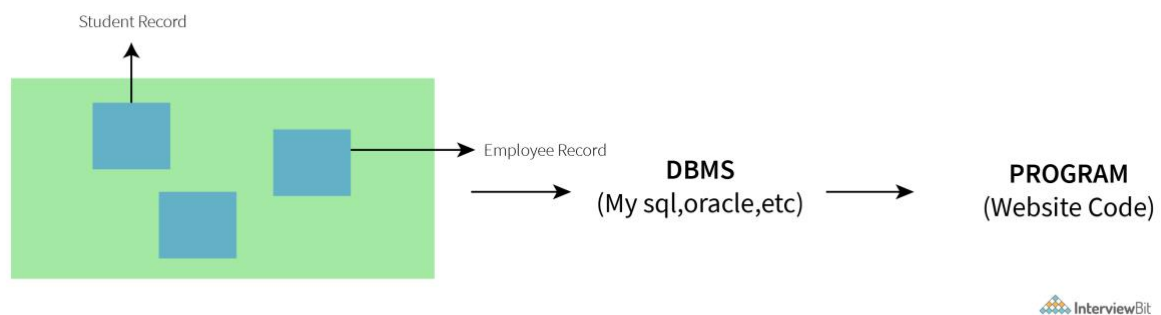


INTERVIEW QUESTIONS ON DBMS

1. What is meant by DBMS and what is its utility? Explain RDBMS with examples.

As the name suggests DBMS or Database Management System is a set of applications or programs that enable users to create and maintain a database. DBMS provides a tool or an interface for performing various operations such as inserting, deleting, updating, etc. into a database. It is software that enables the storage of data more compactly and securely as compared to a file-based system. A DBMS system helps a user to overcome problems like data inconsistency, data redundancy, etc. in a database and makes it more convenient and organized to use it.

Examples of popular DBMS systems are file systems, XML, Windows Registry, etc.



RDBMS stands for Relational Database Management System and was introduced in the 1970s to access and store data more efficiently than DBMS. RDBMS stores data in the form of tables as compared to DBMS which stores data as files. Storing data as rows and columns makes it easier to locate specific values in the database and makes it more efficient as compared to DBMS.

Examples of popular RDBMS systems are MySQL, Oracle DB, etc.

2. What is meant by a database?

A Database is an organized, consistent, and logical collection of data that can easily be updated, accessed, and managed. Database mostly contains sets of tables or objects (anything created using create command is a database object) which consist of records and fields. A tuple or a row represents a single entry in a table. An attribute or a column represents the basic units of data storage, which contain information about a particular aspect of the table. DBMS extracts data from a database in the form of queries given by the user.

3. Mention the issues with traditional file-based systems that make DBMS a better choice?

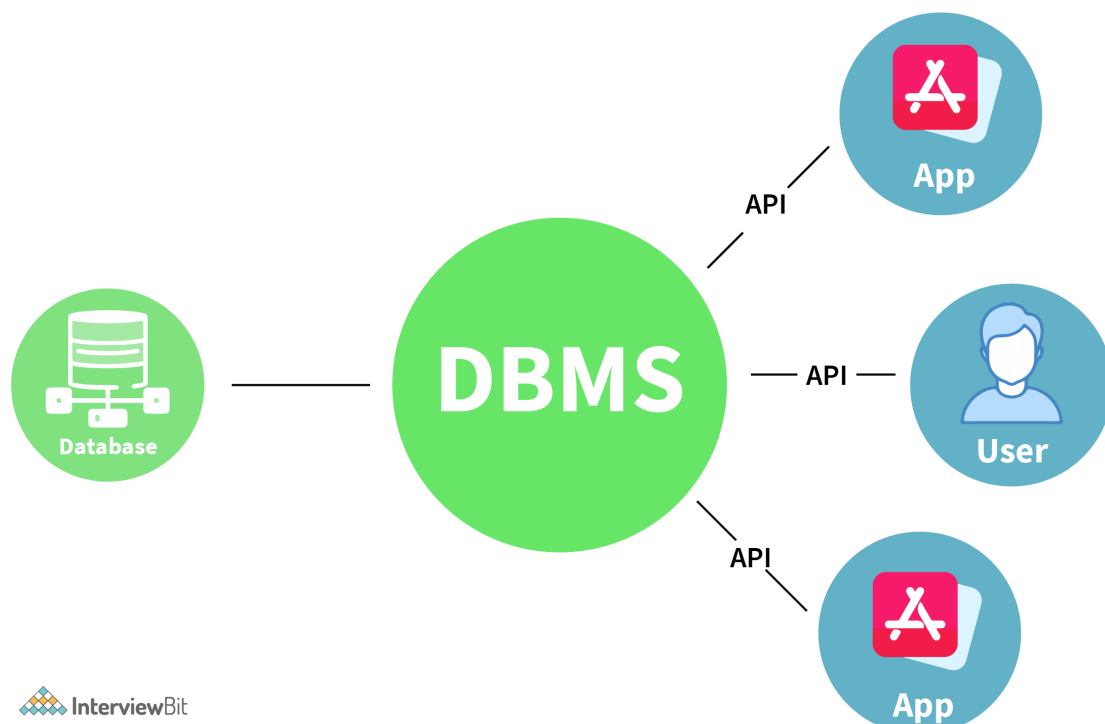
The absence of indexing in a traditional file-based system leaves us with the only option of scanning the full page and hence making the access of content tedious and super slow. The other issue is redundancy and inconsistency as files have many duplicate and redundant data and changing one of them makes all of them inconsistent. Accessing data is harder in traditional file-based systems because data is unorganized in them.

Another issue is the lack of concurrency control, which leads to one operation locking the entire page, as compared to DBMS where multiple operations can work on a single file simultaneously.

Integrity check, data isolation, atomicity, security, etc. are some other issues with traditional file-based systems for which DBMSs have provided some good solutions.

4. Explain a few advantages of a DBMS.

Following are the few advantages of using a DBMS.



- **Data Sharing:** Data from a single database can be simultaneously shared by multiple users. Such sharing also enables end-users to react to changes quickly in the database environment.

- **Integrity constraints:** The existence of such constraints allows storing of data in an organized and refined manner.
- **Controlling redundancy in a database:** Eliminates redundancy in a database by providing a mechanism that integrates all the data in a single database.
- **Data Independence:** This allows changing the data structure without altering the composition of any of the executing application programs.
- **Provides backup and recovery facility:** It can be configured to automatically create the backup of the data and restore the data in the database whenever required.
- **Data Security:** DBMS provides the necessary tools to make the storage and transfer of data more reliable and secure. Authentication (the process of giving restricted access to a user) and encryption (encrypting sensitive data such as OTP, credit card information, etc.) are some popular tools used to secure data in a DBMS.

5. Explain different languages present in DBMS.

Following are various languages present in DBMS:

- **DDL(Data Definition Language):** It contains commands which are required to define the database.
E.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc.
- **DML(Data Manipulation Language):** It contains commands which are required to manipulate the data present in the database.
E.g., SELECT, UPDATE, INSERT, DELETE, etc.
- **DCL(Data Control Language):** It contains commands which are required to deal with the user permissions and controls of the database system.
E.g., GRANT and REVOKE.
- **TCL(Transaction Control Language):** It contains commands which are required to deal with the transaction of the database.
E.g., COMMIT, ROLLBACK, and SAVEPOINT.

6. What is meant by ACID properties in DBMS?

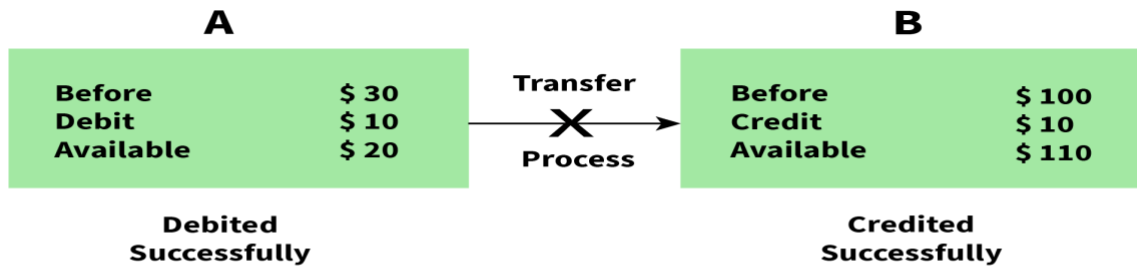
ACID stands for Atomicity, Consistency, Isolation, and Durability in a DBMS these are those properties that ensure a safe and secure way of sharing data among multiple users.

DATABASE TRANSACTIONS

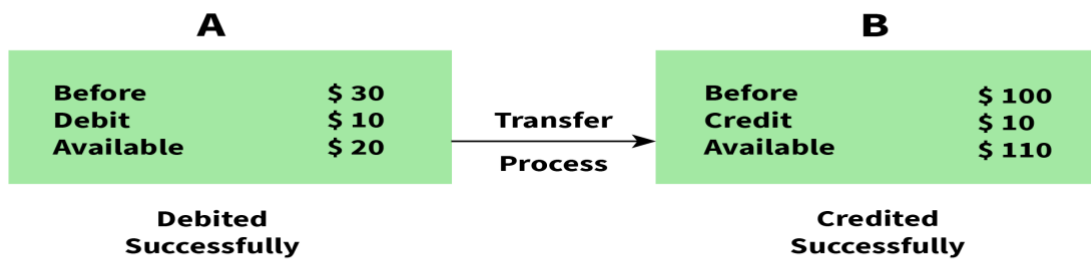
- A Atomic**
All changes to the data must be performed successfully or not at all
- C Consistent**
Data must be in a consistent state before and after the transaction
- I Isolated**
No other process can change the data while the transaction is running
- D Durable**
The changes made by a transaction must persist



- **Atomicity:** This property reflects the concept of either executing the whole query or executing nothing at all, which implies that if an update occurs in a database then that update should either be reflected in the whole database or should not be reflected at all.



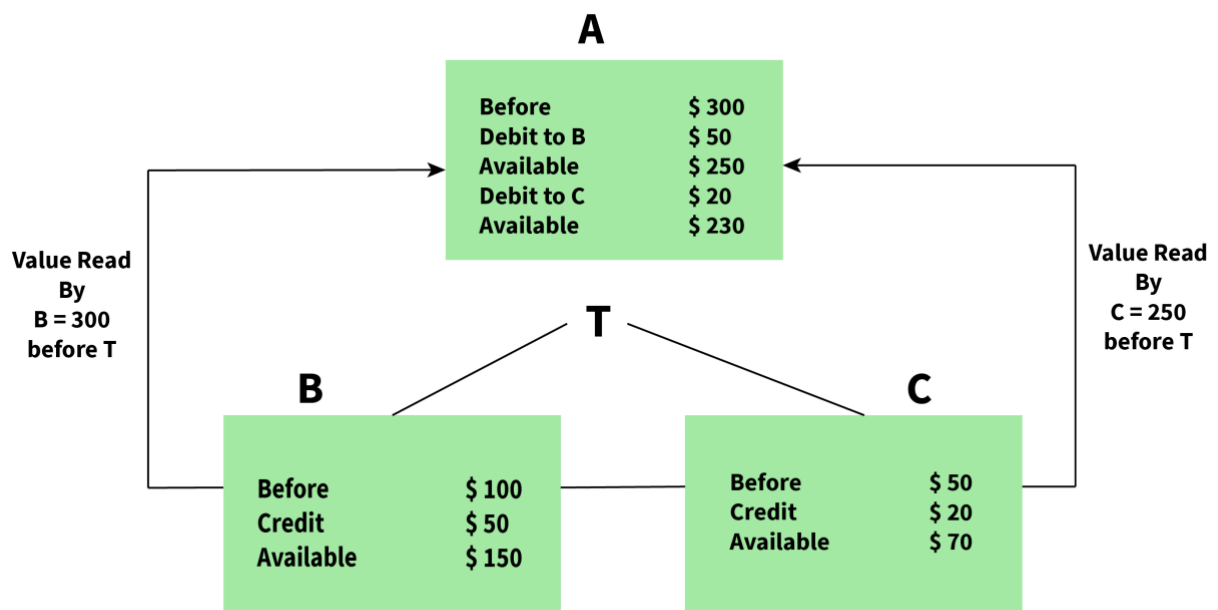
Partial Execution
No Atomicity
Execution Termination



Complete Execution
Atomicity
Execution Successfull



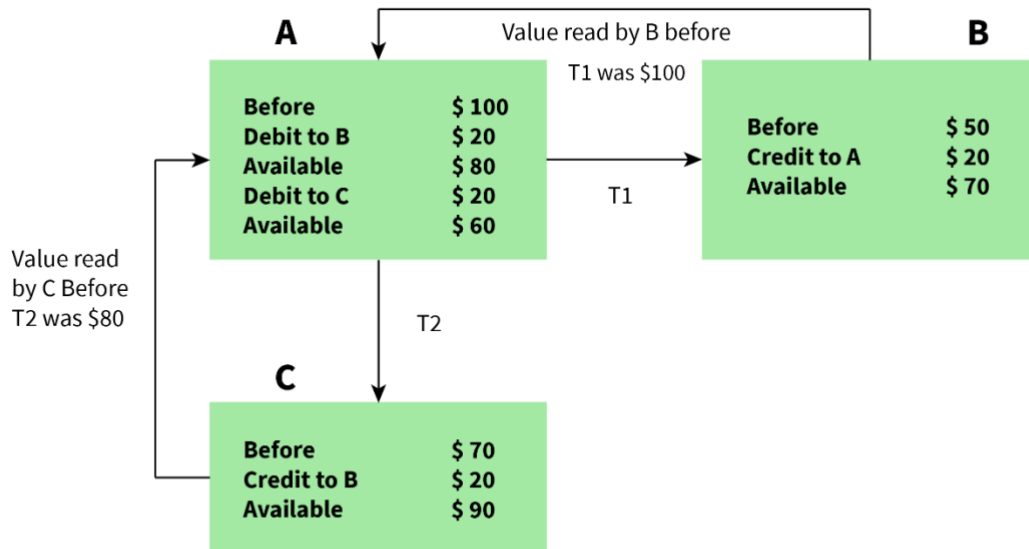
- **Consistency:** This property ensures that the data remains consistent before and after a transaction in a database.



Data Consistent



- **Isolation:** This property ensures that each transaction is occurring independently of the others. This implies that the state of an ongoing transaction doesn't affect the state of another ongoing transaction.



Isolation - Independent execution T1 and T2 by A



- **Durability:** This property ensures that the data is not lost in cases of a system failure or restart and is present in the same state as it was before the system failure or restart.

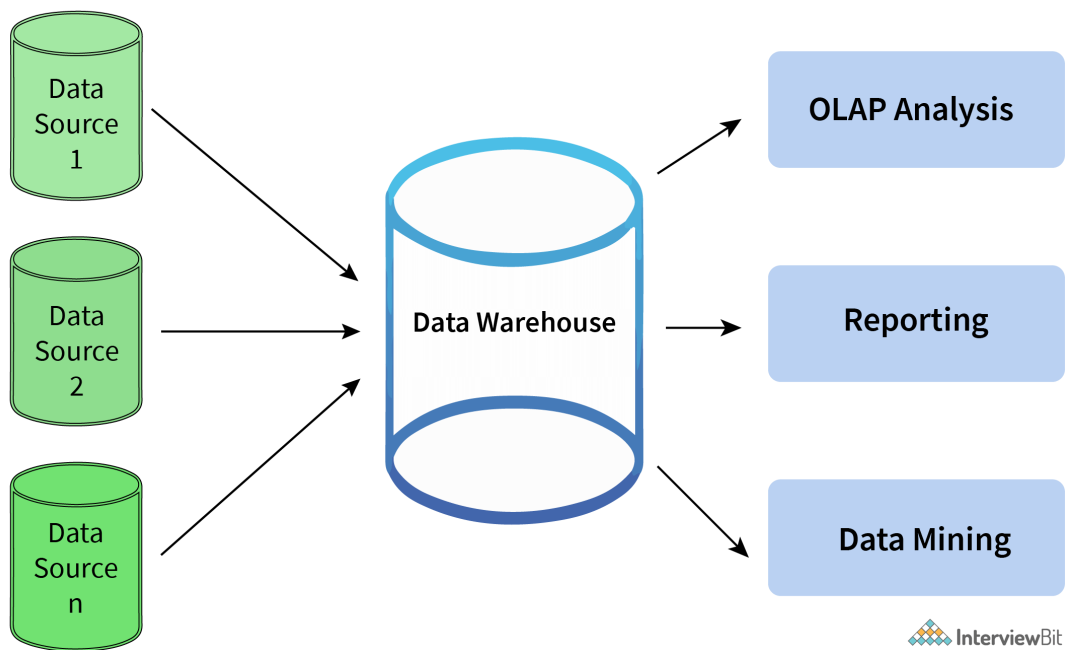
7. Are NULL values in a database the same as that of blank space or zero?

No, a NULL value is very different from that of zero and blank space as it represents a value that is assigned, unknown, unavailable, or not applicable as compared to blank space which represents a character and zero represents a number.

Example: NULL value in “number_of_courses” taken by a student represents that its value is unknown whereas 0 in it means that the student hasn't taken any courses.

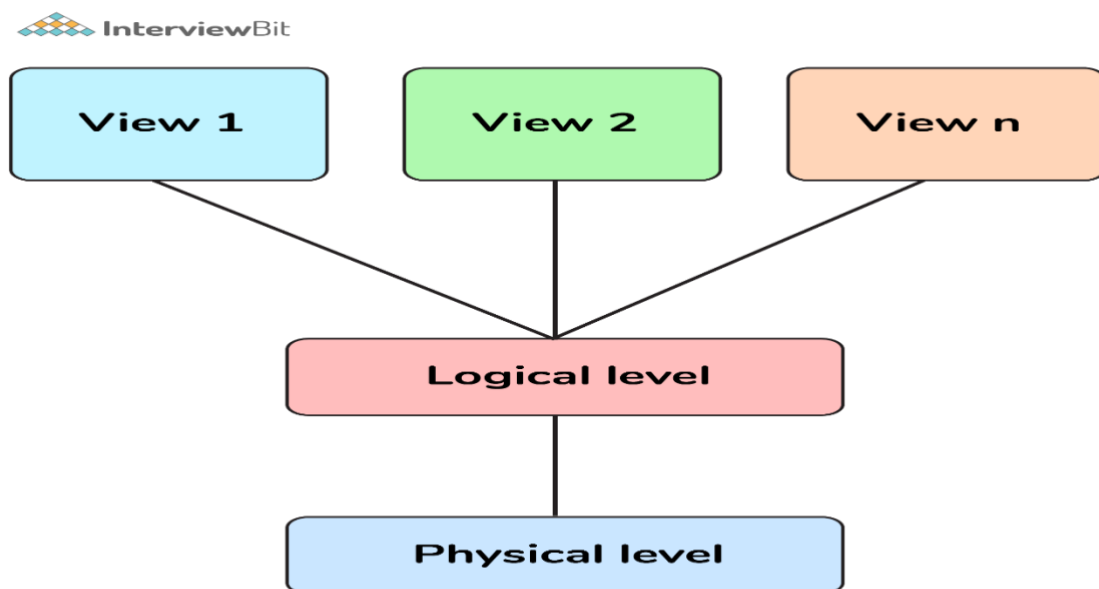
8. What is meant by Data Warehousing?

The process of collecting, extracting, transforming, and loading data from multiple sources and storing them into one database is known as data warehousing. A data warehouse can be considered as a central repository where data flows from transactional systems and other relational databases and is used for data analytics. A data warehouse comprises a wide variety of organization's historical data that supports the decision-making process in an organization.



9. Explain different levels of data abstraction in a DBMS.

The process of hiding irrelevant details from users is known as data abstraction. Data abstraction can be divided into 3 levels:

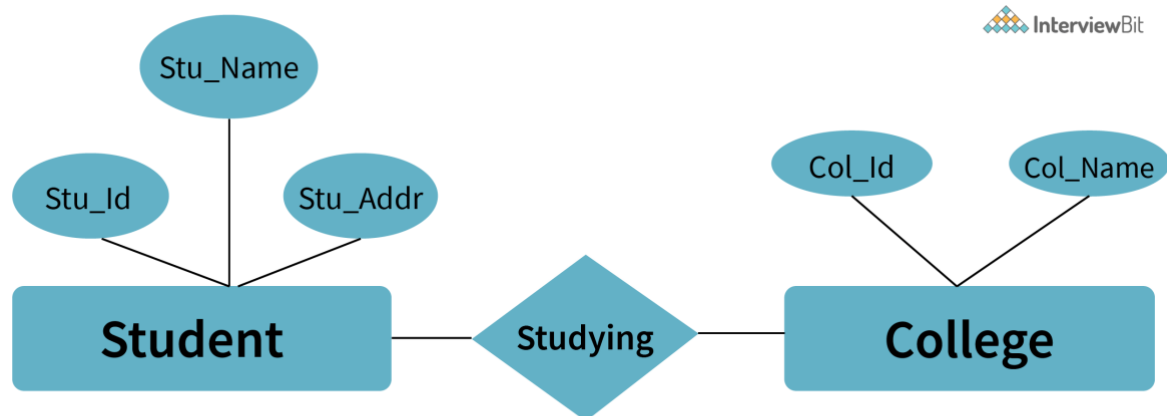


Three levels of data abstraction

- **Physical Level:** it is the lowest level and is managed by DBMS. This level consists of data storage descriptions and the details of this level are typically hidden from system admins, developers, and users.
- **Conceptual or Logical level:** it is the level on which developers and system admins work and it determines what data is stored in the database and what is the relationship between the data points.
- **External or View level:** it is the level that describes only part of the database and hides the details of the table schema and its physical storage from the users. The result of a query is an example of View level data abstraction. A view is a virtual table created by selecting fields from one or more tables present in the database.

10. What is meant by an entity-relationship (E-R) model? Explain the terms Entity, Entity Type, and Entity Set in DBMS.

An entity-relationship model is a diagrammatic approach to a database design where real-world objects are represented as entities and relationships between them are mentioned.



Sample E-R Diagram

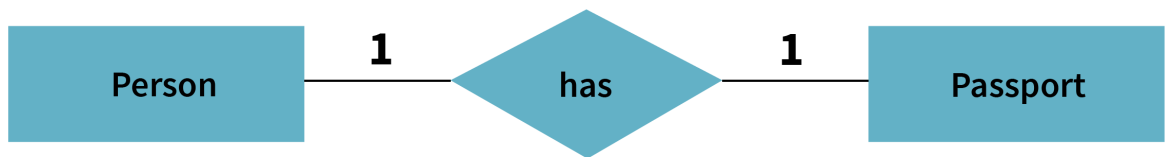
- **Entity:** An entity is defined as a real-world object having attributes that represent characteristics of that particular object. For example, a student, an employee, or a teacher represents an entity.
- **Entity Type:** An entity type is defined as a collection of entities that have the same attributes. One or more related tables in a database represent an entity type. Entity type or attributes can be understood as a characteristic which uniquely identifies the entity. For example, a student represents an entity that has attributes such as student_id, student_name, etc.

- **Entity Set:** An entity set can be defined as a set of all the entities present in a specific entity type in a database. For example, a set of all the students, employees, teachers, etc. represent an entity set.

11. Explain different types of relationships amongst tables in a DBMS.

Following are different types of relationship amongst tables in a DBMS system:

- **One to One Relationship:** This type of relationship is applied when a particular row in table X is linked to a singular row in table Y.



 InterviewBit

- **One to Many Relationship:** This type of relationship is applied when a single row in table X is related to many rows in table Y.



 InterviewBit

- **Many to Many Relationship:** This type of relationship is applied when multiple rows in table X can be linked to multiple rows in table Y.



 InterviewBit

- **Self Referencing Relationship:** This type of relationship is applied when a particular row in table X is associated with the same table.



- It removes complete data from a table in a database.
- It can be rolled back even if required. (truncate can be rolled back in some databases depending on their version but it can be tricky and can lead to data loss). Check this link for more details
- It doesn't maintain a log and deletes the whole table at once and hence it's fast.

14. What is a lock. Explain the major difference between a shared lock and an exclusive lock during a transaction in a database.

A database lock is a mechanism to protect a shared piece of data from getting updated by two or more database users at the same time. When a single database user or session has acquired a lock then no other database user or session can modify that data until the lock is released.

- **Shared Lock:** A shared lock is required for reading a data item and many transactions may hold a lock on the same data item in a shared lock. Multiple transactions are allowed to read the data items in a shared lock.
- **Exclusive lock:** An exclusive lock is a lock on any transaction that is about to perform a write operation. This type of lock doesn't allow more than one transaction and hence prevents any inconsistency in the database.

15. What is meant by normalization and denormalization?

Normalization is a process of reducing redundancy by organizing the data into multiple tables. Normalization leads to better usage of disk spaces and makes it easier to maintain the integrity of the database.

Denormalization is the reverse process of normalization as it combines the tables which have been normalized into a single table so that data retrieval becomes faster. JOIN operation allows us to create a denormalized form of the data by reversing the normalization.

16. Explain different types of Normalization forms in a DBMS.

Following are the major normalization forms in a DBMS:



Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First street Plot No 4	Pirates. the Caribbean Clash of the Titans	Ms.
Robert Phil	3rd street 34	Forgetting Sarah Marshal Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

InterviewBit

Considering the above Table-1 as the reference example for understanding different normalization forms.

- 1NF: It is known as the first normal form and is the simplest type of normalization that you can implement in a database. A table to be in its first normal form should satisfy the following conditions:
 - Every column must have a single value and should be atomic.
 - Duplicate columns from the same table should be removed.
 - Separate tables should be created for each group of related data and each row should be identified with a unique column.

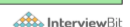
Full Names	Physical Address	Movies Rented	Salutation
Janet Jones	First street Plot No 4	Pirates. the Caribbean	Ms.
Janet Jones	First street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.



Table-1 converted to 1NF form

- **2NF:** It is known as the second normal form. A table to be in its second normal form should satisfy the following conditions:
 - The table should be in its 1NF i.e. satisfy all the conditions of 1NF.
 - Every non-prime attribute of the table should be fully functionally dependent on the primary key i.e. every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted then even the non_key element will be saved in the database.

Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First street Plot No 4	Ms.
2	Robert Phil	3rd street 34	Mr.
3	Robert Phil	5th Avenue	Mr.



Membership ID	Movies Rented
1	Pirates. the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans



Breaking Table-1 into 2 different tables to move it to 2NF.

- **3NF:** It is known as the third normal form. A table to be in its second normal form should satisfy the following conditions:

- The table should be in its 2NF i.e. satisfy all the conditions of 2NF.
- There is no transitive functional dependency of one attribute on any attribute in the same table.

Membership ID	Full Names	Physical Address	SalutationID
1	Janet Jones	First street Plot No 4	2
2	Robert Phil	3rd street 34	1
3	Robert Phil	5th Avenue	1

 InterviewBit

Membership ID	Movies Rented
1	Pirates. the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

 InterviewBit

Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

 InterviewBit

Breaking Table-1 into 3 different tables to move it to 3NF.

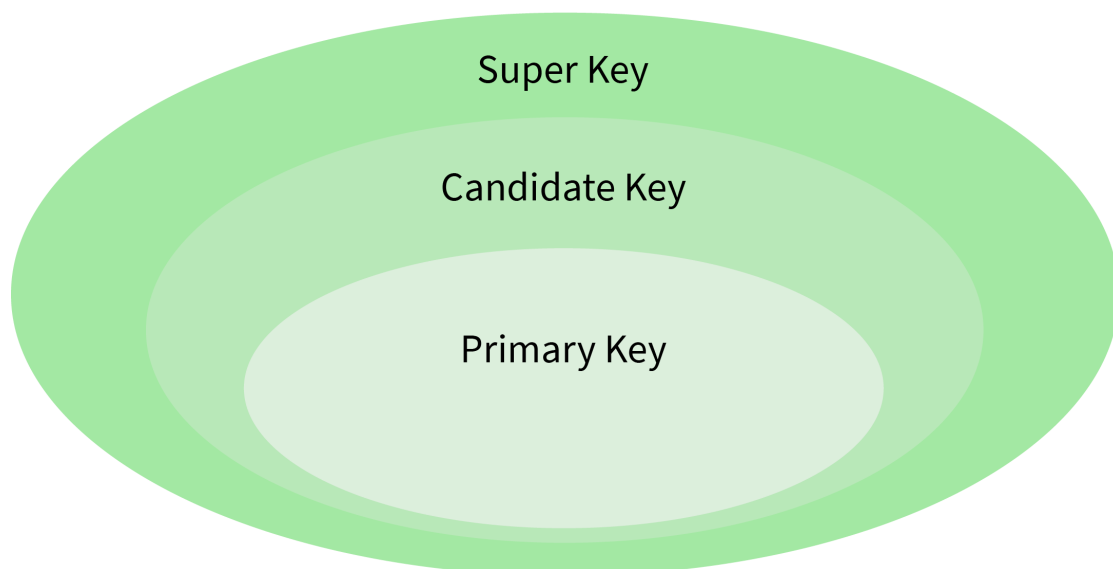
- **BCNF:** BCNF stands for Boyce-Codd Normal Form and is an advanced form of 3NF. It is also referred to as 3.5NF for the same reason. A table to be in its BCNF normal form should satisfy the following conditions:
 - The table should be in its 3NF i.e. satisfy all the conditions of 3NF.

- For every functional dependency of any attribute A on B ($A \rightarrow B$), A should be the super key of the table. It simply implies that A can't be a non-prime attribute if B is a prime attribute.

17. Explain different types of keys in a database.

There are mainly 7 types of keys in a database:

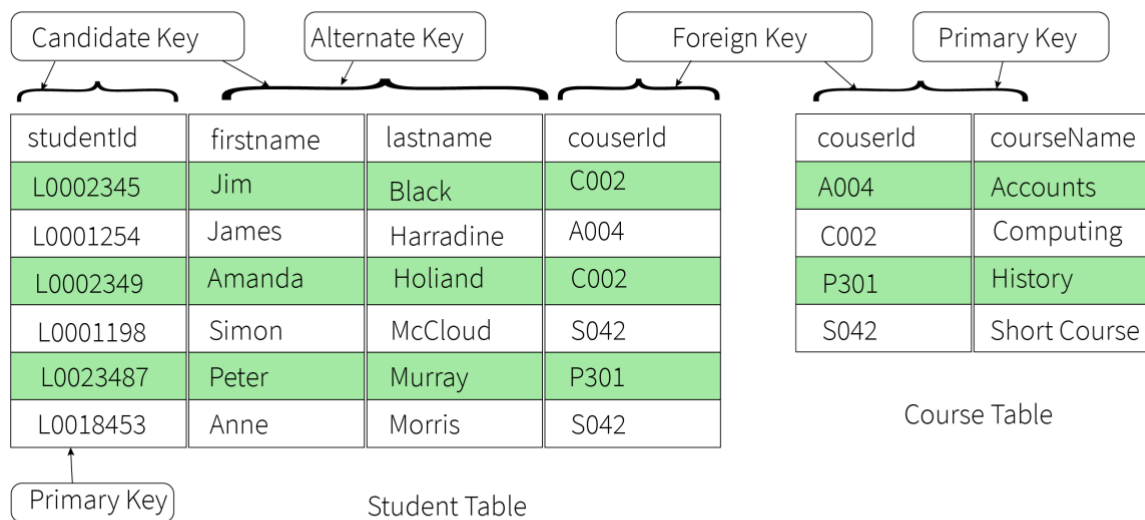
- **Candidate Key:** The candidate key represents a set of properties that can uniquely identify a table. Each table may have multiple candidate keys. One key amongst all candidate keys can be chosen as a primary key. In the below example since studentId and firstName can be considered as a Candidate Key since they can uniquely identify every tuple.
- **Super Key:** The super key defines a set of attributes that can uniquely identify a tuple. Candidate key and primary key are subsets of the super key, in other words, the super key is their superset.



 InterviewBit

- **Primary Key:** The primary key defines a set of attributes that are used to uniquely identify every tuple. In the below example studentId and firstName are candidate keys and any one of them can be chosen as a Primary Key. In the given example studentId is chosen as the primary key for the student table.
- **Unique Key:** The unique key is very similar to the primary key except that primary keys don't allow NULL values in the column but unique keys allow them. So essentially unique keys are primary keys with NULL values.
- **Alternate Key:** All the candidate keys which are not chosen as primary keys are considered as alternate Keys. In the below example, firstname and lastname are alternate keys in the database.

- **Foreign Key:** The foreign key defines an attribute that can only take the values present in one table common to the attribute present in another table. In the below example courseId from the Student table is a foreign key to the Course table, as both, the tables contain courseId as one of their attributes.
- **Composite Key:** A composite key refers to a combination of two or more columns that can uniquely identify each tuple in a table. In the below example the studentId and firstname can be grouped to uniquely identify every tuple in the table.



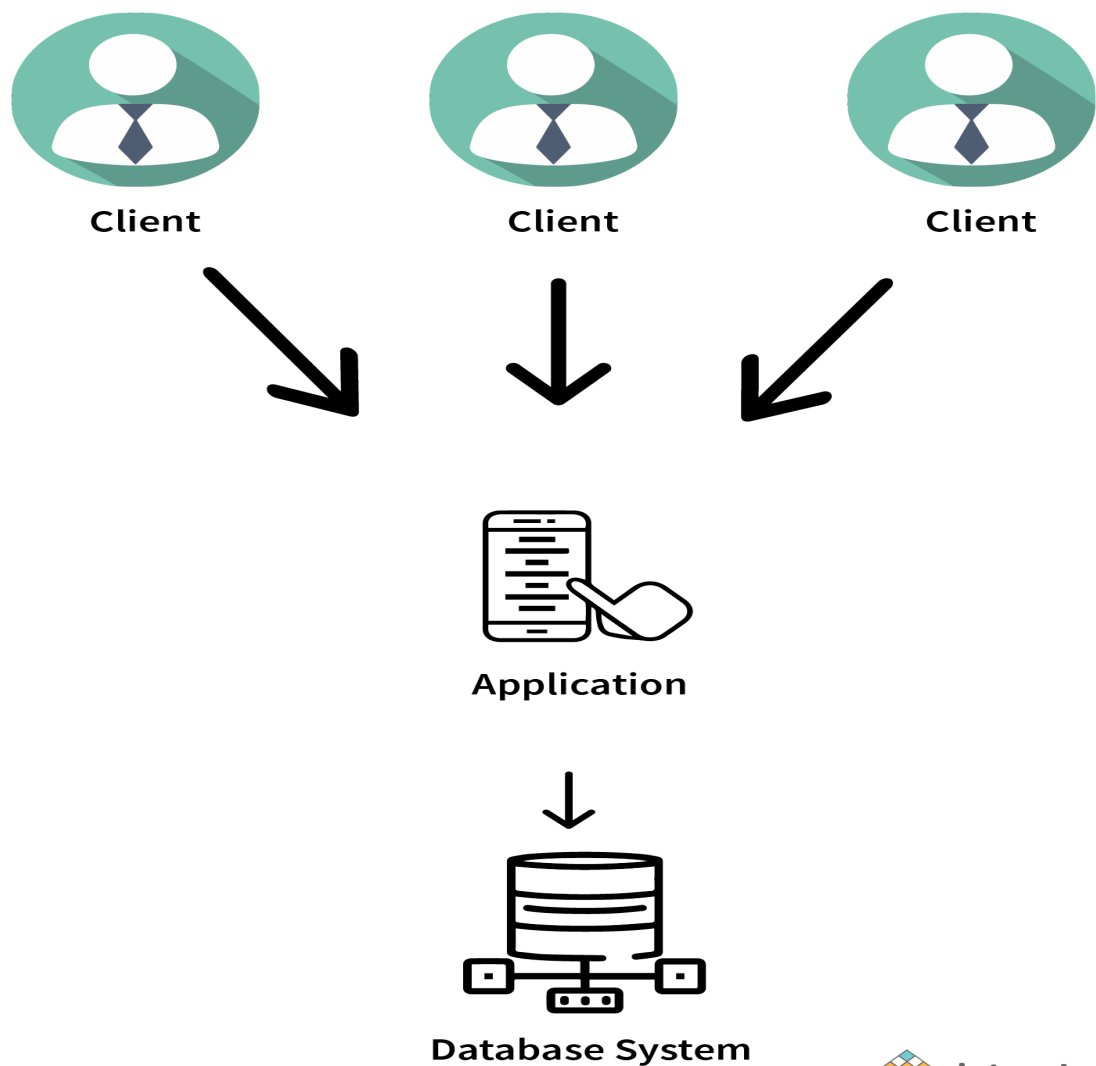
Relationship Between Keys



18. Explain the difference between a 2-tier and 3-tier architecture in a DBMS.

The **2-tier architecture** refers to the client-server architecture in which applications at the client end directly communicate with the database at the server end without any middleware involved.

Example – Contact Management System created using MS-Access or Railway Reservation System, etc.



The above picture represents a 2-tier architecture in a DBMS.

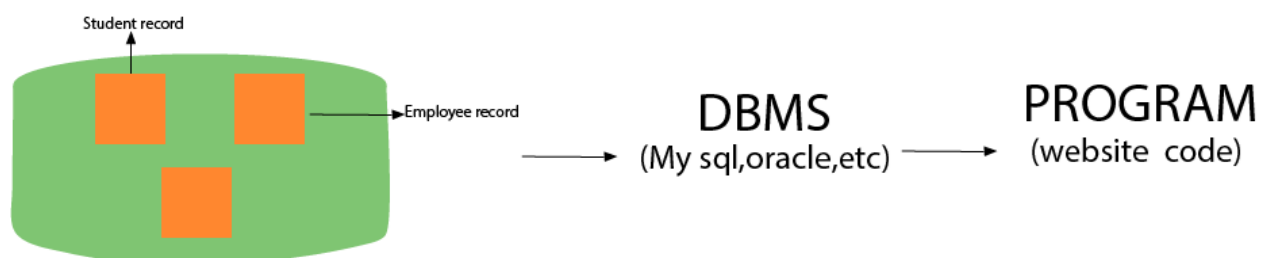
The **3-tier architecture** contains another layer between the client and the server to provide GUI to the users and make the system much more secure and accessible. In this type of architecture, the application present on the client end interacts with an application on the server end which further communicates with the database system.

Example – Designing registration form which contains a text box, label, button or a large website on the Internet, etc.

1) What is DBMS?

DBMS is a collection of programs that facilitates users to create and maintain a database. In other words, DBMS provides us an interface or tool for performing different operations such as the creation of a database, inserting data into it, deleting data from it, updating the data, etc. DBMS is a software in which data is stored in a more secure way as compared to the file-based system. Using DBMS, we can overcome many problems such as- data redundancy, data inconsistency, easy access, more organized and understandable, and so on. There is the name of some popular Database Management System- MySQL, Oracle, SQL Server, Amazon simple DB (Cloud-based), etc.

Working of DBMS is defined in the figure below.



2) What is a database?

A Database is a logical, consistent and organized collection of data that it can easily be accessed, managed and updated. Databases, also known as electronic databases are structured to provide the facility of creation, insertion, updating of the data efficiently and are stored in the form of a file or set of files, on the magnetic disk, tapes and another sort of secondary devices. Database mostly consists of the objects (tables), and tables include of the records and fields. Fields are the basic units of data storage, which contain the information about a particular aspect or attribute of the entity described by the database. DBMS is used for extraction of data from the database in the form of the queries.

3) What is a database system?

The collection of database and DBMS software together is known as a database system. Through the database system, we can perform many activities such as-

The data can be stored in the database with ease, and there are no issues of data redundancy and data inconsistency.

The data will be extracted from the database using DBMS software whenever required. So, the combination of database and DBMS software enables one to store, retrieve and access data with considerate accuracy and security.

4) What are the advantages of DBMS?

- Redundancy control
- Restriction for unauthorized access
- Provides multiple user interfaces
- Provides backup and recovery
- Enforces integrity constraints
- Ensure data consistency
- Easy accessibility
- Easy data extraction and data processing due to the use of queries

5) What is a checkpoint in DBMS?

The Checkpoint is a type of mechanism where all the previous logs are removed from the system and permanently stored in the storage disk.

There are two ways which can help the DBMS in recovering and maintaining the ACID properties, and they are- maintaining the log of each transaction and maintaining shadow pages. So, when it comes to log based recovery system, checkpoints come into existence. Checkpoints are those points to which the database engine can recover after a crash as a specified minimal point from where the transaction log record can be used to recover all the committed data up to the point of the crash.

6) When does checkpoint occur in DBMS?

A checkpoint is like a snapshot of the DBMS state. Using checkpoints, the DBMS can reduce the amount of work to be done during a restart in the event of subsequent crashes. Checkpoints are used for the recovery of the database after the system crash. Checkpoints are used in the log-based recovery system. When due to a system crash we need to restart the system then at that point we use checkpoints. So that, we don't have to perform the transactions from the very starting.

7) What do you mean by transparent DBMS?

The transparent DBMS is a type of DBMS which keeps its physical structure hidden from users. Physical structure or physical storage structure implies to the memory manager of the DBMS, and it describes how the data stored on disk.

8) What are the unary operations in Relational Algebra?

PROJECTION and SELECTION are the unary operations in relational algebra. Unary operations are those operations which use single operands. Unary operations are SELECTION, PROJECTION, and RENAME.

As in SELECTION relational operators are used for example - =,<=,>=, etc.

9) What is RDBMS?

RDBMS stands for Relational Database Management Systems. It is used to maintain the data records and indices in tables. RDBMS is the form of DBMS which uses the structure to identify and access data concerning the other piece of data in the database. RDBMS is the system that enables you to perform different operations such as- update, insert, delete, manipulate and administer a relational database with minimal difficulties. Most of the time RDBMS use SQL language because it is easily understandable and is used for often.

10) How many types of database languages are?

There are four types of database languages:

- **Data Definition Language (DDL)** e.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc. All these commands are used for updating the data that's why they are known as Data Definition Language.
- **Data Manipulation Language (DML)** e.g., SELECT, UPDATE, INSERT, DELETE, etc. These commands are used for the manipulation of already updated data that's why they are the part of Data Manipulation Language.
- **DATA Control Language (DCL)** e.g., GRANT and REVOKE. These commands are used for giving and removing the user access on the database. So, they are the part of Data Control Language.
- **Transaction Control Language (TCL)** e.g., COMMIT, ROLLBACK, and SAVEPOINT. These are the commands used for managing transactions in the database. TCL is used for managing the changes made by DML.

Database language implies the queries that are used for the update, modify and manipulate the data.

11) What do you understand by Data Model?

The Data model is specified as a collection of conceptual tools for describing data, data relationships, data semantics and constraints. These models are used to describe the relationship between the entities and their attributes.

There is the number of data models:

- Hierarchical data model
- network model
- relational model
- Entity-Relationship model and so on.

12) Define a Relation Schema and a Relation.

A Relation Schema is specified as a set of attributes. It is also known as table schema. It defines what the name of the table is. Relation schema is known as the blueprint with the help of which we can explain that how the data is organized into tables. This blueprint contains no data.

A relation is specified as a set of tuples. A relation is the set of related attributes with identifying key attributes

See this example:

Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t=(v_1, v_2, \dots, v_n)$.

13) What is a degree of Relation?

The degree of relation is a number of attribute of its relation schema. A degree of relation is also known as Cardinality it is defined as the number of occurrence of one entity which is connected to the number of occurrence of other entity. There are three degree of relation they are one-to-one(1:1), one-to-many(1:M), many-to-one(M:M).

14) What is the Relationship?

The Relationship is defined as an association among two or more entities. There are three type of relationships in DBMS-

One-To-One: Here one record of any object can be related to one record of another object.

One-To-Many (many-to-one): Here one record of any object can be related to many records of other object and vice versa.

Many-to-many: Here more than one records of an object can be related to n number of records of another object.

15) What are the disadvantages of file processing systems?

- Inconsistent
- Not secure
- Data redundancy
- Difficult in accessing data
- Data isolation
- Data integrity
- Concurrent access is not possible
- Limited data sharing
- Atomicity problem

16) What is data abstraction in DBMS?

Data abstraction in DBMS is a process of hiding irrelevant details from users. Because database systems are made of complex data structures so, it makes accessible the user interaction with the database.

For example: We know that most of the users prefer those systems which have a simple GUI that means no complex processing. So, to keep the user tuned and for making the access to the data easy, it is necessary to do data abstraction. In addition to it, data abstraction divides the system in different layers to make the work specified and well defined.

17) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what the relationship among those data is.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interacts with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is entirely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmers have the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

18) What is DDL (Data Definition Language)?

Data Definition Language (DDL) is a standard for commands which defines the different structures in a database. Most commonly DDL statements are CREATE, ALTER, and DROP. These commands are used for updating data into the database.

19) What is DML (Data Manipulation Language)?

Data Manipulation Language (DML) is a language that enables the user to access or manipulate data as organized by the appropriate data model. For example- SELECT, UPDATE, INSERT, DELETE.

There is two type of DML:

Procedural DML or Low level DML: It requires a user to specify what data are needed and how to get those data.

Non-Procedural DML or High level DML: It requires a user to specify what data are needed without specifying how to get those data.

20) Explain the functionality of DML Compiler.

The DML Compiler translates DML statements in a query language that the query evaluation engine can understand. DML Compiler is required because the DML is the family of syntax element which is very similar to the other programming language which requires compilation. So, it is essential to compile the code in the language which query evaluation engine can understand and then work on those queries with proper output.

21) What is Relational Algebra?

Relational Algebra is a Procedural Query Language which contains a set of operations that take one or two relations as input and produce a new relationship. Relational algebra is the basic set of operations for the relational model. The decisive point of relational algebra is that it is similar to the algebra which operates on the number.

There are few fundamental operations of relational algebra:

- select
- project
- set difference
- union
- rename,etc.

22) What is Relational Calculus?

Relational Calculus is a Non-procedural Query Language which uses mathematical predicate calculus instead of algebra. Relational calculus doesn't work on mathematics fundamentals such as algebra, differential, integration, etc. That's why it is also known as predicate calculus.

There is two type of relational calculus:

- Tuple relational calculus
- Domain relational calculus

23) What do you understand by query optimization?

The term query optimization specifies an efficient execution plan for evaluating a query that has the least estimated cost. The concept of query optimization came into the frame when there were a number of methods, and algorithms existed for the same task then the question arose that which one is more efficient and the process of determining the efficient way is known as query optimization.

There are many benefits of query optimization:

- It reduces the time and space complexity.
- More queries can be performed as due to optimization every query comparatively takes less time.
- User satisfaction as it will provide output fast

24) What do you mean by durability in DBMS?

Once the DBMS informs the user that a transaction has completed successfully, its effect should persist even if the system crashes before all its changes are reflected on disk. This property is called durability. Durability ensures that once the transaction is committed into the database, it will be stored in the non-volatile memory and after that system failure cannot affect that data anymore.

25) What is normalization?

Normalization is a process of analysing the given relation schemas according to their functional dependencies. It is used to minimize redundancy and also used to minimize insertion, deletion and update distractions. Normalization is considered as an essential process as it is used to avoid data redundancy, insertion anomaly, updation anomaly, deletion anomaly.

There most commonly used normal forms are:

- First Normal Form(1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)
- Boyce & Codd Normal Form(BCNF)

26) What is Denormalization?

Denormalization is the process of boosting up database performance and adding of redundant data which helps to get rid of complex data. Denormalization is a part of database optimization technique. This process is used to avoid the use of complex and costly joins. Denormalization doesn't refer to the thought of not to normalize instead of that denormalization takes place after normalization. In this process, firstly the redundancy of the data will be removed using normalization process than through denormalization process we will add redundant data as per the requirement so that we can easily avoid the costly joins.

27) What is functional Dependency?

Functional Dependency is the starting point of normalization. It exists when a relation between two attributes allow you to determine the corresponding attribute's value uniquely. The functional dependency is also known as database dependency and defines as the relationship which occurs when one attribute in a relation uniquely determines another attribute. It is written as $A \rightarrow B$ which means B is functionally dependent on A.

28) What is the E-R model?

E-R model is a short name for the Entity-Relationship model. This model is based on the real world. It contains necessary objects (known as entities) and the relationship among these objects. Here the primary objects are the entity, attribute of that entity, relationship set, an attribute of that relationship set can be mapped in the form of E-R diagram.

In E-R diagram, entities are represented by rectangles, relationships are represented by diamonds, attributes are the characteristics of entities and represented by ellipses, and data flow is represented through a straight line.

29) What is an entity?

The Entity is a set of attributes in a database. An entity can be a real-world object which physically exists in this world. All the entities have their attribute which in the real world considered as the characteristics of the object.

For example: In the employee database of a company, the employee, department, and the designation can be considered as the entities. These entities have some characteristics which will be the attributes of the corresponding entity.

30) What is an Entity type?

An entity type is specified as a collection of entities, having the same attributes. Entity type typically corresponds to one or several related tables in the database. A characteristic or trait which defines or uniquely identifies the entity is called entity type.

For example, a student has student_id, department, and course as its characteristics.

31) What is an Entity set?

The entity set specifies the collection of all entities of a particular entity type in the database. An entity set is known as the set of all the entities which share the same properties.

For example, a set of people, a set of students, a set of companies, etc.

32) What is an Extension of entity type?

An extension of an entity type is specified as a collection of entities of a particular entity type that are grouped into an entity set.

33) What is Weak Entity set?

An entity set that doesn't have sufficient attributes to form a primary key is referred to as a weak entity set. The member of a weak entity set is known as a subordinate entity. Weak entity set does not have a primary key, but we need a mean to differentiate among all those entries in the entity set that depend on one particular strong entity set.

34) What is an attribute?

An attribute refers to a database component. It is used to describe the property of an entity. An attribute can be defined as the characteristics of the entity. Entities can be uniquely identified using the attributes. Attributes represent the instances in the row of the database.

For example: If a student is an entity in the table then age will be the attribute of that student.

35) What are the integrity rules in DBMS?

Data integrity is one significant aspect while maintaining the database. So, data integrity is enforced in the database system by imposing a series of rules. Those set of integrity is known as the integrity rules.

There are two integrity rules in DBMS:

Entity Integrity : It specifies that "Primary key cannot have a NULL value."

Referential Integrity: It specifies that "Foreign Key can be either a NULL value or should be the Primary Key value of other relation

36) What do you mean by extension and intension?

Extension: The Extension is the number of tuples present in a table at any instance. It changes as the tuples are created, updated and destroyed. The actual data in the database change quite frequently. So, the data in the database at a particular moment in time is known as extension or database state or snapshot. It is time dependent.

Intension: Intension is also known as Data Schema and defined as the description of the database, which is specified during database design and is expected to remain unchanged. The Intension is a constant value that gives the name, structure of tables and the constraints laid on it.

37) What is System R? How many of its two major subsystems?

System R was designed and developed from 1974 to 1979 at IBM San Jose Research Centre. System R is the first implementation of SQL, which is the standard relational data query language, and it was also the first to demonstrate that RDBMS could provide better transaction processing performance. It is a prototype which is formed to show that it is possible to build a Relational System that can be used in a real-life environment to solve real-life problems.

Following are two major subsystems of System R:

- Research Storage
- System Relational Data System

38) What is Data Independence?

Data independence specifies that "the application is independent of the storage structure and access strategy of data." It makes you able to modify the schema definition at one level without altering the schema definition in the next higher level.

It makes you able to modify the schema definition in one level should not affect the schema definition in the next higher level.

There are two types of Data Independence:

Physical Data Independence: Physical data is the data stored in the database. It is in the bit-format. Modification in physical level should not affect the logical level.

For example: If we want to manipulate the data inside any table that should not change the format of the table.

Logical Data Independence: Logical data is the data about the database. It basically defines the structure. Such as tables stored in the database. Modification in logical level should not affect the view level.

For example: If we need to modify the format of any table, that modification should not affect the data inside it.

NOTE: Logical Data Independence is more difficult to achieve.

39) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what relationship among those data.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interact with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is absolutely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmer has the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

40) What is Join?

The Join operation is one of the most useful activities in relational algebra. It is most commonly used way to combine information from two or more relations. A Join is always performed on the basis of the same or related column. Most complex queries of SQL involve JOIN command.

There are following types of join:

- Inner joins: Inner join is of 3 categories. They are:
 - Theta join
 - Natural join
 - Equi join
- Outer joins: Outer join have three types. They are:
 - Left outer join
 - Right outer join
 - Full outer join

41) What is 1NF?

1NF is the **First Normal Form**. It is the simplest type of normalization that you can implement in a database. The primary objectives of 1NF are to:

- Every column must have atomic (single value)
- To Remove duplicate columns from the same table
- Create separate tables for each group of related data and identify each row with a unique column

42) What is 2NF?

2NF is the **Second Normal Form**. A table is said to be 2NF if it follows the following conditions:

- The table is in 1NF, i.e., firstly it is necessary that the table should follow the rules of 1NF.
- Every non-prime attribute is fully functionally dependent on the primary key, i.e., every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted, then even the non_key element will still be saved in the database.

43) What is 3NF?

3NF stands for **Third Normal Form**. A database is called in 3NF if it satisfies the following conditions:

- It is in second normal form.
- There is no transitive functional dependency.
- For example: $X \rightarrow Z$

Where:

$X \rightarrow Y$

Y does not \rightarrow X

$Y \rightarrow Z$ so, $X \rightarrow Z$

44) What is BCNF?

BCMF stands for **Boyce-Codd Normal Form**. It is an advanced version of 3NF, so it is also referred to as 3.5NF. BCNF is stricter than 3NF.

A table complies with BCNF if it satisfies the following conditions:

- It is in 3NF.
- For every functional dependency $X \rightarrow Y$, X should be the super key of the table. It merely means that X cannot be a non-prime attribute if Y is a prime attribute.

46) What is stored procedure?

A stored procedure is a group of SQL statements that have been created and stored in the database. The stored procedure increases the reusability as here the code or

the procedure is stored into the system and used again and again that makes the work easy, takes less time in processing and decreases the complexity of the system. So, if you have a code which you need to use again and again then save that code and call that code whenever it is required.

47) What is the difference between a DELETE command and TRUNCATE command?

DELETE command: DELETE command is used to delete rows from a table based on the condition that we provide in a WHERE clause.

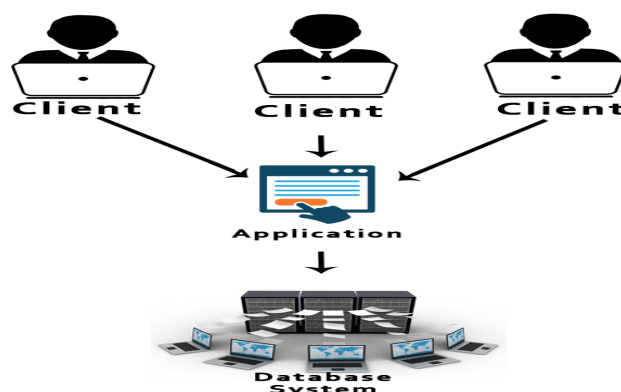
- DELETE command delete only those rows which are specified with the WHERE clause.
- DELETE command can be rolled back.
- DELETE command maintain a log, that's why it is slow.
- DELETE use row lock while performing DELETE function.

TRUNCATE command: TRUNCATE command is used to remove all rows (complete data) from a table. It is similar to the DELETE command with no WHERE clause.

- The TRUNCATE command removes all the rows from the table.
- The TRUNCATE command cannot be rolled back.
- The TRUNCATE command doesn't maintain a log. That's why it is fast.
- TRUNCATE use table log while performing the TRUNCATE function.

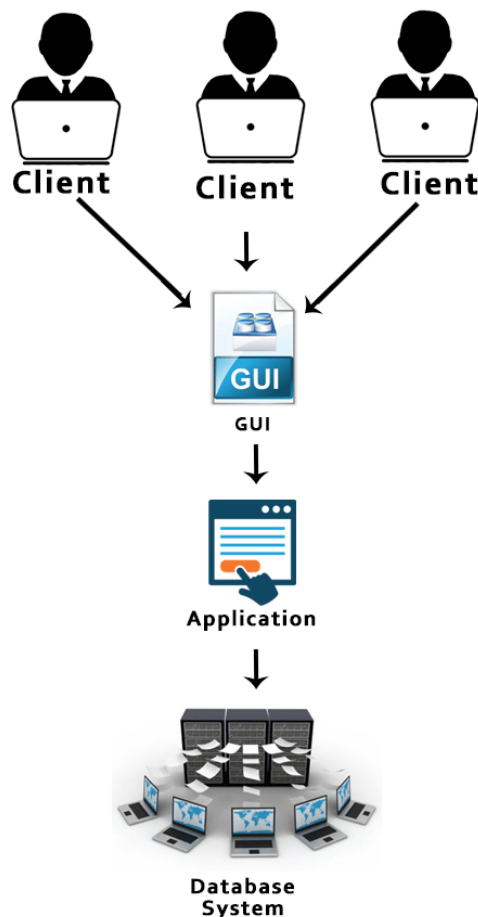
48) What is 2-Tier architecture?

The **2-Tier architecture** is the same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.



49) What is the 3-Tier architecture?

The 3-Tier architecture contains another layer between the client and server. Introduction of 3-tier architecture is for the ease of the users as it provides the GUI, which, make the system secure and much more accessible. In this architecture, the application on the client-end interacts with an application on the server which further communicates with the database system.



50) How do you communicate with an RDBMS?

You have to use Structured Query Language (SQL) to communicate with the RDBMS. Using queries of SQL, we can give the input to the database and then after processing of the queries database will provide us the required output.

51) What is the difference between a shared lock and exclusive lock?

Shared lock: Shared lock is required for reading a data item. In the shared lock, many transactions may hold a lock on the same data item. When more than one transaction is allowed to read the data items then that is known as the shared lock.

Exclusive lock: When any transaction is about to perform the write operation, then the lock on the data item is an exclusive lock. Because, if we allow more than one transaction then that will lead to the inconsistency in the database.

52) Describe the types of keys?

There are following types of keys:

Primary key: The Primary key is an attribute in a table that can uniquely identify each record in a table. It is compulsory for every table.

Candidate key: The Candidate key is an attribute or set of an attribute which can uniquely identify a tuple. The Primary key can be selected from these attributes.

Super key: The Super key is a set of attributes which can uniquely identify a tuple. Super key is a superset of the candidate key.

Foreign key: The Foreign key is a primary key from one table, which has a relationship with another table. It acts as a cross-reference between tables.

Q1. What are the differences between a DBMS and RDBMS?

DBMS	RDBMS
Provides an organized way of managing, retrieving, and storing from a collection of logically related information	Provides the same as that of DBMS, but it provides relational integrity

Q2. Explain the terms database and DBMS. Also, mention the different types of DBMS.

A software application that interacts with databases, applications, and users to capture and analyze the required data. The data stored in the database can be retrieved, deleted and modified based on the client's requirement.

The different types of DBMS are as follows:

- **Relational DBMS (RDBMS):** This type of DBMS, uses a structure which allows the users to access data in relation to another piece of data in a database. In this type of DBMS, data is stored in the form of tables.
- **Hierarchical DBMS:** As the name suggests, this type of DBMS has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.
- **Network DBMS:** This type of DBMS supports many-to-many relations wherein multiple member records can be linked.

- **Object-oriented DBMS:** Uses small individual software called object to store pieces of data and the instructions for the actions to be done with the data.

Q3. What are the advantages of DBMS?

The advantages of DBMS are as follows:

- **Sharing of Data:** Multiple users can use data from the same database simultaneously.
- **Integrity constraints:** These constraints allow the data to be stored in a database in a refined manner.
- **Redundancy control:** Supports a mechanism to control the redundancy of data by integrating all the data into a single database.
- **Data Independence:** Allows to change the structure of the data without affecting the structure of any of the running application programs.
- **Provide backup and recovery facility:** Provides a feature of 'backup and recovery' to automatically create the data backup and restore the data as and when required.

Q4. Mention the different languages present in DBMS

The different languages present in DBMS are as follows:

- **DDL(Data Definition Language)** – Consists of commands which are used to define the database.
- **DML(Data Manipulation Language)** – Consists of commands which are used to manipulate the data present in the database.
- **DCL(Data Control Language)** – Consists of commands which deal with the user permissions and controls of the database system.
- **TCL(Transaction Control Language)** – Consist of commands which deal with the transaction of the database.

Q5. What do you understand by query optimization?

Query optimization is the phase that identifies a plan for evaluation query that has the least estimated cost. This phase comes into the picture when there are a lot of algorithms and methods to execute the same task.

The advantages of query optimization are as follows:

- The output is provided faster
- A larger number of queries can be executed in less time
- Reduces time and space complexity

Q6. Do we consider NULL values the same as that of blank space or zero?

A NULL value is not at all same as that of zero or a blank space. The NULL value represents a value which is unavailable, unknown, assigned or not applicable whereas zero is a number and blank space is a character.

Q7. What do you understand by aggregation and atomicity?

Aggregation	Atomicity
This is a feature of the E-R model which allows a relationship set to participate in another relationship set.	This property states that a database modification must either follow all the rules or nothing at all. So, if one part of the transaction fails, then the entire transaction fails.

Q8. What are the different levels of abstraction in the DBMS?

There are three levels of data abstraction in DBMS. They are:

- **Physical Level:** It is the lowest level of abstraction and describes how the data is stored.
- **Logical Level:** This is the next level of abstraction after the Physical level. This layer determines what data is stored in the database, and what is the relationship between the data points.
- **View Level:** The View Level is the highest level of abstraction and it describes only a part of the entire database.

Q9. What is an entity-relationship model?

It is a diagrammatic approach to database design, where you represent real-world objects as entities and mention relationships between them. This approach helps the team of DBAs' to understand the schema easily.

Q10. What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?

- **Entity:** An entity is a real-world object having attributes, which are nothing but characteristics of that particular object. For example, an employee can be an entity. This particular entity can have attributes such as empid, empname, etc.
- **Entity Type:** Entity type is nothing but a collection of entities, having the same attributes. Generally, an entity type refers to one or more related tables in a particular database. So, you can understand, entity type as a characteristic which uniquely identifies the entity. For example, An employee can have attributes such as empid, empname, department, etc.

- **Entity Set:** An entity set is the collection of all the entities of a particular entity type in a database. For example, a set of employees, a set of companies, and a set of people can come under an entity set.

Q11. What are relationships and mention different types of relationships in the DBMS

A relationship in DBMS is the scenario where two entities are related to each other. In such a scenario, the table consisting of foreign key references to that of a primary key of the other table.

The different types of relationships in DBMS are as follows:

- **One-to-One Relationship** – Used when a single row in Table A is related to a single row in Table B.
- **One-to-Many Relationship** – Used when a single row in Table A is related to many rows in table B.
- **Many-to-Many Relationship** – Used when many rows in table A can be related to many rows in table B.
- **Self -Referencing Relationship** – Used when a record in table A is related to the same table itself.

Q12. What is concurrency control?

This is a process of managing simultaneous operations in a database so that database integrity is not compromised. The following are the two approaches involved in concurrency control:

- **Optimistic approach** – Involves versioning
- **Pessimistic approach** – Involves locking

Q15. What are the different types of keys in the database?

There are mainly 7 types of Keys, that can be considered in a database. I am going to consider the below tables to explain to you the various keys.

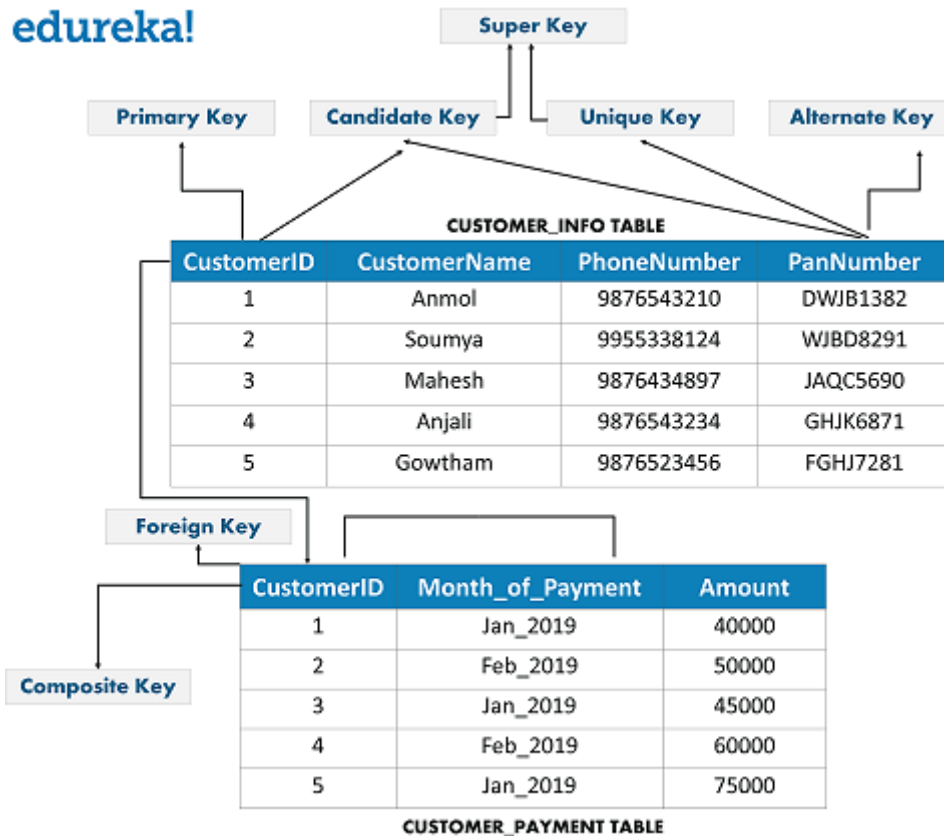


Fig 1: Different Types of Keys in Database – DBMS Interview Questions

- **Candidate Key** – This is a set of attributes which can uniquely identify a table. Each table can have more than a candidate key. Apart from this, out of all the candidate keys, one key can be chosen as the Primary key. In the above example, since CustomerID and PanNumber can uniquely identify every tuple, they would be considered as a Candidate Key.
- **Super Key** – This is a set of attributes which can uniquely identify a tuple. So, a candidate key, primary key, and a unique key is a superkey, but vice-versa isn't true.
- **Primary Key** – This is a set of attributes which are used to uniquely identify every tuple. In the above example, since CustomerID and PanNumber are candidate keys, any one of them can be chosen as a Primary Key. Here CustomerID is chosen as the primary key.
- **Unique Key** – The unique key is similar to the primary key, but allows NULL values in the column. Here the PanNumber can be considered as a unique key.

- **Alternate Key** – Alternate Keys are the candidate keys, which are not chosen as a Primary key. From the above example, the alternate key is PanNumber
- **Foreign Key** – An attribute that can only take the values present as the values of some other attribute, is the foreign key to the attribute to which it refers. in the above example, the CustomerID from the Customers Table is referred to the CustomerID from the Customer_Payment Table.
- **Composite Key** – A composite key is a combination of two or more columns that identify each tuple uniquely. Here, the CustomerID and Date_of_Payment can be grouped together to uniquely identify every tuple in the table.

Q16. What do you understand by correlated subqueries in DBMS?

A correlated subquery is also a sort of subquery reliant on another query. So, when subqueries are executed for each of the rows of outer queries, then they are termed as correlated subqueries. Each subquery is executed a single time for every row of the outer query.

You can also understand correlated subqueries as those queries, which are used for row-by-row processing by the parent statement. Here, the parent statement can be SELECT, UPDATE or DELETE statement.

Q17. Explain Database partitioning and its importance.

Data partitioning is the process of dividing a logical database into independent units for the betterment of availability, performance, and manageability.

The importance of database partitioning is as follows:

- Enables you to access large parts of a specific partition
- Cheap and slower storage can be used to store data
- Improves query performance

Q18. What do you understand by functional dependency and transitive dependency in DBMS?

Functional Dependency: A functional dependency is a constraint that is used in describing the relationship among different attributes in a relation.

Example: Consider a relation “A1” having attributes X and Y. The functional dependency among these two attributes will be $X \rightarrow Y$, this implies that Y is functionally dependent on X.

Transitive Dependency: A transitive dependency is a constraint that can only occur in a relation of three or more attributes.

Example: Consider a relation “A1” having attributes X, Y and Z. Now, $X \rightarrow Z$ is said to hold transitive dependency, only if the following functional dependencies holds true:

- $X \rightarrow Y$
- $Y \not\rightarrow X$
- $Y \rightarrow Z$

Q19. What is the difference between two and three-tier architectures?

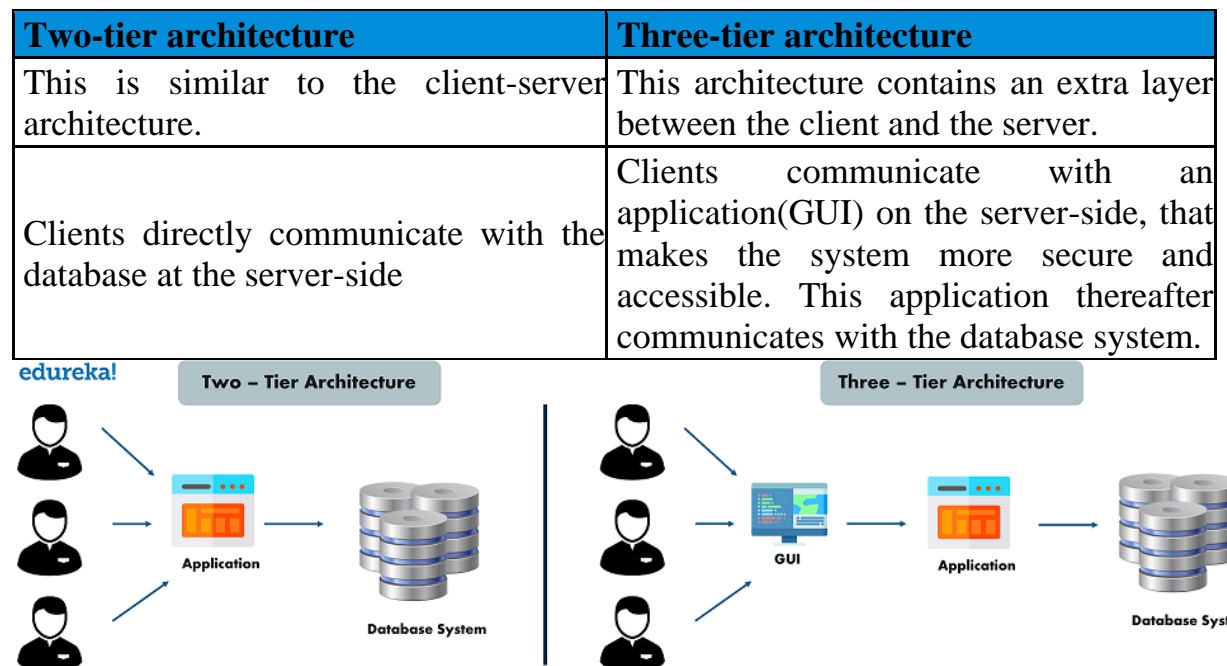


Fig 2: Two-Tier vs Three-Tier Architecture – DBMS Interview Questions

Q20. Mention the differences between Unique Key and Primary Key

Unique Key	Primary Key
Unique Key can have a NULL value	The primary key cannot have a NULL value
Each table can have more than one unique key	Each table can have only one primary key

Q21. What is a checkpoint in DBMS and when does it occur?

A checkpoint is a mechanism where all the previous logs are removed from the system and are permanently stored on the storage disk. So, basically, checkpoints are those points from where the transaction log record can be used to recover all the committed data up to the point of crash.

Q22. Mention the differences between Trigger and Stored Procedures

Triggers	Stored Procedures
A special kind of stored procedure that is not called directly by a user. In fact, a trigger is created and is programmed to fire when a specific event occurs.	A group of SQL statements which can be reused again and again. These statements are created and stored in the database.
A trigger cannot be called or execute directly by a user. Only when the corresponding events are fired, triggers are created.	Can execute stored procedures by using the exec command, whenever we want.
You cannot schedule a trigger.	You can schedule a job to execute the stored procedure on a pre-defined time.
Cannot directly call another trigger within a trigger.	Call a stored procedure from another stored procedure.
Parameters cannot be passed as input	Parameters can be passed as input
Cannot return values.	Can return zero or n values.
Transactions are not allowed within a trigger.	You can use transactions within a stored procedure.

Q23. What are the differences between Hash join, Merge join and Nested loops?

Hash join	Merge join	Nested loops
The hash join is used when you have to join large tables.	Merge join is used when projections of the joined tables are sorted on the join columns.	The nested loop consists of an outer loop and an inner loop.

Q24. What do you understand by Proactive, Retroactive and Simultaneous Update?

- **Proactive Update:** These updates are applied to the database before it becomes effective in the real-world environment.
- **Retroactive Update:** These retroactive updates are applied to a database after it becomes effective in the real-world environment.
- **Simultaneous Update:** These updates are applied to the database at the same instance of time as it becomes effective in a real-world environment.

Q25. What are indexes? Mention the differences between the clustered and non-clustered index

Indexes are data structures responsible for improving the speed of data retrieval operations on a table. This data structure uses more storage space to maintain extra copies of data by using additional writes. So, indexes are mainly used for searching algorithms, where you wish to retrieve data in a quick manner.

The differences between clustered and non-clustered index are as follows:

Clustered Index	Non-clustered Index
A clustered index is faster	Non clustered index is relatively slower
Alters the way records are stored in a database as it sorts out rows by the column which is set to be clustered index	Does not alter the way it was stored but it creates a separate object within a table which points back to the original table rows after searching
One table can only have one clustered index	One table can only have many non clustered indexes

Q26. What do you understand by intension and extension?

Intension: Intension or most commonly known as Database schema defines the description of the database. This is specified during the database design and mostly remains unchanged.

Extension: Extension is the number of tuples available in the database at any instance of time. This value keeps changing as and when the tuples are created, updated and destroyed. So, the data present in the database at a specific instance of time is known as the extension of the database or most commonly known as the snapshot of the database.

Q27. What do you understand by cursor? Mention the different types of cursor

A cursor is a database object which helps in manipulating data, row by row and represents a result set.

The types of cursor are as follows:

- **Implicit cursor:** This type of cursor is declared automatically as soon as the execution of SQL takes place. Here, the user is not indicated about the declaration of the cursor.
- **Explicit cursor:** This type of cursor is defined by the PL/ SQL, as it handles a query in more than a single row.

Q28. Explain the terms specialization and generalization

- **Specialization:** Specialization is a process of defining a set of subclasses of the entity type. Here, each subclass will contain all the attributes and relationships of the parent entity. Apart from this, the subclasses may contain additional attributes and relationships specific to itself.
- **Generalization:** Generalization is a process of finding relations, common attributes for a particular set of entities; and finally defining a common superclass for them.

Q29. What do you understand by Data Independence?

When you say an application has data independence, it implies that the application is independent of the storage structure and data access strategies of data.

Q30. What are the different integrity rules present in the DBMS?

The different integrity rules present in DBMS are as follows:

- **Entity Integrity:** This rule states that the value of the primary key can never be NULL. So, all the tuples in the column identified as the primary key should have a value.
- **Referential Integrity:** This rule states that either the value of the foreign key is NULL or it should be the primary key of any other relation.

Q31. What does Fill Factor concept mean with respect to indexes?

Fill Factor is used to mention the percentage of space left on every leaf-level page, which is packed with data. Usually, the default value is 100.

Q32. What is Index hunting and how does it help in improving query performance?

The process of boosting a collection of indexes is known as Index hunting. This is done as indexes improve the query performance and the speed at which they are processed.

It helps in improving query performance in the following way:

- The best queries are suggested using the query optimizer.
- Index, query distribution and their performance are used as metrics to check the effect
- Databases are tuned into a small collection of problem queries.

Q33. What are the differences between network and hierarchical database model?

Network Database Model	Hierarchical Database Model
Each parent node can have multiple children nodes and vice versa.	A top-down structure where each parent node can have many child nodes. But, a child node can have only a single parent node.
Supports one-to-one, one-to-many, and many-to-many relationships	Supports one-to-one and one-to-many relationships

Q34. Explain what is a deadlock and mention how it can be resolved?

Deadlock is a situation which occurs when two transactions wait on a resource which is locked or other transaction holds. Deadlocks can be prevented by making all the transactions acquire all the locks at the same instance of time. So, once deadlock occurs, the only way to cure is to abort one of the transactions and remove the partially completed work.

Q35. What are the differences between an exclusive lock and a shared lock?

Exclusive Lock	Shared Lock
An exclusive lock is a lock on a data item when a transaction is about to perform the write operation.	A shared lock allows more than one transaction to read the data items.

Next, in this article on DBMS interview questions, let us discuss the top questions asked about SQL.

Q1. What are the differences between DROP, TRUNCATE and DELETE commands?

DROP	TRUNCATE	DELETE
Used to delete a database, table or a view	Used to delete all rows from a table	Used to delete a row in the table
Data cannot be rolled back	Data cannot be rolled back	Data can be rolled back
A DDL command	A DDL command	A DML command.
Slower than TRUNCATE	Faster than DROP and DELETE	Slower than TRUNCATE
Deletes the full structure of the table	Preserves the structure of the table	Deletes the structure of the row from a table

Q2. What is SQL and where is it used?

SQL aka Structured Query Language is the core of the relational database which is used for accessing and managing the databases. This language is used to manipulate and retrieve data from a structured data format in the form of tables and holds relationships between those tables. So, in layman terms, you can use SQL to communicate with the database.

Q3. What do you understand by sub-queries in SQL?

A subquery is a query inside another query where a query is defined to retrieve data or information back from the database. In a subquery, the outer query is called as the main query whereas the inner query is called subquery. Subqueries are always executed first and the result of the subquery is passed on to the main query. It can be nested inside a SELECT, UPDATE or any other query. A subquery can also use any comparison operators such as >, < or =.

Q4. Mention the differences between UNION and UNION ALL

UNION	UNION ALL
Combines the result of two or more SELECT statements consisting of distinct values	Combines the result set of two or more SELECT statements consisting of duplicate values
Syntax: UNION	Syntax: UNION ALL
Has low performance than UNION ALL, as duplicate rows need to be removed.	Has better performance than UNION, as duplicate rows need not have to be removed.

Q5. What do you understand by CLAUSE in SQL?

CLAUSE in SQL is used to limit the result set by mentioning a condition to the query. So, you can use a CLAUSE to filter rows from the entire set of records.

Example: WHERE HAVING clause.

Q6. Mention the differences between HAVING and WHERE clause?

HAVING	WHERE
Used only with SELECT statement	Used in a GROUP BY clause
Used with the GROUP BY function in a query	Applied to each row before they are a part of the GROUP BY function in a query

Note: Whenever GROUP BY is not used, HAVING behaves like a WHERE clause.

Q7. How can you perform pattern matching in SQL?

You can perform pattern matching in SQL by using the LIKE operator. With the LIKE operator, you can use the following symbols:

1. %(Percentage sign) – To match zero or more characters.
2. _ (Underscore) –To match exactly one character.

Example:

```
SELECT * FROM Customers WHERE CustomerName LIKE 's%'
```

```
SELECT * FROM Customers WHERE CustomerName like 'xyz_'
```

Q8. Mention few case manipulation functions in SQL

There are three case manipulation functions in SQL, namely:

LOWER: This function returns the string in lowercase. It takes a string as an argument and returns it by converting it into lower case.

Syntax: LOWER('string')

UPPER: This function returns the string in uppercase. It takes a string as an argument and returns it by converting it into uppercase.

Syntax: UPPER('string')

INITCAP: This function returns the string with the first letter in uppercase and the rest of the letters in lowercase.

Syntax: INITCAP('string')

Q9. What are joins in SQL and what are the different types of joins?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

- Inner Join
- Right Join
- Left Join
- Full Join

Q10. What do you understand by the view and mention the steps to create, update and drop a view?

A view in SQL is a single table, which is derived from other tables. So, a view contains rows and columns similar to a real table and has fields from one or more table.

To create a view, use the following syntax:

```
1CREATE VIEW ViewName AS
2SELECT Column1, Column2, ..., ColumnN
3FROM TableName
4WHERE Condition;
```

To update a view, use the following syntax:

```
1CREATE VIEW OR REPLACE ViewName AS
2SELECT Column1, Column2, ..., ColumnN
3FROM TableName
4WHERE Condition;
```

To drop a view, use the following syntax:

```
1DROP VIEW ViewName;
```

Next, in this article on DBMS interview questions, let us discuss the most frequently asked queries about SQL.

SQL Query Based Interview Questions

Q1. Write a query to create a duplicate table with and without data present?

Consider you have a table named Customers, having details such as CustomerID, CustomerName and so on. Now, if you want to create a duplicate table named 'DuplicateCustomer' with the data present in it, you can mention the following query:

```
1CREATE TABLE DuplicateCustomer AS SELECT * FROM Customers;
```

Similarly, if you want to create a duplicate table without the data present, mention the following query:

```
1CREATE TABLE DuplicateCustomer AS SELECT * FROM Customers WHERE 1=2;
```

Q2. Mention a query to calculate the even and odd records from a table

To write a query to calculate the even and odd records from a table, you can write two different queries by using the MOD function.

So, if you want to retrieve the even records from a table, you can write a query as follows:

1SELECT CustomerID FROM (SELECT rowno, CustomerID from Customers) where mod(rowno,2)=1
Similarly, if you want to retrieve the odd records from a table, you can write a query as follows:

1SELECT CustomerID FROM (SELECT rowno, CustomerID from Customers) where mod(rowno,2)=1

Q3. Write a query to remove duplicate rows from a table?

To remove duplicate rows from a table, you have to initially select the duplicate rows from the table without using the DISTINCT keyword. So, to select the duplicate rows from the table, you can write a query as follows:

1SELECT CustomerNumber FROM Customers WHERE ROWID (SELECT MAX (rowid) FROM Customers WHERE CustomerNumber = CustomerNumber)
Now, to delete the duplicate records from the Customers table, mention the following query:

1DELETE FROM Customers WHERE ROWID (SELECT MAX (rowid) FROM Customers WHERE CustomerNumber = CustomerNumber)

Q4. Mention a query to add email validation to your database

1SELECT Email FROM Customers WHERE NOT REGEXP_LIKE(Email, '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$');

Q5. Write a query to retrieve the last day of next month in Oracle.

To write a query to retrieve the last day of the next month in Oracle, you can write a query as follows:

1SELECT LAST_DAY (ADD_MONTHS (SYSDATE,1)) from dual;

Q #1) What is DBMS used for?

Answer: DBMS, commonly known as Database Management System, is an application system whose main purpose revolves around the **data**. This is a system that allows its user to store the data, define it, retrieve it and update the information about the data inside the database.

Q #2) What is meant by a Database?

Answer: In simple terms, Database is a collection of data in some organized way to facilitate its user's to easily access, manage and upload the data.

Q #3) Why is the use of DBMS recommended? Explain by listing some of its major advantages.

Answer: Some of the major advantages of DBMS are as follows:

- **Controlled Redundancy:** DBMS supports a mechanism to control the redundancy of data inside the database by integrating all the data into a single database and as data is stored at only one place, the duplicity of data does not happen.

- **Data Sharing:** Sharing of data among multiple users simultaneously can also be done in DBMS as the same database will be shared among all the users and by different application programs.
- **Backup and Recovery Facility:** DBMS minimizes the pain of creating the backup of data again and again by providing a feature of 'backup and recovery' which automatically creates the data backup and restores the data whenever required.
- **Enforcement of Integrity Constraints:** Integrity Constraints are very important to be enforced on the data so that the refined data after putting some constraints are stored in the database and this is followed by DBMS.
- **Independence of data:** It simply means that you can change the structure of the data without affecting the structure of any of the application programs.

Q #4) What is the purpose of normalization in DBMS?

Answer: Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys in order to fulfill certain properties.

The properties include:

- To minimize the redundancy of the data.
- To minimize the Insert, Delete and Update Anomalies.

Q #5) What are the different types of languages that are available in the DBMS?

Answer: Basically, there are 3 types of languages in the DBMS as mentioned below:

- **DDL:** DDL is **Data Definition Language** which is used to define the database and schema structure by using some set of SQL Queries like **CREATE, ALTER, TRUNCATE, DROP** and **RENAME**.
- **DCL:** DCL is **Data Control Language** which is used to control the access of the users inside the database by using some set of SQL Queries like **GRANT** and **REVOKE**.
- **DML:** DML is **Data Manipulation Language** which is used to do some manipulations in the database like Insertion, Deletion, etc. by using some set of SQL Queries like **SELECT, INSERT, DELETE** and **UPDATE**.

Q #6) What is the purpose of SQL?

Answer: SQL stands for **Structured Query Language** whose main purpose is to interact with the relational databases in the form of inserting and updating/modifying the data in the database.

Q #7) Explain the concepts of a Primary key and Foreign Key.

Answer: **Primary Key** is used to uniquely identify the records in a database table while **Foreign Key** is mainly used to link two or more tables together, as this is a particular field(s) in one of the database tables which are the primary key of some other table.

Example: There are 2 tables – Employee and Department. Both have one common field/column as ‘ID’ where ID is the primary key of the **Employee** table while this is the foreign key for the **Department** table.

Q #8) What are the main differences between Primary key and Unique Key?

Answer: Given below are few differences:

- The main difference between the Primary key and Unique key is that the Primary key can never have a null value while the Unique key may consist of null value.
- In each table, there can be only one primary key while there can be more than one unique key in a table.

Q #9) What is the concept of sub-query in terms of SQL?

Answer: Sub-query is basically the query which is included inside some other query and can also be called as an inner query which is found inside the outer query.

Q #10) What is the use of DROP command and what are the differences between DROP, TRUNCATE and DELETE commands?

Answer: **DROP** command is a DDL command which is used to drop/delete the existing table, database, index or view from the database.

The major difference between DROP, TRUNCATE and DELETE commands are:

DROP and **TRUNCATE** commands are the **DDL** commands which are used to delete tables from the database and once the table gets deleted, all the privileges and indexes that are related to the table also get deleted. These 2 operations cannot be rolled back and so should be used only when necessary.

DELETE command, on the other hand, is a **DML** Command which is also used to delete rows from the table and this can be rolled back.

Note: It is recommended to use the ‘WHERE’ clause along with the **DELETE** command else the complete table will get deleted from the database.

Q #11) What is the main difference between UNION and UNION ALL?

Answer: **UNION** and **UNION ALL** are used to join the data from 2 or more tables but **UNION** removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas **UNION ALL** does not remove the duplicate rows, it just picks all the data from the tables.

Q #13) What is Correlated Subquery in DBMS?

Answer: A Subquery is also known as a nested query i.e. a query written inside some query. When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subquery.

Example of Non-Correlated Subquery is:

```
SELECT * from EMP WHERE 'RIYA' IN (SELECT Name from DEPT  
EMP.EMPID=DEPT.EMPID);
```

Here, the inner query is not executed for each of the rows of the outer query.

Q #14) Explain Entity, Entity Type, and Entity Set in DBMS?

Answer:

Entity is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database. **For Example**, any person, book, etc.

Entity Type is a collection of entities that have the same attributes. **For Example**, the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type which holds the entities having the same attributes.

Entity Set is a collection of entities of the same type. **For Example**, A collection of the employees of a firm.

Q #15) What are the different levels of abstraction in the DBMS?

Answer: There are 3 levels of data abstraction in the DBMS.

They include:

- **Physical Level:** This is the lowest level of the data abstraction which states how the data is stored in the database.
- **Logical Level:** This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.
- **View Level:** This is the highest level in the data abstraction which shows/states only a part of the database.

Q #16) What integrity rules exist in the DBMS?

Answer: There are 2 major integrity rules that exist in the DBMS.

They are:

- **Entity Integrity:** This states a very important rule that value of a Primary key can never have a NULL value.
- **Referential Integrity:** This rule is related to the Foreign key which states that either the value of a Foreign key is a NULL value or it should be the primary key of any other relation.

Q #17) What is E-R model in the DBMS?

Answer: E-R model is known as an **Entity-Relationship model** in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

Q #18) What is a functional dependency in the DBMS?

Answer: This is basically a constraint which is useful in describing the relationship among the different attributes in a relation.

Example: If there is some relation 'R1' which has 2 attributes as Y and Z then the functional dependency among these 2 attributes can be shown as **Y->Z** which states that Z is functionally dependent on Y.

Q #22) What is BCNF in the DBMS?

Answer: BCNF is the **Boyce Codd Normal Form** which is stricter than the 3NF. Any table is said to have in the BCNF if it satisfies the following 2 conditions:

- A table is in the 3NF.
- For each of the functional dependency $X \rightarrow Y$ that exists, X is the super key of a table.

Q #23) What is a CLAUSE in terms of SQL?

Answer: This is used with the SQL queries to fetch specific data as per the requirements on the basis of the conditions that are put in the SQL. This is very helpful in picking the selective records from the complete set of the records.

For Example, There is a query which has WHERE condition or the query with the HAVING clause.

Q #24) How can you get the alternate records from the table in the SQL?

Answer: If you want to fetch the odd numbers then the following query can be used:

SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=1;

If you want to fetch the even numbers, then the following query can be used:

SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=0;

Q #25) How is the pattern matching done in the SQL?

Answer: With the help of the LIKE operator, pattern matching is possible in the SQL. '%' is used with the LIKE operator when it matches with the 0 or more characters and '_' is used to match the one particular character.

Example:

SELECT * from Emp WHERE name like 'b%';

SELECT * from Emp WHERE name like 'hans_';

Q #26) What is a join in the SQL?

Answer: A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

Q #27) What are different types of joins in SQL?

Answer: There are 4 types of SQL Joins:

- **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.
- **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
- **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
- **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

Q #28) What is meant by trigger?

Answer: Trigger is one of the very important codes or programs which get executed **automatically in response to the events** that occur in a table or a view. **For Example,** If a new record is inserted in an employee database then the data gets created automatically in the related tables like salary, department and roles tables.

Q #29) Explain the Stored Procedure.

Answer: Stored Procedure is a group of SQL statements in the form of a function that has some unique name and is stored in relational database management systems(RDBMS) and can be accessed whenever required.

Q #30) What is RDBMS?

Answer: RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

Q #31) What are the different type of relationships in the DBMS?

Answer: Relationships in DBMS depicts an association between the tables.

Different types of relationships are:

- **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. **Example:** Among a married couple, both wife and husband can have only one spouse.
- **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one or none records in the related table. **Example:** A Mother can have many children.

- **Many-to-Many:** This states that both the tables can be related to many other tables. **Example:** One can have many siblings and so do they have.

1. Define a DBMS (Database Management System). What is it used for?

Answer: DBMS are software applications that help you build and maintain logically related data. Here are a few advantages of using a DBMS:

- Limit data redundancy
- Restrict unauthorized access
- Enforce integrity constraints.
- More data availability, sharing, protection, and storage

2. What is a database?

Answer: A database is a collection of ordered data that's stored electronically. Since it's organized and stored, you can easily access, aggregate, and manage it. We use a DBMS to manage and monitor a database.

3. How would you define a DBMS checkpoint?

Answer: A DBMS checkpoint is a point before which all the transactions in the DBMS were committed, and the database was in a consistent state. It is a point where you take a backup, and all the dirty pages (in-memory modified pages) and old transaction logs are cleared from the system to release memory space. The old data is then stored permanently on a storage disk. For example, a manual checkpoint can be added as follows:

CHECKPOINT 10

Where 10 is the duration in seconds.

4. Explain the different database language types.

Answer: There are four types of database languages:

- **DDL (Data Definition Language)** is used to define the database structure or skeleton, for example, creating the schema, tables, indexes, constraints, etc. Some DDL statements are created: alter, drop, rename, truncate, comment.
- **DCL (Data Control Language)** is used to retrieve saved data. Permissions and access control come under DCL and are done using the grant and revoke commands.
- **DML (Data Manipulation Language)** is used for accessing and manipulating data. Some DML statements are select, insert, delete, update, lock table, explain plan, etc. Basically, DML statements handle user requests.

- **TCL (Transaction Control Language)** is used to save or roll back changes done by DML. The commands used for TCL are commit and rollback.

5. What are the drawbacks of file processing systems?

Answer: There are several disadvantages of file processing systems:

- Data redundancy and inconsistencies
- Inefficient data access
- Data stored in different formats is challenging to share
- Issues with concurrent access

6. What is DBMS data abstraction?

Answer: Data abstraction is the hiding of irrelevant information from the user to facilitate a smoother interaction. There are three levels of abstraction:

- **Physical Level:** Lowest level of abstraction that tells how data is physically stored in memory. Data can be accessed through sequential or random access. Files are organized using B+ trees and hashing methods.
- **Logical Level:** Level that stores information in the form of tables. The relationships between various entities are also stored as simple structures.
- **View Level:** Top-most extent of abstraction. Users can view only a part of the actual database, in the form of rows and columns. It is possible to have multiple views of the same database. Storage and implementation details are hidden from the users.

7. What is functional dependency?

Answer: A functional dependency expresses the relationship between the non-key attributes and the primary key of a table. There are two types of functional dependency – trivial and non-trivial. The relationship can be described as $A \rightarrow B$, where A is the determinant, and B is the dependent column.

8. What is Denormalization?

Answer: Denormalization is a database optimization technique where redundant data is added to one or more tables to avoid joins. This makes data retrieval faster and more efficient.

9. Explain the difference between extension and intension.

Answer: Intension is the constant value specified during the database design phase, i.e. the description of the database or the schema, and is unlikely to change frequently. Extension is the actual data at a particular time, also referred to as the database snapshot. It is likely to change frequently.

10. What is System R?

Answer: System R is a database management system that provides a high level of data independence and database abstraction from the users. It has data control features like triggered transactions, authorization, integrity assertions, and provisions for data consistency.

11. What are the normal forms of BCNF, 1NF, 2NF, and 3NF?

Answer: All the above are normal forms of data, i.e. organizing data in such a way as to avoid data redundancy and insert, delete & update anomalies.

- **1NF:** Also known as the First Normal Form. This rule states that a column of a table can hold only an atomic (single) value.
- **2NF:** Short for second normal form. To apply the rule of 2NF, a table should be in 1NF. The rule states that any non-prime attribute (i.e. attribute that is not part of any candidate key) should not be dependent on the proper subset of the table's candidate key.
- **3NF:** Third normal form, the table should already be in 2NF. The rule states that any transitive dependency (functional) of a “non-prime” feature on a super key should be removed.
- **BCNF:** Advanced and stricter version of 3NF, also more popularly known as 3.5NF. For BCNF, a table should be in 3NF and follow the rule that for any functional dependency $A \rightarrow B$, A should be the super key of the table.

13. Explain the difference between a DELETE command and TRUNCATE command.

Answer:

DELETE	TRUNCATE
DML type	DDL type
Supports WHERE Condition	Doesn't support WHERE condition
Can acquire a row lock	Locks the entire table and page
Speed of execution is less	It is faster than delete
There is a log for each row	A single log that indicates the deallocation of a page

Example: DELETE FROM
employee WHERE emp_id = 2012;

Example: TRUNCATE
TABLE employee_temp

14. Explain 2-Tier Architecture.

Answer: In DBMS, 2-Tier architecture is a type of database architecture where the User Interface (UI) or view layer (also called the presentation) runs on a client machine (desktop, laptop, tablet, phone, etc.) and the data is stored on a server. To call the database, clients use the ODBC connectivity APIs. This architecture enhances the security of the database, as the client can't access the database directly.

15. What are the different types of database keys?

Answer: There are several types of DBMS keys:

- **Primary Key:** Column or set of columns that identify a particular row in a table
- **Super Key:** Single key or set of multiple keys that identifies rows of a table
- **Candidate Key:** A set of attributes that identify tuples of a table uniquely. It is essentially a super key without repeated attributes
- **Alternate Key:** One or more columns of a table that collectively identify each of a table uniquely
- **Foreign Key:** A common column that defines the relationship between two tables. maintains data integrity
- **Compound Key:** Consists of two more attributes that uniquely identify a record, even when the column is not unique by itself
- **Composite Key:** Primary key that has two or more attributes like {customer_id + mobile_number}
- **Surrogate Key:** Created when there is no naturally available primary key for a table. provides a unique identity to a row in the table

17. What is the purpose of normalization in DBMS?

Answer: Normalization is an important process that removes redundancy from a set of relations. It can reduce irregularities in the insert, delete and update commands by dividing huge tables into smaller tables and using keys to link them.

18. How Are entity, entity type, and entity set different?

Answer:

- **Entity:** An object with physical existence that is independent and can be differentiated from other objects: for example, an employee, student, course, job, etc.
- **Entity type:** A collection of similar entities.

- **Entity set:** A combination of entities of one type at any point in time

19. What is a **CLAUSE** in terms of SQL?

Answer: **CLAUSE** is an optional statement in SQL that defines a condition to present data. For example,

select * from employee WHERE emp_id = 2012;

Here, **WHERE** is the clause that identifies the row to be obtained. Some other clauses in SQL are **GROUP BY**, **HAVING**, and **ORDER BY**.

20. What is the difference between aggregation and atomicity?

Answer: Atomicity means that all actions are carried out at once, or none are carried out. This means that if a transaction is incomplete or fails in between everything will be rolled back to a previous stable state. Contrarily, we use aggregation to express relationships between various entities.

21. Explain the different types of relationships in the DBMS.

Answer: There are three types of relationships in DBMS:

- **One-to-One:** Each table record is related to only one record in another table. For example: **employee_id**, and **unique_ssn_id**
- **One-to-many or vice-versa (many-to-one):** A record in one table can be related to more than one record in another table. For example: **employee_id** and **phone_number**
- **Many-to-Many:** A record in the first table can be related to more than one record in the second table, and vice-versa. For example: **customer_id** and **product_id**.

22. Explain correlated subqueries in DBMS.

Answer: Correlated subqueries are subqueries that are executed once for each row processed by its main statement. For example:

select student_id, student_name, marks from students outer where salary > (select avg(marks) from students where subject = outer.students);

23. What is the difference between two- and three-tier architectures?

Answer: The two-tier architecture follows a client-server architecture, where there is a straight communication link between client and server. Two-tier architecture gives a high performance as there is no middle layer. The client can be any device like a mobile, desktop, or laptop, while all the data sits in a database.

Three-tier architecture consists of the following:

- **Presentation layer:** A web browser, java applet, WAP phone, etc.
- **Business layer:** Business logic like data validation, insertions, calculations, etc.
- **Data access layer:** This is a data source like database, mainframe system, or ERP system. The business layer acts as a middle tier that separates the business logic from the presentation and data layer for better reusability, data integrity, scalability, and performance.

• **What is the Difference between Trigger and Stored Procedures?**

Answer: Users can explicitly invoke stored procedures. Stored procedures can take certain input parameters and return output values, like any other program. Triggers are called spontaneously (on their own) when a specific event occurs. Triggers can't take any input values nor return any output values.

24. What is the difference between Hash Join, Merge Join, and Nested Loops?

Answer:

- **Hash-join:** In this type of join, the database does full scanning of the main table, builds a RAM hash table, and then searches for matching tables in the other table. Hash-join is faster than the nested loop join but consumes more RAM resources.
- **Merge join:** This join uses simple concatenation followed by sorting (removing duplicates) and is the most effective technique out of all the joins.
- **Nested loop join:** In this type of scan, the driving table (main) accesses rows through index range scans, and the result set of the driving table is nested with the probe of the other (secondary) table using the index scan method.

25. Distinguish between proactive, retroactive, and simultaneous Update.

Answer:

- **Proactive** updates are those that eventuate in the database before they become useful in the real world.
- **Retroactive** updates come about after they become effective in the real world.
- **Simultaneous** updates occur at the same time as they become effective in the real world.

26. Explain the differences between clustered and non-clustered indexes.

Answer:

Clustered Index	Non-clustered Index
Defines the order of physical data storage even if they are inserted in random order.	Does not sort the physical data. Indexes are stored in a different place than the actual data.
Only one clustered index per table as data can be sorted in a single way only.	More than one non-clustered index in a table is possible.
In many databases, the primary key constraint creates a clustered index on the same column.	A non-clustered index can be created as: <code>CREATE NON-CLUSTERED INDEX IX_tblStudent_deptt ON student(deptt ASC)</code>
Example: if there is a table student that has a primary key student_id, the data will be stored in ascending order of student_id like 1,2,3,4. The clustered index will be automatically created on student_id.	An index is stored in a separate table along with its address (a reference to the row), deptt row address ECE <address>
No extra space required for these indexes.	Consumes storage space.

27. What is the difference between specialization and generalization?

Answer:

Generalization	Specialization
Creating groupings from various entity sets	Creating subgrouping within an entity set
Starts with several entity sets and creates a high-level entity with some common features	Starts with a single entity set and then creates a different set using different features

Applied to a group of entities	Applied to a single entity
Follows bottom-up approach	Follows top-down approach

28. Explain the differences between network and hierarchical database models.

Answer: In a hierarchical database model, data is organized into nodes in a tree-like structure. A node is connected to only one parent node above it. Hence, data in this model has a one-to-many relationship. An example of this model is the Document Object Model (DOM) often used in web browsers.

The network database model is a refined version of a hierarchical model. Here, too, data is organized in a tree-like structure. However, one child node can be connected to multiple parent nodes. This gives rise to a many-to-many relationship between data nodes. IDMS (Integrated Database Management System), Integrated Data Store (IDS) are examples of Network Databases.

Hierarchical model	Network model
Parent-child relationship between records	Pointers or links relationships
Update and delete operations are prone to data inconsistencies	No data inconsistencies
Doesn't support many to many relationships	Supports many to many relationships
Creates a tree structure and data traversal is a bit complex	Creates graph structure where data traversal is easy, as a node can be accessed both ways, i.e. parent-child or vice-versa

29. What is deadlock?

Answer: Deadlock happens when a set of processes are blocked. In a deadlock, each process holds up a resource that the other process requires. The other process waits for a resource to be released by the same or another process.

30. What is an RDBMS?

Answer: An relational database management system (RDBMS) is an application that allows you to create, update, and administer a relational database. An RDBMS

is organized into tables, records, and columns, and database tables have a well-defined relationship. Tables communicate and share information, which enables data search, data organization, and reporting. An RDBMS is a subset of a DBMS.

31. What is an object-oriented database model?

Answer: In an object-oriented database model, data is represented by objects. For example, a multimedia file or record in a relational database is stored as a data object as opposed to an alphanumeric value.

32. What is SQL?

Answer: SQL (Structured Query Language) is a programming language used to communicate with data stored in databases. SQL language is relatively easy to write, read, and interpret.

33. What are DDL, DML, and DCL statements in SQL?

Answer:

- **DDL:** Data Definition Language is used to define the database and schema structure by using a set of SQL Queries like CREATE, ALTER, TRUNCATE, DROP and RENAME.
- **DCL:** Data Control Language is used to control the access of the users to the database by using a set of commands like GRANT and REVOKE in the SQL Query.
- **DML:** Data Manipulation Language is used for maintaining data by using SQL queries like SELECT, INSERT, DELETE and UPDATE.

34. What is index hunting?

Answer: A database index is a data structure that improves the speed of data retrieval operations on a database. Index hunting is the procedure of boosting the collection of indexes, done by using methods like query optimization and query distribution.

35. What is a distributed database?

Answer: A distributed database is a collection of multiple interconnected databases that are spread physically across various locations. The databases can be on the same network or multiple networks. A DDBMS (Distributed DBMS) integrates data logically so it appears as one single database to the user.

35. What is database partitioning and why is it important?

Answer: Database partitioning is a process where a logical database is divided into distinct independent parts. Database objects like tables, indexes are subdivided, managed, and accessed at the granular level.

Partitioning is a powerful functionality that increases performance with decreased cost. It improves the manageability and availability of data.

36. What is static SQL?

Answer: In a static SQL, the SQL statements are embedded or hard-coded in the application and they do not change at runtime. The process for data access is predetermined, hence more swift and efficient. The SQL statements are compiled at compile time

37. What is dynamic SQL?

Answer: In a dynamic SQL, SQL statements are constructed at runtime, where the application can allow the user to create the queries. Basically, you can build your query at runtime. It is comparatively slower than static SQL as the query is compiled at runtime.

38. Define data warehousing.

Answer: Data Warehousing is a technique that aggregates a large amount of data from one or more sources. Data analysis is performed on the data to make strategic business decisions for organizations.

39. Name one open-source RDBMS.

Answer: MySQL is an open-source relational database management system, which stores data in tables and maintains a relationship between the data. It uses the most powerful query language, SQL, for database access and has a very powerful syntax to create simple and complex queries for data retrieval and structure. Data in MySQL is organized and conforms to a certain format, and is the most popular structured database today. Its 'free' source code is available for study, modification, and distribution.

40. What is MongoDB?

Answer: MongoDB is a non-relational unstructured open-source database. This document-oriented database stores your data in collections made of individual documents. In MongoDB, a document is a big JSON object with no particular format or schema. MongoDB represents JSON documents in a binary-encoded format named BSON.

Read about the [difference between MongoDB vs MySQL](#).

41. What is BI (Business Intelligence)?

Answer: Business intelligence (BI) is a technology-driven process for analyzing data and presenting information to help executives, managers, and other corporate end-users make business decisions. There are many tools available like SAP

Business Intelligence, MicroStrategy, Sisense, and so on. The tools are user-friendly and help you gather data from varied sources for analysis.

42. What is the role of a DBA in DBMS?

Answer: The Database Administrator (DBA) plays some important roles in DBMS:

- Installing and configuring databases
- Performance monitoring
- Security planning and implementation
- Data backup and recovery
- Troubleshooting
- Data migration

43. What is an ER Diagram in DBMS?

Answer: An entity-relationship model or an entity-relationship diagram is a visual representation of data represented as entities. Attributes and relationships are set between entities.

44. What is an entity in an ER diagram?

Answer: An entity can be an identifiable, real-world object. For example, in a library database, books, publishers, and members can be considered entities. All these entities have some attributes or properties that give them their identity. In an ER model, the entities are related to each other.

45. What is data mining?

Answer: Data mining is a process of sorting through a large amount of data to identify patterns and trends. It uses complex mathematical and statistical algorithms to segment data for the prediction of likely outcomes. There are many tools for data mining like RapidMiner, Orange, Weka, and so on.

46. What is query optimization?

Answer: Query optimization is an important feature when it comes to the performance of a database. A query optimization identifies an execution plan that has the least estimated cost and time for evaluating and executing a query.

47. What is a catalog?

Answer: A catalog is a table that contains information like the structure of each file, the type, and storage format of each data item, and various constraints on the data. The information stored in the catalog is called metadata.

48. How many types of relationships exist in database designing?

Answer: There are three major relationship models in database design:

One-to-one: A row in table (A) is related to only one row in another table (B).

One-to-many: A row in table (A) is linked to many rows in another table (B). But a row in table (B) is linked to only one row in table (A).

Many-to-many: A row in table (A) is linked to many rows in table (B) and vice-versa.

49. What are the primitive operations common in all database management systems?

Answer: Addition, deletion, and modification are the most important primitive operations common to all DBMS.

50. What is cardinality in context to a database?

Answer: In SQL, the term cardinality refers to the uniqueness of data values contained in a particular column (attribute) of a database table. The lower the cardinality, the more duplicate values in a column.

51. What is SQL SERVER?

Answer: SQL Server is an RDBMS developed by Microsoft. It is a very stable and popular server. The latest version of SQL Server is SQL Server 2017.

52. When should we use indexes?

Answer: We can use indexes to enforce uniqueness, facilitate sorting, and enable fast retrieval by column values. A frequently-used column is a good candidate for an index to be used with suitable conditions in WHERE clauses.

53. What is a hashing technique in a database?

Answer: Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the shorter hashed key than to find it using the original value.

54. Describe concurrency control.

Answer: Concurrency control is the process of managing simultaneous operations on a database so that database integrity is not compromised. There are two approaches to concurrency control.

Locking: Controlling access to data items using locks

Versioning: Using Multi-Version Concurrency Control

1) Define Database.

A prearranged collection of figures known as data is called database.

2) What is DBMS?

Database Management Systems (DBMS) are applications designed especially which enable user interaction with other applications.

3) What are the various kinds of interactions catered by DBMS?

The various kind of interactions catered by DBMS are:

- Data definition
- Update
- Retrieval
- Administration

4) Segregate database technology's development.

The development of database technology is divided into:

- Structure or data model
- Navigational model
- SQL/ relational model

5) Who proposed the relational model?

Edgar F. Codd proposed the relational model in 1970.

6) What are the features of Database language?

A database language may also incorporate features like: DBMS-specific Configuration and management of storage engine Computations to modification of query results by computations, like summing, counting, averaging, grouping, sorting and cross-referencing Constraint enforcement Application Programming Interface

7) What do database languages do?

As special-purpose languages, they have:

- Data definition language
- Data manipulation language

- Query language

8) Define database model.

A data model determining fundamentally how data can be stored, manipulated and organised and the structure of the database logically is called database model.

9) What is SQL?

Structured Query Language (SQL) being ANSI standard language updates database and commands for accessing.

10) Enlist the various relationships of database.

The various relationships of database are:

- One-to-one: Single table having drawn relationship with another table having similar kind of columns.
- One-to-many: Two tables having primary and foreign key relation.
- Many-to-many: Junction table having many tables related to many tables.

11) Define Normalization.

Organized data void of inconsistent dependency and redundancy within a database is called normalization.

12) Enlist the advantages of normalizing database.

Advantages of normalizing database are:

- No duplicate entries
- Saves storage space
- Boasts the query performances.

13) Define Denormalization.

Boosting up database performance, adding of redundant data which in turn helps rid of complex data is called denormalization.

14) Define DDL and DML.

Managing properties and attributes of database is called Data Definition Language(DDL).

Manipulating data in a database such as inserting, updating, deleting is defined as Data Manipulation Language. (DML)

15) Enlist some commands of DDL.

They are:

CREATE:

Create is used in the CREATE TABLE statement. Syntax is:

CREATE TABLE [column name] ([column definitions]) [table parameters]

ALTER:

It helps in modification of an existing object of database. Its syntax is:

ALTER objecttype objectname parameters.

DROP:

It destroys an existing database, index, table or view. Its syntax is:

DROP objecttype objectname.

16) Define Union All operator and Union.

Full recordings of two tables is Union All operator. A distinct recording of two tables is Union.

17) Define cursor.

A database object which helps in manipulating data row by row representing a result set is called cursor.

18) Enlist the cursor types.

They are:

- Dynamic: it reflects changes while scrolling.
- Static: doesn't reflect changes while scrolling and works on recording of snapshot.
- Keyset: data modification without reflection of new data is seen.

19) Enlist the types of cursor.

They types of cursor are:

- Implicit cursor: Declared automatically as soon as the execution of SQL takes place without the awareness of the user.
- Explicit cursor: Defined by PL/ SQL which handles query in more than one row.

20) Define sub-query.

A query contained by a query is called Sub-query.

21) Why is group-clause used?

Group-clause uses aggregate values to be derived by collecting similar data.

22) Compare Non-clustered and clustered index

Both having B-tree structure, non-clustered index has data pointers enabling one table many non-clustered indexes while clustered index is distinct for every table.

23) Define Aggregate functions.

Functions which operate against a collection of values and returning single value is called aggregate functions

24) Define Scalar functions.

Scalar function is depended on the argument given and returns sole value.

25) What restrictions can you apply when you are creating views?

Restrictions that are applied are:

- Only the current database can have views.
- You are not liable to change any computed value in any particular view.
- Integrity constants decide the functionality of INSERT and DELETE.
- Full-text index definitions cannot be applied.
- Temporary views cannot be created.
- Temporary tables cannot contain views.
- No association with DEFAULT definitions.
- Triggers such as INSTEAD OF is associated with views.

26) Define “correlated subqueries”.

A ‘correlated subquery’ is a sort of sub query but correlated subquery is reliant on another query for a value that is returned. In case of execution, the sub query is executed first and then the correlated query.

27) Define Data Warehousing.

Storage and access of data from the central location in order to take some strategic decision is called Data Warehousing. Enterprise management is used for managing the information whose framework is known as Data Warehousing.

28) Define Join and enlist its types.

Joins help in explaining the relation between different tables. They also enable you to select data with relation to data in another table.

The various types are:

- **INNER JOINS:** Blank rows are left in the middle while more than equal to two tables are joined.
- **OUTER JOINS:** Divided into Left Outer Join and Right Outer Join. Blank rows are left at the specified side by joining tables in other side.

Other joins are **CROSS JOINS**, **NATURAL JOINS**, **EQUI JOIN** and **NON-EQUI JOIN**.

29) What do you mean by Index hunting?

Indexes help in improving the speed as well as the query performance of database. The procedure of boosting the collection of indexes is named as Index hunting.

30) How does Index hunting help in improving query performance?

Index hunting helps in improving the speed as well as the query performance of database. The followed measures are achieved to do that:

- The query optimizer is used to coordinate the study of queries with the workload and the best use of queries suggested based on this.
- Index, query distribution along with their performance is observed to check the effect.
- Tuning databases to a small collection of problem queries is also recommended.

31) Enlist the disadvantages of query.

The disadvantages of query are:

- No indexes
- Stored procedures are excessively compiled.
- Triggers and procedures are without SET NOCOUNT ON.
- Complicated joins making up inadequately written query.
- Cursors and temporary tables showcase a bad presentation.

32) Enlist ways to efficiently code transactions.

Ways to efficiently code transactions:

- User input should not be allowed while transactions.
- While browsing, transactions must not be opened of data.
- Transactions must be kept as small as possible.
- Lower transaction segregation levels.
- Least information of data must be accessed while transacting.

33) What is Executive Plan?

Executive plan can be defined as:

- SQL Server caches collected procedure or the plan of query execution and used thereafter by subsequent calls.
- An important feature in relation to performance enhancement.
- Data execution plan can be viewed textually or graphically.

34) Define B-trees.

A data structure in the form of tree which stores sorted data and searches, insertions, sequential access and deletions are allowed in logarithmic time.

35) Differentiate Table Scan from Index Scan.

Iterating over all the table rows is called Table Scan while iterating over all the index items is defined as Index Scan.

36) What do you mean by Fill Factor concept with respect to indexes?

Fill Factor can be defined as being that value which defines the percentage of left space on every leaf-level page that is to be packed with data. 100 is the default value of Fill Factor.

37) Define Fragmentation.

Fragmentation can be defined as a database feature of server that promotes control on data which is stored at table level by the user.

38) Differentiate Nested Loop, Hash Join and Merge Join.

Nested loop (loop over loop)

An outer loop within an inner loop is formed consisting of fewer entries and then for individual entry, inner loop is individually processed.

E.g.

- Select col1.*, col2.* from coll, col2 where coll.col1=col2.col2;

It's processing takes place in this way:

For i in (select * from col1) loop

For j in (select * from col2 where col2=i.col1) loop

Results are displayed;

End of the loop;

End of the loop;

The Steps of nested loop are:

- Identify outer (driving) table
- Assign inner (driven) table to outer table.
- For every row of outer table, access the rows of inner table.

Nested Loops is executed from the inner to the outer as:

- outer_loop
- inner_loop
- Hash join

While joining large tables, the use of Hash Join is preferred.

Algorithm of Hash Join is divided into:

- Build: It is a hash table having in-memory which is present on the smaller table.
- Probe: this hash value of the hash table is applicable for each second row element.
- Sort merge join

Two independent sources of data are joined in sort merge join. Their performance is better as compared to nested loop when the data volume is big enough but it is not good as hash joins generally. The full operation can be divided into parts of two:

Sort join operation :

Get first row R1 from input1

Get first row R2 from input2.

Merge join operation:

'while' is not present at either loop's end.
if R1 joins with R2
next row is got R2 from the input 2
return (R1, R2)
else if R1 < style="">> next row is got from R1 from input 1
else
next row is got from R2 from input 2
end of the loop

39) What is Database partitioning?

Division of logical database into independent complete units for improving its management, availability and performance is called Database partitioning.

40) Explain the importance of partitioning.

Splitting of one table which is large into smaller database entities logically is called database partitioning. Its benefits are:

- To improve query performance in situations dramatically when mostly rows which are heavily accessed are in one partition.
- Accessing large parts of a single partition
- Slower and cheaper storage media can be used for data which is seldom used.

41) Define Database system.

DBMS along with database is called Database system.

42) What do you mean by Query Evaluation Engine?

Query Evaluation Engine executes the low-level instructions that are generated by the compiler.

43) Define DDL Interpreter.

DDL statements are interpreted and recorded in tables called metadata.

44) Define Atomicity and Aggregation.

Atomicity: It's an all or none concept which enables the user to be assured of incomplete transactions to be taken care of. The actions involving incomplete transactions are left undone in DBMS.

Aggregation: The collected entities and their relationship are aggregated in this model. It is mainly used in expressing relationships within relationships.

45) Enlist the various transaction phases.

The various transaction phases are:

- Analysis Phase.
- Redo Phase
- Undo Phase

46) Define Object-oriented model.

Compilations of objects make up this model in which values are stored within instance variables which is inside the object. The object itself comprises bodies of object for its operation which are called methods. Objects containing same kind of variables and methods are called classes.

47) Define Entity.

It can be defined as being a 'thing' with an independent existence in the real world.

48) What do you mean by Entity type?

A set of entries having similar attributes are entity types.

49) Define Entity Set.

Compilation of all entries of any particular type of entry in the database is called Entity Set.

50) What do you mean by Entity type extension?

Compilation of similar entity types into one particular type which is grouped together as an entity set.