MODULE – 5 DATABASE

1. What do you understand By Database ?

- A database is an organized collection of structured information, or data, typically stored electronically in computer system.
- It is controlled by a database management system (DBMS).
- Data within databases is typically modeled in rows and columns in a series of table to make processing and data querying efficient.
- The data can be access easily accessed, managed, modified, update, controlled, and organized.
- Most databases use Structured Query Language (SQL) for writing and quering data.

2. What is Normalization?

- Normalization is the process of oragnizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations.
- Database normalization will make your job a lot esier from the user who accesses table to the database administrator who is responsible for overall management of every object in the database.
- Database normal forms:-
- 1 NF (first normal form)
- 2 NF (second normal form)
- 3 NF (third normal form)
- 4 NF (forth normal form)
- BCNF (Boyce-codd normal form)

3. What is Difference between DBMS and RDBMS ?

- ### DBMS :-

- DBMS stores data as a file.
- Data elements need to access individually.
- No relationship between data
- Normalization is not present.
- DBMS dose not support distributed database.
- It deals with small quntity of data.
- Security is less.

- ### RDBMS :-

- RDBMS stores data in tabular form.
- Multiple data elements can be accessed at the same time.
- Data stored in the form of tables which are related to each other.
- Normalization is present.
- RDBMS supports distributed database.
- It deals with large amount of data.
- More security.

4. What is MF Cod Rule of RDBMS Systems?

- **Codd's rule in DBMS also known as Codd's 12 rules/commandments is** a set of thirteen rules (numbered 0 to 12) that define a database to be a correct Relational Database Management System (RDBMS)

Rule 1: The *information rule*:

- All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.
-

Rule 2: The *guaranteed access rule*:

- Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

Rule 3: *Systematic treatment of null values*:

- Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.

Rule 4: *Dynamic online catalog based on the relational model*:

- The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

Rule 5: The *comprehensive data sublanguage rule*:

A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items:

- Data definition.
- View definition.
- Data manipulation (interactive and by program).
- Integrity constraints.
- Authorization.
- Transaction boundaries (begin, commit and rollback).

Rule 6: The *view updating rule*:

- All views that are theoretically updatable are also updatable by the system.

Rule 7: Relational Operations Rule / *Possible for high-level insert, update, and delete*:

- The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.

Rule 8: *Physical data independence*:

- Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.

-

5. What is MF Cod Rule of RDBMS Systems?

- Codd's rule in DBMS also known as Codd's 12 rules/commandments is a set of thirteen rules (numbered 0 to 12) that define a database to be a correct Relational Database Management System (RDBMS).

Rule 1: The information rule:

- All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.

-

Rule 2: The guaranteed access rule:

- Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

Rule 3: Systematic treatment of null values:

- Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.

Rule 4: Dynamic online catalog based on the relational model:

- The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

Rule 5: The comprehensive data sublanguage rule:

A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items:

- Data definition.

- View definition.

- Data manipulation (interactive and by program).

- Integrity constraints.

- Authorization.

- Transaction boundaries (begin, commit and rollback).

Rule 6: The view updating rule:

- All views that are theoretically updatable are also updatable by the system.

Rule 7: Relational Operations Rule / Possible for high-level insert, update, and delete:

- The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.

Rule 8: Physical data independence:

- Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.

Rule 9: Logical data independence:

- Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.

Rule 10: Integrity independence:

- Integrity constraints specific to a particular relational data base must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.

Rule 11: Distribution independence:

- The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only.

Rule 12: The nonsubversion rule:

If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).

6. What do you understand By Data Redundancy?

- Data Redundancy refers to the prectice of keeping data in two or more places within a Database or DataStorage system. Data Redundancy ensures an organization can provide continued opretions or services in the event something happens to its data .

- For Example , in the case of data corruption or data loss.

- The Concept applies to areas such as databases, computer memory and file storage systems.

- Data redundancy can occur within an organization intentially or accidentally . If done intentionally, the same data is kept in different locations with the organizing making a consious efforts to protect it and ensure its consistancy.

- This data is often used for backups or disaster recovery.

7. What is DDL Interpreter?

- DDL Interpreter DDL expands to Data Definition Language .

- DDL Interpreter as the name suggests interprets the DDL statements such as schema definition statements like create, delete etc.

8. What is DML compiler in SQL ?

- DML stands for Data Manipulation Language .

- It is a computer programming Language used for adding , deleting and modifying data in database.

9. What is SQL key Constraints writing an Example of SQL Key Constraints.

- SQL constraints are used to specify rules for the data in a table.

- Constraints are used to limit the type of data that can go into table.

- The Primary Key Constraint uniquely identifies each record in a table.

- EX

- CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255) NOT NULL,

Age int

);

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

- Save Point is a command in SQL that is used with the rollback command.

- It is a command in transection Control Language that is used to mark the transection in table.

- SavePoint is helpful when we want to roll back only a small part of a table and not the whole table.

- SavePoint is a bookmark in SQL

- SSAVEPOINT SP1;

//Savepoint created.

DELETE FROM Student WHERE AGE = 20;

//deleted

SAVEPOINT SP2;

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

- A trigger is a Stored Procedure in a DataBase that Automatically invokes whenever a special event in the database occurs.

- For Example, a trigger can be invoked when a row is inserted into a specified table or when table columns are updated in simple words a trigger is a collection of SQL statements with perticular names that are stored in system memory.

- Create trigger deep

- On emp

- For

- Insert,update ,delete

- As

- Print ‘you can not insert,update and delete this table I’

- Rollback;

-