

## **DBMS basic INTERVIEW QUESTION**

### **1) What is DBMS**

**Ans-** DBMS stands for Database Management System, is a set of applications or programs that enable users to create and maintain a database

### **2) What is rdbms?**

**Ans-**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.

E.g mysql, oracle db etc.

### **3) What is database?**

**Ans-** A Database is an organized, consistent, and logical collection of data that can easily be updated, accessed, and managed.

### **4) Dbms vs file system?**

## **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



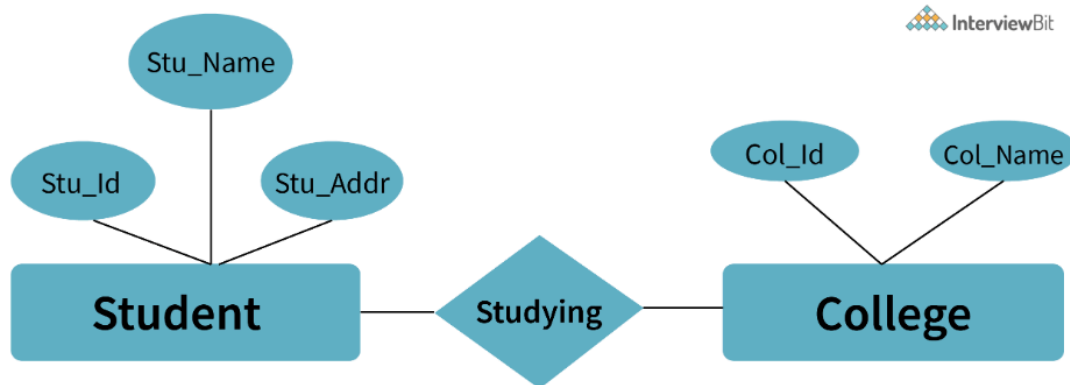
- **Atomicity:** This property reflects the concept of either executing the whole query or executing nothing at all, which implies that if an update occurs in a database then that update should either be reflected in the whole database or should not be reflected at all.
- **Consistency:** This property ensures that the data remains consistent before and after a transaction in a database.
- **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.
-

- **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.

### Intermediate question:-

#### 1. 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.



- **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

## **2. 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

## **3. 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

#### **4. Explain the difference between the DELETE and TRUNCATE command in a DBMS.**

DELETE command: this command is needed to delete rows from a table based on the condition provided by the WHERE clause.

- It deletes only the rows which are specified by the WHERE clause.
- It can be rolled back if required.
- It maintains a log to lock the row of the table before deleting it and hence it's slow.

TRUNCATE command: this command is needed to remove complete data from a table in a database. It is like a DELETE command which has no WHERE clause.

- It removes complete data from a table in a database.
- It can't 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

#### **8. What is Data Warehousing?**

The process of collecting, extracting, transforming, and loading data from multiple sources and storing them in one database is known as data warehousing.

##### **1. 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 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.

## **2. 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.
- **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.