Modifying Data

- Modifying Data mainly comes under Data Manipulation Language in SQL. There are mainly four types of operations we can perform on a table/database.

1. INSERT
2. UPDATE
3. DELETE
4. RENAME

For explaining all of these operations we will use the below-mentioned table- (Assume it to be empty at the beginning)

Table Name - **Student**

Student_id	Student_name	Batch	Course

- **INSERT**

To perform the INSERT operation, we have to use the INSERT INTO statement. There are two ways to use this statement as mentioned below.

Mention both the column names and the values you want to insert -

```
INSERT INTO Student (Student_id, Student_name, Batch, Course)
VALUES (1, Ojasv, 2018, SQL);
```

Output:-

Student_id	Student_name	Batch	Course
1	Ojasv	2018	SQL

Mention only the values of for the columns. Make sure the order of

occurrence of values is the same as the order of columns -

INSERT INTO Student
VALUES (2,Kuldeep, 2018, MongoDB);

Output:-

Student_id	Student_name	Batch	Course
1	Ojasv	2018	SQL
2	Kuldeep	2018	MongoDB

For storing date and time fields in SQL we have the following date fields

1. DATE - format YYYY-MM-DD
2. YEAR - format YYYY or YY
3. DATETIME - format: YYYY-MM-DD HH:MI: SS
4. TIMESTAMP - format: YYYY-MM-DD HH:MI: SS

General form:

INSERT INTO table_name

VALUES (value1,value2...);

Example:

Consider a Table name : **order** with following attributes

o_id	pname	o_date

INSERT function will be shown below:

INSERT INTO order
VALUES ('1', 'table', '2022-01-13');

o_id	pname	o_date
1	table	2022-01-13

- **UPDATE**

To perform update operations on a table/database we have to use UPDATE statement.

General Form -

```
UPDATE TABLE
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Note: Always remember to use WHERE with the UPDATE statement otherwise values will be updated for all the columns.

Ex- UPDATE Kuldeep to Kuldeep Ravaliya and MongoDB to MongoDB & SQL.

Query:

```
UPDATE Student
SET Student_name = 'Kuldeep Ravaliya' AND Course = 'MongoDB & SQL'
WHERE Student_id = 2;
```

Output:

Student_id	Student_name	Batch	Course
1	Ojasv	2018	SQL
2	Kuldeep Ravaliya	2018	MongoDB & SQL

- **Additional Information:**

The functions below can be used in the INSERT and UPDATE clauses to add the current timestamp in the column value.

1. CURDATE(): The CURDATE() function returns the current date value in the 'YYYYMMDD' format or "YYYY-MM-DD' format. It depends on whether a string or numeric value is used in the table function.

   General Form:

   ```
   SELECT CURDATE();
   ```

2. CURTIME(): The CURTIME() returns the current time value in HHMMSS.uuuuuu format or 'HH:MM: SS' format. It depends on whether a string or numeric value is used in the table function.

   General Form:

```
SELECT CURTIME();
```

- **DELETE**

To perform the delete operation on the existing table/database we use the DELETE statement

General form-

```
DELETE FROM table
WHERE condition;
```

Ex-

```
DELETE FROM Students
WHERE Student_name = 'Kuldeep Ravaliya';
```

Output:

Student_id	Student_name	Batch	Course
1	Ojasv	2018	SQL

Note- We can also delete all the records from the table using - DELETE FROM TABLE - this all table rows will be deleted but the table will still be there for use.

- **RENAME**

To perform Rename operation on a table we can use RENAME statement -

Ex- We can change the name of the student table 'Student' to 'Student1' using RENAME statement, syntax for that –

```
RENAME TABLE Student TO Student1;
```

Output:

Student1 -

Student_id	Student_name	Batch	Course
1	Ojasv	2018	SQL

- **REPLACE INTO**

To perform Replace operation on a table we can use REPLACE INTO statement.
General form-

```
REPLACE [INTO] T_name(column_name(s))
VALUES(value_list);
```

Ex-

```
REPLACE INTO Student1(student_name)
VALUES ('1','LOKESH', 2019,'SQL');
```

Output:

Student_id	Student_name	Batch	Course
1	Lokesh	2018	SQL

- **Difference between UPDATE & REPLACE INTO?**

UPDATE	REPLACE INTO
This is used to modify existing data in the table.	This works exactly like INSERT, but if an old row in the table has the same value as a new role for a PRIMARY KEY or a UNIQUE index, the old row is deleted before the new row is inserted.

- **DELETE in Safe Mode** -

By default **safe update mode** is enabled. We can disable it by this command:

```
SET SQL_SAFE_UPDATES = 0;
```

After this command, the DELETE clause can be used.

To enable the safe mode, we can use the following command:

SET SQL_SAFE_UPDATES = 1;

Safe mode exists to disallow update and delete operations without the use of PRIMARY KEY in a WHERE clause. Hence, we can prevent the loss of data with the help of DELETE in Safe Mode.

- **DELETE CASCADE :**
  This clause is used to delete multiple records from more than one table, linked through foreign key.

- **UPDATE CASCADE :**
  This clause is used to update multiple records from more than one table, linked through foreign key.

- **REPLACE :**
  - It is used to update the already present tuple data in a relation.
  - When we use REPLACE query with the help of WHERE clause in PRIMARY KEY column, then the row present will get update.
  - If there is no reference of the primary key, then a new tuple entry will be added in the relation, with updated values.