**MODULE: 5**

**Database**

**Basic of Database**

1. **What do you understand by Database.**

**Ans. Database** is an **organized collection** of **structured information or data**, typically stored and accessed electronically from a computer system. It can contain any type of data, including **words, numbers, images, videos, and files**.

You can organize data into **tables, rows, columns, and index** it to make it easier to find relevant information.

The main purpose of the database is to **operate a large amount of information** by **storing, retrieving, and managing data.**

1. **What is Normalization?**

**Ans. Normalization** is a process used in **database design** to **organize tables and minimize redundancy and dependency** by organizing fields and table of a database.

It typically involves organizing data into **multiple tables and establishing relationships** between these tables using **keys (e.g., primary keys, foreign keys).**

It follows a **series of normalization forms** (e.g., First Normal Form (**1NF**), Second Normal Form (**2NF**), Third Normal Form (**3NF**)) to systematically reduce redundancy and dependency, etc.

**BCNF** stands for **Boyce-Codd Normal Form**, which is a higher level of database normalization than the Third Normal Form (**3NF**).

1. **What is Difference between DBMS and RDBMS?**

**Ans.**

|  |  |
| --- | --- |
| DBMS | RDBMS |
| It is a software system that allows users to define, create, maintain, and control access to databases. | It is a type of DBMS that manages data in a structured format using tables, rows, and columns. |
| It provides mechanisms for storing, retrieving, and manipulating data in a database. | It also enforces the relational model principles, which include defining relationships between tables using keys. |
| It can store data in different ways, depending on the model it supports (e.g., files, documents, key-value pairs). | It stores data in a structured manner within tables, with each table having a defined schema (set of columns and their data types). |
| It may use different query languages, depending on the data model (e.g., SQL, NoSQL query languages). | It primarily uses SQL (Structured Query Language) for querying and manipulating data. |
| It may or may not support normalization techniques for organizing data to minimize redundancy and dependency. | It typically supports normalization up to higher forms such as BCNF (Boyce-Codd Normal Form) to ensure efficient data organization and integrity. |
| Ex. MongoDB (NoSQL DBMS), SQLite, and Microsoft Access. | Ex. MySQL, PostgreSQL, Oracle Database, and SQL Server. |

1. **What is MF Cod Rule of RDBMS Systems?**

**Ans.** A DB that solely contains a relational data model cannot be called a Relational DB Management System or RDBMS. Some rules determine if a DB is the correct RDBMS. **Dr Edgar F. Codd**, who has extensive knowledge on the **DB system’s Relational Model**, proposed these principles in 1985.

A DB that follows the rule is referred to as a real relational DB management system (RDBMS).

**Codd’s rules are a set of rules that are widely used in relational DB’s**.

The Rules are following:

Rule 0: The Foundation Rule

Rule 1: The Information Rule

Rule 2: The Guaranteed Access Rule

Rule 3: The Systematic Treatment of Null Values

Rule 4: The Dynamic/Active Online Catalog on the basis of the Relational Model

Rule 5: The Comprehensive Data SubLanguage Rule

Rule 6: The View Updating Rule

Rule 7: The Relational Level Operation (or High-Level Insert, Delete, and Update) Rule

Rule 8: The Physical Data Independence Rule

Rule 9: The Logical Data Independence Rule

Rule 10: The Integrity Independence Rule

Rule 11: The Distribution Independence Rule

Rule 12: The Non-Subversion Rule

1. **What do you understand By Data Redundancy?**

**Ans. Redundancy** means having multiple copies of the same data in the database.

This problem arises when a database is not normalized. Suppose a table of student details attributes is: student ID, student name, college name, college rank, and course opted.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student\_ID** | **Name** | **Contact** | **College** | **Course** | **Rank** |
| 100 | Himanshu | 7300934851 | GEU | B.Tech | 1 |
| 101 | Ankit | 7900734858 | GEU | B.Tech | 1 |
| 102 | Ayush | 7300936759 | GEU | B.Tech | 1 |
| 103 | Ravi | 7300901556 | GEU | B.Tech | 1 |

It can be observed that values of attribute college name, college rank, and course are being repeated which can lead to problems.

Problems caused due to redundancy are:

1. Insertion anomaly, 2) Deletion anomaly, 3) Updation anomaly
2. **What is DDL Interpreter?**

**Ans. Data Definition Language (DDL)** statements are used to define the structure of the database schema.

This includes creating and modifying database objects such as tables, indexes, views, and constraints.

Examples of DDL statements include **CREATE, ALTER, DROP, TRUNCATE**, etc.

Here’s some common DDL statements and their purposes:

**CREATE:** Used to create new database objects like tables, views, indexes, etc.

**ALTER:** Modifies the structure of existing database objects.

**DROP:** Deletes database objects (tables, views, indexes, etc.) from the database.

**TRUNCATE:** Removes all records from a table without removing the table structure itself.

**COMMENT:** Adds comments to the data dictionary about database objects.

1. **What is DML Compiler in SQL?**

**Ans. Data Manipulation Language (DML)** statements in SQL are used **to retrieve, insert, update, and delete data from a database**. These statements allow users and applications to interact with the data stored in tables within the database.

Examples of DML statements include **INSERT, UPDATE, DELETE, etc**.

Here’s a brief overview of some common DML statements and their purposes:

**INSERT**: Adds new rows of data into a table.

**UPDATE**: Modifies existing data within a table.

**DELETE**: Removes rows of data from a table.

1. **What is SQL Key Constraints writing an Example of SQL Key Constraints.**

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

* They 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.
* Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.
* The following constraints are commonly used in SQL:

**NOT NULL** - Ensures that a column cannot have a NULL value

**UNIQUE** - Ensures that all values in a column are different

**PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

**FOREIGN KEY** - Prevents actions that would destroy links between tables

**CHECK** - Ensures that the values in a column satisfies a specific condition

**DEFAULT** - Sets a default value for a column if no value is specified

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

**Ans. SAVEPOINT** creates points within the groups of transactions in which to **ROLLBACK.** Itis a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.

**Syntax:**

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.

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

**Ans. 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 specific table columns are updated.

**In simple words, a trigger is a collection of SQL statements with particular names that are stored in system memory**.

It belongs to a specific class of stored procedures that are automatically invoked in response to database server events.

Every trigger has a table attached to it.Because a trigger cannot be called directly, unlike a stored procedure, it is referred to as a special procedure.

A trigger is automatically called whenever a data modification event against a table takes place, which is the main distinction between a trigger and a procedure.

On the other hand, a stored procedure must be called directly.

**Syntax:**

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]

The following are the key differences between triggers and stored procedures:

1. Triggers cannot be manually invoked or executed.
2. There is no chance that triggers will receive parameters.
3. A transaction cannot be committed or rolled back inside a trigger.