1. What do you understand By Database

Ans - • DBMS stands for Data Base Management System.

• Database is a collection of inter-related data and Management System is a set of programs to store and retrieve those data.

• DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

• For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

2. What is Normalization?

Ans - • Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.

• Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.

• Most Commonly used normal forms :-

First Normal Form:

• First normal form(1NF) Second normal form(2NF) Third normal form(3NF) Boyce & Code normal form (BCNF)

• If a relation contain composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute.

• A relation is in first normal form if every attribute in that relation is singled valued attribute.

Second Normal Form:

• To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency.

• relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

• Partial Dependency – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

Third Normal Form:

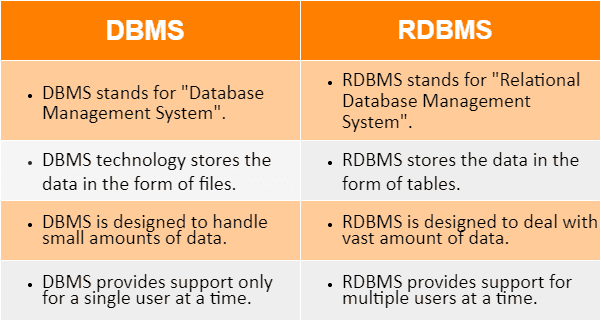
• A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

• A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y X is a super key. Y is a prime attribute (each element of Y is part of some candidate key).

• Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

3. What is Difference between DBMS and RDBMS?

Ans –



4. What is MF Cod Rule of RDBMS Systems?

Ans - **Rule 1: The Information Rule**

All information, whether it is user information or metadata, that is stored in a database must be entered as a value in a cell of a table. It is said that everything within the database is organized in a table layout.

**Rule 2: The Guaranteed Access Rule**

Each data element is guaranteed to be accessible logically with a combination of the table name, primary key (row value), and attribute name (column value).

**Rule 3: Systematic Treatment of NULL Values**

Every Null value in a database must be given a systematic and uniform treatment.

**Rule 4: Active Online Catalog Rule**

The database catalog, which contains metadata about the database, must be stored and accessed using the same relational database management system.

**Rule 5: The Comprehensive Data Sublanguage Rule**

A crucial component of any efficient database system is its ability to offer an easily understandable data manipulation language ([DML](https://www.geeksforgeeks.org/dml-full-form/)) that facilitates defining, querying, and modifying information within the database.

**Rule 6: The View Updating Rule**

All views that are theoretically updatable must also be updatable by the system.

**Rule 7: High-level Insert, Update, and Delete**

A successful database system must possess the feature of facilitating high-level insertions, updates, and deletions that can grant users the ability to conduct these operations with ease through a single query.

**Rule 8: Physical Data Independence**

Application programs and activities should remain unaffected when changes are made to the physical storage structures or methods.

**Rule 9: Logical Data Independence**

Application programs and activities should remain unaffected when changes are made to the logical structure of the data, such as adding or modifying tables.

**Rule 10: Integrity Independence**

Integrity constraints should be specified separately from application programs and stored in the catalog. They should be automatically enforced by the database system.

**Rule 11: Distribution Independence**

The distribution of data across multiple locations should be invisible to users, and the database system should handle the distribution transparently.

**Rule 12: Non-Subversion Rule**

If the interface of the system is providing access to low-level records, then the interface must not be able to damage the system and bypass security and integrity constraints.

5. What do you understand By Data Redundancy?

Ans - Data redundancy means the occurrence of duplicate copies of similar data. It is done intentionally to keep the same piece of data at different places, or it occurs accidentally.

Sometimes, it is done on purpose for recovery or backup of data, faster access of data, or updating data easily. Redundant data costs extra money, demands higher storage capacity, and requires extra effort to keep all the files up to date.

Advantages of data redundancy in DBMS:

* **Provides Data Security:** Data redundancy can enhance data security as it is difficult for cyber attackers to attack data that are in different locations.
* **Provides Data Reliability:** Reliable data improves accuracy because organizations can check and confirm whether data is correct.
* **Create Data Backup:** Data redundancy helps in backing up the data.

6. What is DDL Interpreter?

Ans - A DDL interpreter performs the task of interpreting the DDL statements and recording their definitions in tables or data dictionaries that contain metadata. It helps in defining the logical design of the schemas.

DDL is a standardized language with commands to define the storage groups (stogroups), different structures and objects in a database. DDL statements create, modify and remove database objects, such as tables, indexes and stogroups. DDL is also used in a generic sense to refer to any language that describes data.

* CREATE – to create database and its objects like (table, index, views, store procedure, function and triggers).
* ALTER – alters the structure of the existing database.
* DROP – delete objects from the database.
* TRUNCATE – remove all records from a table; also, all spaces allocated for the records are removed.
* COMMENT – add comments to the data dictionary.
* RENAME – rename an object.

7. What is DML Compiler in SQL?

Ans - DML is short name of Data Manipulation Language which deals with data manipulation, and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in database.

* SELECT – retrieve data from one or more tables.
* INSERT – insert data into a table.
* UPDATE – updates existing data within a table.
* DELETE – delete all records from a table.
* MERGE – UPSERT operation (insert or update)

8. What is SQL Key Constraints writing an Example of SQL Key Constraints

Ans - Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

|  |  |
| --- | --- |
| Constraint | Description |
| NOT NULL | values cannot be null |
| UNIQUE | values cannot match any older value |
| PRIMARY KEY | used to uniquely identify a row |
| FOREIGN KEY | references a row in another table |
| CHECK | validates condition for new value |
| DEFAULT | set default value if not passed |
| CREATE INDEX | used to speedup the read process |

1. SQL NOT NULL on CREATE TABLE

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255) NOT NULL,

Age int

);

1. SQL UNIQUE Constraint

CREATE TABLE Persons (

ID int NOT NULL UNIQUE,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int

);

1. SQL PRIMARY KEY Constraint

CREATE TABLE Persons (

ID int NOT NULL PRIMARY KEY,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int

);

1. SQL FOREIGN KEY on CREATE TABLE

CREATE TABLE Orders (

OrderID int NOT NULL,

OrderNumber int NOT NULL,

PersonID int,

PRIMARY KEY (OrderID),

FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)

);

1. SQL CHECK Constraint

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int CHECK (Age>=18)

);

1. SQL DEFAULT Constraint

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

City varchar(255) DEFAULT 'Ahmedabad'

);

1. SQL CREATE INDEX Statement

CREATE INDEX index\_name

ON table\_name (column1, column2, ...);

Example:

CREATE INDEX idx\_lastname

ON Persons (LastName);

1. What is save Point? How to create a save Point write a Query?

* Savepoint is a command in SQL that is used with the rollback command.
* It is a command in Transaction Control Language that is used to mark the transaction in a table.
* Consider you are making a very long table, and you want to roll back only to a certain position in a table then; this can be achieved using the savepoint.
* If you made a transaction in a table, you could mark the transaction as a certain name, and later on, if you want to roll back to that point, you can do it easily by using the transaction's name.
* Savepoint is helpful when we want to roll back only a small part of a table and not the whole table. In simple words, we can say savepoint is a bookmark in SQL.

Syntax for Savepoint command:

**SAVEPOINT SAVEPOINT\_NAME;**

This command is used only in the creation of SAVEPOINT among all the transactions. In general ROLLBACK is used to undo a group of transactions.

Syntax for rolling back to Savepoint command:

**ROLLBACK TO SAVEPOINT\_NAME;**

you can ROLLBACK to any SAVEPOINT at any time to return the appropriate data to its original state.

Example: Delete those records from the table which have age = 20 and then ROLLBACK the changes in the database by keeping Savepoints.

Queries:

SAVEPOINT SP1; //Savepoint created.

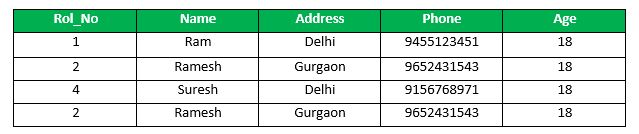
DELETE FROM Student WHERE AGE = 20; //deleted

SAVEPOINT SP2; //Savepoint created.

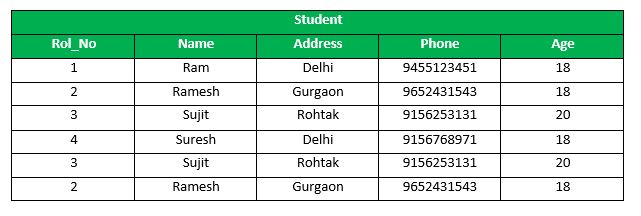
Here SP1 is first SAVEPOINT created before deletion

.In this example one deletion have taken place. After deletion again SAVEPOINT SP2 is created.

OUTPUT :



Deletion have been taken place, let us assume that you have changed your mind and decided to ROLLBACK to the SAVEPOINT that you identified as SP1 which is before deletion. deletion is undone by this statement ,

ROLLBACK TO SP1; //Rollback completed.   


10.What is trigger and how to create a Trigger in SQL?

Ans - A trigger is a set of SQL statements that reside in system memory with unique names. It is a specialized category of stored procedure that is called automatically when a database server event occurs. Each trigger is always associated with a table.

A trigger is called a special procedure because it cannot be called directly like a stored procedure. The key distinction between the trigger and procedure is that a trigger is called automatically when a data modification event occurs against a table. A stored procedure, on the other hand, must be invoked directly.