamic and static hash techniques.



Refer: [Difference Between Dynamic and Static Hashing]

11. What is a timestamp?

- Timestamp is a unique identifier created by the DBMS to identify the relative starting time of a transaction.
- Typically, timestamp values are assigned in the order in which the transactions are submitted to the system.

- So, a timestamp can be thought of as the transaction start time. Therefore, time stamping is a method of concurrency control in which each transaction is assigned a transaction timestamp.

12. List the different file organisations.

➔ File Organisation refers to the logical relationships among various records that constitute the file, particularly with respect to the means of identification and access to any specific record.

➔ In simple terms, Storing the files in a certain order is called file Organisation.

Types of File Organisations –

Various methods have been introduced to Organise files.

- Sequential File Organisation
- Heap File Organisation
- Hash File Organisation
- B+ Tree File Organisation
- Clustered File Organisation

13. What is log?

- Log is nothing but a file which contains a sequence of records, each log record refers to a write operation. All the log records are recorded step by step in the log file. We can say, log files store the history of all updates activities.
- Log contains start of transaction, transaction number, record number, old value, new value, end of transaction etc. For example, mini statements in bank ATMs.

- If within an ongoing transaction, the system crashes, then by using log files, we can return back to the previous state as if nothing has happened to the database.
- The log is kept on disk so that it is not affected by failures except disk and failures.

14. List the steps in Query processing.

Query Processing is the activity performed in extracting data from the database. In query processing, it takes various steps for fetching the data from the database. The steps involved are:

1. Parsing and translation
2. Optimization
3. Evaluation

The query processing works in the following way:

Parsing

the parser of the query processor module checks the syntax of the query, the user's privileges to execute the query, the table names and attribute names, etc.

Translation

If we have written a valid query, then it is converted from high level language SQL to low level instruction in Relational Algebra.

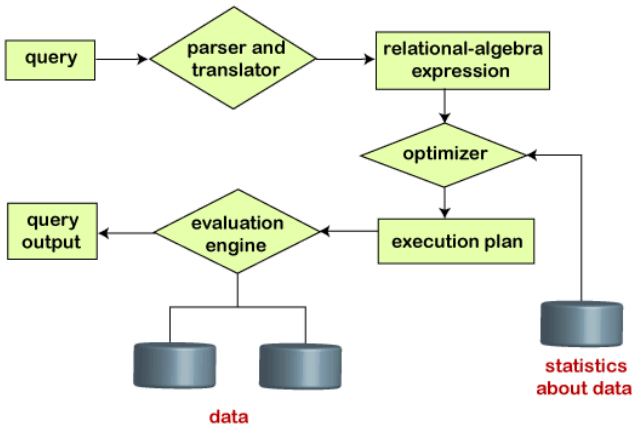
Optimizer

Optimizer uses the statistical data stored as part of a data dictionary. The statistical data are information about the size of the table, the length of

records, the indexes created on the table, etc. Optimizer also checks for the conditions and conditional attributes which are parts of the query.

Evaluation:

At this stage, we choose one execution plan of the several we have developed. This Execution plan accesses data from the database to give the final result.



Steps in query processing

15. What is blind writing?

Blind write is simply when a transaction writes without reading. i.e a transaction has WRITE(Q), but no READ(Q) before it. So, the transaction is writing to the database "blindly" without reading the previous value.

If there is no read that happens prior to the first write then it is said to be a *blind write*.

T1	T2	T3
	$R_2(X)$	
$R_1(X)$		
		$W_3(X)$
	$W_2(X)$	

- W3(X) is a blind write, as there is no read before write [R3(X) before W3(X)]
- W2(X) is not a blind write, as a read happens before write [R2(X) before W2(X)]

16. Describe immediate database modification.

The immediate database modification technique allows database modification to be output to the database while the transaction is still in the active state. The data modification written by active transactions are called **"uncommitted modification"**.
If the system crashes or transaction aborts, then the old value field of the log records is used to restore the modified data items to the value they had prior to the start of the transaction. This restoration is accomplished through the undo operation.
Refer: What is Immediate database modification? | Practice | GeeksforGeeks

17. Discuss the advantages of heap file organisation.

Heap File Organisation

This is the simplest form of file organisation. Here records are inserted at the end of the file as and when they are inserted. There is no sorting or ordering of the records. Once the data block is full, the next record is stored in the new block.

Advantages of Heap file organisation:

- Very good method of file organisation for bulk insertion. i.e.; when there is a huge amount of data needed to load into the database at a time, then this method of file organisation is best suited. They are simply inserted one after the other in the memory blocks.

- It is suited for very small files as the fetching of records is faster in them. As the file size grows, linear search for the record becomes time consuming.

18. What is transaction failure?

The transaction failure occurs when it fails to execute or when it reaches a point from where it can't go any further. If a few transactions or processes are hurt, then this is called a transaction failure.
Reasons for a transaction failure could be -

1. Logical errors: If a transaction cannot complete due to some code error or an internal error condition, then the logical error occurs.
2. Syntax error: It occurs where the DBMS itself terminates an active transaction because the database system is not able to execute it. For example, The system aborts an active transaction, in case of deadlock or resource unavailability.

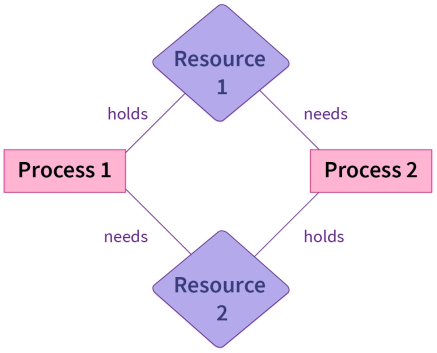
19. Identify when a transaction system is in a deadlock state.

- In a database, a deadlock is an unwanted situation in which two or more transactions are waiting indefinitely for one another to give up locks.
- Deadlock is said to be one of the most feared complications in DBMS as it brings the whole system to a Halt.

Necessary Conditions for Deadlock:
The four necessary conditions for a deadlock to arise are as follows.

- Mutual Exclusion: Only one process can use a resource at any given time i.e. the resources are non-sharable.

- Hold and wait: A process is holding at least one resource at a time and is waiting to acquire other resources held by some other process.
- No preemption: The resource can be released by a process voluntarily i.e. after execution of the process.
- Circular Wait: A set of processes are waiting for each other in a circular fashion.



20. What is the locking protocol?

In this type of protocol, any transaction cannot read or write data until it acquires an appropriate lock on it. There are two types of lock:

1. Shared lock:

- It is also known as a Read-only lock. In a shared lock, the data item can only be read by the transaction.
- It can be shared between the transactions because when the transaction holds a lock, then it can't update the data on the data item.

2. Exclusive lock:

- In the exclusive lock, the data item can be both read and written by the transaction.
- This lock is exclusive, and in this lock, multiple transactions do not modify the same data simultaneously.