Differentiate between dense and sparse index files.

Dense index	Sparse index
Larger index size.	Smaller index size
Less time to locate data.	More time to locate data
More overhead for insertions and deletions.	Less overhead for insertions and deletions.
Records need not to be clustered.	Records need to be clustered.
Less computing time in RAM	More computing time in RAM
Data pointers point to each record in the data file.	Data pointers point to fewer records in data file.
Faster search performance	Slower search performance

What do you understand by data mining?

Data Mining is the process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data. Data mining is also called Knowledge Discovery in Database. It is used in healthcare, fraud detection, lie detection, education, banking, etc.

What are checkpoints?

Checkpoint is used to declare a point before which the DBMS was in a consistent state, and all transactions were committed. There are two types of Checkpoints:

1. **Automatic Checkpoint:** They are updated more frequently and are used for larger databases.
2. **Manual Checkpoint:** They are updated less frequently and are used for smaller databases.

Differentiate between shared disk and shared memory parallel database architectures.

Feature	Shared Disk Architecture	Shared Memory Architecture
Memory	Each processor has its own private memory	All processors share a common memory
Interconnect	High-speed network connecting processors to shared disk	High-speed bus connecting processors to shared memory

Feature	Shared Disk Architecture	Shared Memory Architecture
Scalability	Good scalability	Limited scalability
Data Consistency	Requires distributed locking	Requires memory locking
Fault Tolerance	Better fault tolerance	Less fault tolerance
Performance Bottleneck	Interconnect and shared disk become bottlenecks	Memory bus become a bottleneck
Example	RAC	SMP

List the different types of failures in database systems.

1. Transaction Failure: If a transaction is not able to execute then it is called transaction failure.

Reason for a transaction failure in DBMS:

- **Logical error:** Mistakes in the code or internal faults.
- **System error:** Termination of an active transaction by the database itself due to system issues.

2. System Crash: A system crash is caused by hardware or software breakdown. They are also called soft failures and are responsible for the data losses in the volatile memory.

3. Data-transfer Failure: It is caused by disk failure and results in loss of content from disk storage. Backup copy of the data should be made to prevent data loss.

What is the need of normalization?

1. It is used to remove the duplicate data and database anomalies from the relation.
2. It helps to reduce redundancy and complexity.
3. It divides the large database table into smaller tables and link them using relationship.
4. It avoids duplicate data in a table.
5. It reduces the chances of anomalies in a database.

What do you understand by Triggers. Differentiate between Row-Level and Statement-level triggers.

Trigger is defined as stored program that is automatically executed whenever some event takes place. Trigger is of two types: **Row Level** and **Statement Level** trigger.

Row Level Triggers	Statement Level Triggers
Executes once for each row in a transaction.	Executes once for each transaction.
Used for data auditing purpose.	Used for enforcing additional security on the transactions.
“FOR EACH ROW” clause is present in CREATE TRIGGER command.	“FOR EACH STATEMENT” clause is present in CREATE TRIGGER command.
Example: If 1500 rows are to be inserted into a table, the row level trigger would execute 1500 times.	Example: If 1500 rows are to be inserted into a table, the statement level trigger would execute only once.

Lossy and Lossless decomposition

Lossless Decomposition	Lossy Decomposition
The decompositions R1, R2, R2…Rn for a relation schema R are said to be Lossless if there natural join results the original relation R.	The decompositions R1, R2, R2…Rn for a relation schema R are said to be Lossy if there natural join results into addition of extraneous tuples with the original relation R.
There is no loss of information so it is also called non-additive join decomposition	There is loss of information so it is also called careless decomposition.
The common attribute of the sub relations is a superkey of any one of the relation.	The common attribute of the sub relation is not a superkey of any of the sub relation.

Aggregation and ternary relationship

Aggregation	Ternary Relationship
Abstraction to represent relationships as higher-level entity sets	Relationship between three entities.
Simplifies complex structures into higher-level entities	Models interdependencies among three entities

Used when one entity is composed of multiple entities	Used when a relationship inherently involves three entities
Represented using rectangle enclosing related entities and their relationships	Represented using diamond connected to three entities

Triggers and Assertions

Assertion	**Triggers**
Does not maintain track of changes made in table.	Maintains track of changes made in table.
Smaller syntax	Larger syntax
Not used by modern databases	Used by modern databases
Used to enforces business rules and constraints.	Used to execute actions in response to data changes.
Activation is checked after a transaction completes	Activation is checked during a transaction
Granularity applies to the entire database	Granularity applies to a specific table or view
Uses SQL statements Easy to debug	Uses procedural code Difficult to debug
Example: CHECK constraints, FOREIGN KEY constraints	Example: AFTER INSERT triggers, INSTEAD OF triggers

What is a minimal set of functional dependencies? Does every set of dependencies have a minimal equivalent set? Is it always unique?

A set of functional dependencies is called minimal set if each FD in it is a −

- Simple FD.
- Left reduced FD.
- Non-redundant FD.

It is also called Canonical cover or Minimal cover or Irreducible Set

Steps to find minimal set:

Step 1: First split the all left-hand attributes of all functional dependencies.

Step 2: Now remove all redundant functional dependencies.

Step 3: Find the Extraneous attribute and remove it.

Not every set of dependencies has a minimal equivalent set. Finding the minimal set of dependencies is not always unique. There can be multiple minimal sets of dependencies that are equivalent in terms of determining the same set of attributes.

Explain the issues that are to be addressed for a distributed database design.

1. **Heterogeneity**: Heterogeneity refers to the differences in network types, hardware, operating systems, and software within a distributed system. Middleware helps to bridge these differences.
2. **Openness**: Openness determines how easily new resource-sharing services can be integrated into the system and interact with other systems.
4. **Scalability**: The scalability of the system should remain efficient even with a significant increase in the number of users and resources connected.
5. **Security**: Encryption protects shared resources and keeps sensitive information secrets when transmitted.
5. **Failure Handling**: When some faults occur in hardware or software program, it causes failure.
6. **Concurrency**: Shared resource in a distributed system must operate correctly in a concurrent environment.
7. **Transparency**: The user should be unaware of where the services are located and the transfer from a local machine to a remote one should be transparent.

What do you understand by a data warehouse? Discuss the multi-tier architecture of a data warehouse.

The data warehouse is an integrated, subject-oriented, time-variant, and non-volatile collection of data.

Multi-tier Architecture:

1. **Bottom Tier:** It consists of the **Data Warehouse server**, which is an RDBMS. Data from operational databases and external sources are extracted using application program interfaces called a gateway. Examples of gateways are **ODBC** and **JDBC**.
2. **Middle Tier:** It consists of an **OLAP server** for fast querying of the data warehouse. The OLAP server is implemented using either

    - **Relational OLAP (ROLAP) model**, i.e., an extended relational DBMS that maps functions on multidimensional data to standard relational operations.
    - **Multidimensional OLAP (MOLAP) model**, i.e., a server that directly implements multidimensional information and operations.

3. **Top Tier:** It consists of **front-end tools** for displaying results provided by OLAP, as well as additional tools for data mining of the OLAP-generated data.

Discuss the various types of updates possible in a data warehouse with the help of an example.

- **Full Refresh:** Replacing entire dataset in the data warehouse with the latest data from the source systems.
- **Incremental Load:** Adding only the new data since the last update to the existing dataset.
- **Historical Correction:** Updating historical data in the data warehouse to correct errors.
- **Type 1 Dimension Updates:** Overwriting existing dimensional data with the latest information, without preserving historical changes.
- **Type 2 Dimension Updates:** Capturing changes to dimensional data over time while preserving historical information.
- **Aggregate Updates:** Modifying pre-calculated aggregate data in the data warehouse.

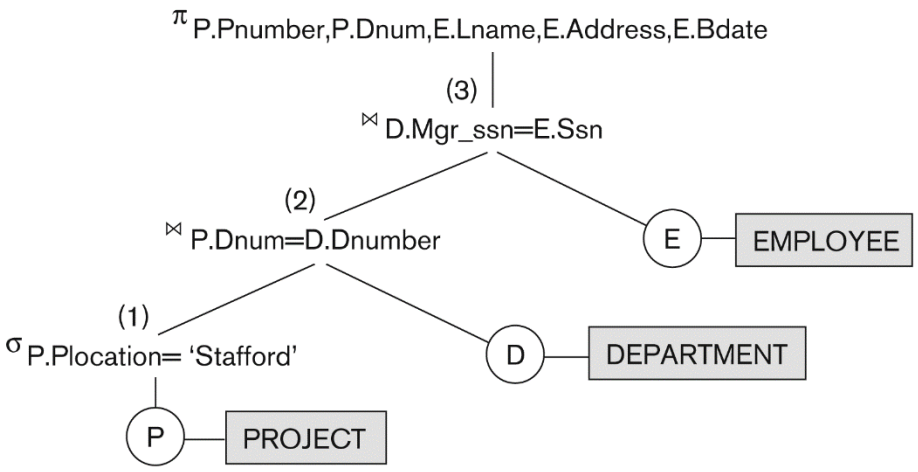
Discuss the typical phases of query processing with the help of a diagram.

Phases of query processing:

1. **Parsing and translation:** When a user executes any query, the parser builds a parse-tree and translate it into relational algebra. Consider a query:

```
SELECT    P.NUMBER,P.DNUM,E.LNAME,
          E.ADDRESS, E.BDATE
FROM      PROJECT AS P,DEPARTMENT AS D,
                EMPLOYEE AS E
WHERE     P.DNUM=D.DNUMBER AND
          D.MGRSSN=E.SSN AND
          P.PLOCATION='STAFFORD';
```

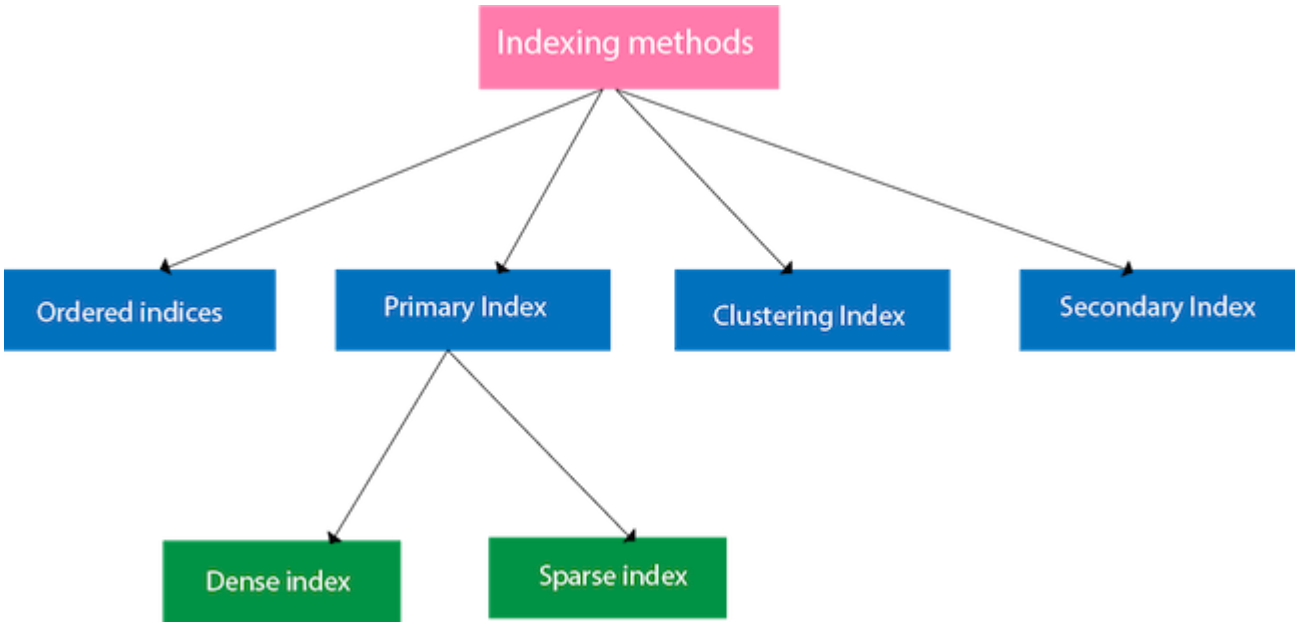
(a)



2. **Optimization:** Optimization process follows some factors like indexing, joins, etc. These factors help in determining the most efficient query execution plan. The main goal of this step is to retrieve the required data with minimal cost in terms of resources and time.
3. **Evaluation:** After finding the best execution plan, the DBMS starts the execution of the optimized query. In this step, DBMS performs operations on the data like selection, insertion, update, and so on.

Indexing

Indexing is used to optimize the performance of a database by minimizing the number of disk accesses.



- **Ordered indices:** Sorted indices are known as ordered indices.
- **Primary index:** Index created on the basis of the primary key of the table is called primary index.
- **Dense Index:** It contains an index record for every search key value in the data file.
- **Sparse Index:** Index points to the records in the main table in a gap.
- **Cluster Index:** We group two or more columns to get the unique value and create index out of them
- **Secondary Index:** Huge range for the columns is selected then each range is further divided into smaller ranges. The mapping of the first level is stored in the primary memory. The mapping of the second level and actual data are stored in the secondary memory.

Extended ER diagram

Extended ER diagram is a diagrammatic technique for displaying the Subclass and Superclass; Specialization and Generalization; Union and Aggregation etc.

Specialization is the process of defining subclasses from a superclass, while generalization is the process of defining a superclass from two or more subclasses. Union is a combination of two or more entities. Aggregation groups multiple entities into a single entity. Inheritance allows subtypes to inherit attributes and relationships from their supertype.