**SE - Database - Full stack and Back end**

-What do you understand By Database.

A database is a structured collection of data that is organized and stored in a way that allows for efficient retrieval, updating, and management of that data. It serves as a repository for information that can be easily accessed, managed, and updated. Databases are a crucial component in modern computing and are used in a wide range of applications, including websites, business systems, scientific research, and more.

-What is Normalization?

Normalization is a process used in database design to organize data in a way that reduces redundancy and dependency. The primary goal of normalization is to structure a relational database in such a way that it minimizes data redundancy and avoids certain types of data anomalies when performing operations like insertion, deletion, and updates.

-What is Difference between DBMS and RDBMS?

DBMS (Database Management System) and RDBMS (Relational Database Management System) are terms often used in the context of databases, but they refer to slightly different concepts:

-> DBMS (Database Management System):

A DBMS is a software application or a set of programs that enable the creation, organization, retrieval, and management of databases.

It provides an interface for interacting with the database, allowing users and applications to store, retrieve, and manipulate data.

DBMS may or may not enforce relationships between tables, and it doesn't necessarily follow the principles of the relational model.

-> RDBMS (Relational Database Management System):

RDBMS is a specific type of DBMS that is based on the relational model of data.

In an RDBMS, data is organized into tables with rows and columns, and relationships between tables are established using keys.

RDBMS follows principles like normalization and uses SQL (Structured Query Language) for defining, querying, and manipulating the data.

Examples of RDBMS include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

-What is MF Cod Rule of RDBMS Systems?

It seems there might be a confusion or a miscommunication in your question. As of my last knowledge update in September 2021, there isn't a widely recognized concept or rule in the field of relational database management systems (RDBMS) known as "MF Cod Rule." It's possible that there may be a misunderstanding or a term that has emerged after my last update.

If "MF Cod Rule" refers to a specific concept, rule, or technology introduced after September 2021, I would recommend checking the latest resources, documentation, or authoritative databases related to databases and RDBMS for the most up-to-date information.

If you have additional context or details about what "MF Cod Rule" might refer to, please provide more information, and I'll do my best to help clarify or provide relevant information based on the knowledge available up to my last update.

- What do you understand By Data Redundancy?

Data redundancy refers to the storage of the same data in multiple places within a database or across multiple databases. It occurs when identical or similar data is duplicated unnecessarily, which can lead to several issues and inefficiencies in a database system.

-What is DDL Interpreter?

DDL (Data Definition Language) Interpreter refers to a component within a Database Management System (DBMS) responsible for processing and executing Data Definition Language statements. DDL is a subset of SQL (Structured Query Language) used to define and manage the structure of a database. DDL statements are used to create, modify, and delete database objects such as tables, indexes, and constraints.

-What is DML Compiler in SQL?

DML is a subset of SQL (Structured Query Language) used to manipulate data stored in a database. DML statements include operations such as SELECT (retrieve data), INSERT (insert new records), UPDATE (modify existing records), and DELETE (remove records). These statements are not compiled in the traditional sense; they are executed directly when issued.

-What is SQL Key Constraints writing an Example of SQL Key Constraints?

SQL, key constraints are rules that define how values in a column or a set of columns should behave concerning uniqueness and relationships between tables. There are primarily two types of key constraints: Primary Key and Foreign Key.

-> Primary Key Constraint:

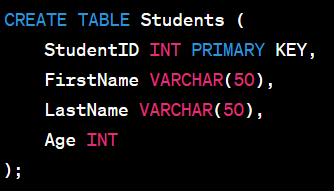
A primary key uniquely identifies each record in a table. It must contain unique values, and it cannot have NULL values.

There can be only one primary key in a table.

Primary keys are defined using the PRIMARY KEY constraint.

Example of creating a table with a primary key:

Ex:



In this example, the StudentID column is defined as the primary key. It means that each StudentID must be unique, and it cannot be NULL.

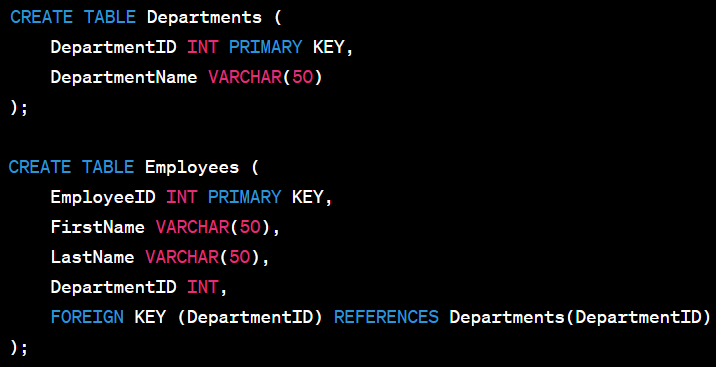
-> Foreign Key Constraint:

A foreign key establishes a link between two tables. It is a field in one table that refers to the primary key in another table.

The foreign key ensures referential integrity, meaning that values in the foreign key column must match values in the referenced primary key column.

Foreign keys are defined using the FOREIGN KEY constraint.

Example of creating two tables with a foreign key relationship:



In this example, the Employees table has a foreign key (DepartmentID) that references the primary key (DepartmentID) in the Departments table. This establishes a relationship between the two tables based on the department ID.

These key constraints play a crucial role in maintaining data integrity and ensuring that relationships between tables are well-defined and enforced.

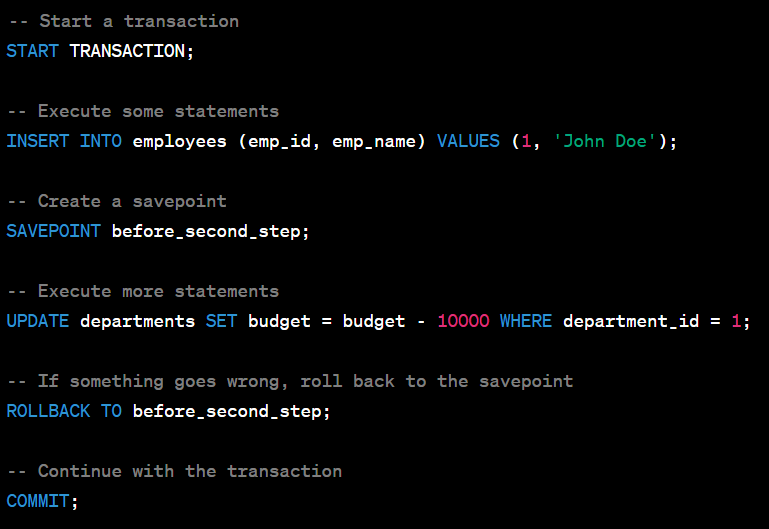
-What is save Point? How to create a save Point write a Query?

Savepoint is a point within a transaction where you can mark the current state and later roll back to that state if needed. Savepoints are useful in scenarios where you want to partially roll back a transaction without affecting the entire transaction.

Syntax:



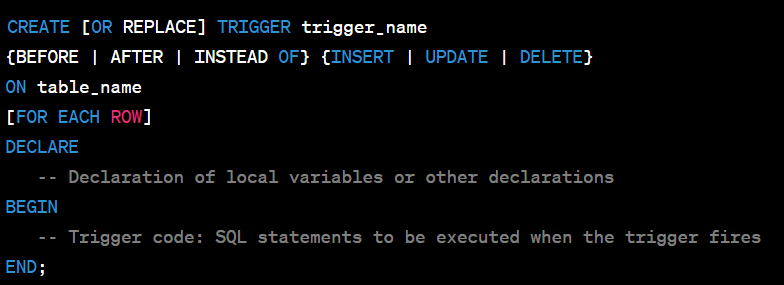
Example:



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

Trigger is a set of instructions that are automatically executed or "triggered" in response to certain events on a particular table or view. These events typically include data manipulation language (DML) events like INSERT, UPDATE, or DELETE operations. Triggers are often used to enforce business rules, perform logging, or maintain data integrity.

Syntax:



Example:

