**MODULE – 4 (Database)**

**Q. What do you understand by Database?**

**Ans.** A structured set of data held in a computer, especially one that is accessible in various ways.

**Q. What is Normalization?**

**Ans.**

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce redundancy from the database table.

### Q. What is the Difference between DBMS and RDBMS?

**Ans.** DBMS stands for Database Management System and RDBMS is the acronym for the Relational Database Management system. In DBMS, the data is stored as a file, whereas in RDBMS, data is stored in the form of tables.

**Q. What is E.F. Codd Rule of RDBMS System?**

**Ans.** E.F Codd was a Computer Scientist who invented the Relational model for Database management. Based on relational model, the Relational database was created. Codd proposed 13 rules popularly known as Codd's 12 rules to test DBMS's concept against his relational model. Codd's rule actualy define what quality a DBMS requires in order to become a Relational Database Management System(RDBMS). Till now, there is hardly any commercial product that follows all the 13 Codd's rules. Even Oracle follows only eight and half(8.5) out of 13.

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

**Ans.** Redundancy in DBMS is having several copies of the same data in the database, for example, storing the complete details of the department such as department\_id, department\_name, and department\_head repeatedly in every student record. Redundancy may cause inconsistency in data when they are not properly updated.

**Q. What is DDL Interpreter?**

**Ans.** 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. The result of this interpretation is a set of a table that contains the meta-data which is stored in the data dictionary.

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

Ans. DML Compiler DML expands to Data Manipulation Language in DBMS. DML Compiler again as the name suggests compiles(or translates) the DML statements such as select, update and delete statements into low-level instructions which is nothing but the machine-readable object code to make it executable.

**Q. What is SQL Key Constraints. Write an Example of SQL Key Constraints.**

Ans. 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 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](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that the values in a column satisfies a specific condition
* [DEFAULT](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column if no value is specified
* [CREATE INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly

***EXAMPLE:***

1. **NOT NULL CONSTRAINT**

CREATE TABLE Colleges (

college\_id INT NOT NULL,

college\_code VARCHAR(20) NOT NULL,

college\_name VARCHAR(50)

);

1. **UNIQUE CONSTRAINT**

CREATE TABLE Colleges (

college\_id INT NOT NULL UNIQUE,

college\_code VARCHAR(20) UNIQUE,

college\_name VARCHAR(50)

);

1. **PRIMARY KEY CONSTRAINT**

CREATE TABLE Colleges (

college\_id INT PRIMARY KEY,

college\_code VARCHAR(20) NOT NULL,

college\_name VARCHAR(50)

);

1. **FOREIGN KEY CONSTRAINT**

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id int REFERENCES Customers(id)

);

1. **CHECK CONSTRAINT**

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

amount int CHECK (amount >= 100)

);

1. **DEFAULT CONSTRAINT**

CREATE TABLE College (

college\_id INT PRIMARY KEY,

college\_code VARCHAR(20),

college\_country VARCHAR(20) DEFAULT 'US'

);

1. **CREATE INDEX CONSTRAINT**

-- create table

CREATE TABLE Colleges (

college\_id INT PRIMARY KEY,

college\_code VARCHAR(20) NOT NULL,

college\_name VARCHAR(50)

);

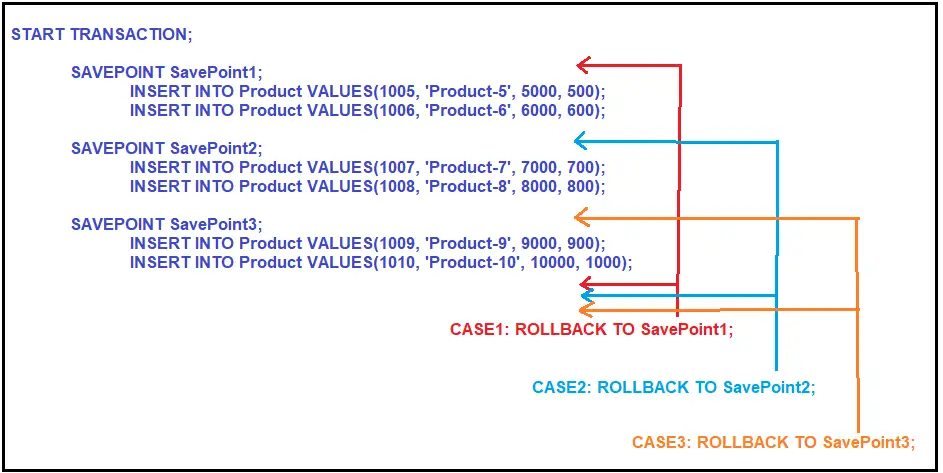
-- create index

CREATE INDEX college\_index

ON Colleges(college\_code);

**Q. What is save Point? How to create a save Point. Write a Query?**

**Ans.** The SAVEPOINT is used for dividing (or) breaking a transaction into multiple units so that the user has a chance of roll backing the transaction up to a specified point. That means using Save Point we can roll back a part of a transaction instead of the entire transaction.



**CASE1:ROLLBACKTOSavePoint1**  
When we execute the above Rollback command, it will roll back the statements which are starting from SavePoint1, and before the rollback statement. That means in our example, it will roll back all the 6 Insert statements.

**CASE2:ROLLBACKTOSavePoint2**  
When we execute the above Rollback statement, then it will roll back the statements which are starting from SavePoint2 and before the Rollback statement. That means, in this case, it will roll back 4 Insert statements.

**CASE3:ROLLBACKTOSavePoint3**  
When we execute the above Rollback statement, it will roll back the statements which are present after the SavePoint3 and before the Rollback Command. That means in this case, it will roll back two insert statements.

Now, I hope you understand the concept of SAVEPOINT and also understand how to perform Partial Rollback in MySQL using SAVEPOINT.

**Q. 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.

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.

**Syntax:**

* *create trigger [trigger\_name]*
* *[before | after]*
* *{insert | update | delete}*
* *on [table\_name]*
* *[for each row]*
* *[trigger\_body]*

**Explanation of Syntax:**

1. Create trigger [trigger\_name]: Creates or replaces an existing trigger with the trigger\_name.
2. [before | after]: This specifies when the trigger will be executed.
3. {insert | update | delete}: This specifies the DML operation.
4. On [table\_name]: This specifies the name of the table associated with the trigger.
5. [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each affected row.
6. [trigger\_body]: This provides the operation to be performed as the trigger is fired.