Module 6: MySQL Manipulate Data

INSERT Statement:

□ Insert statement is used to add one or more rows in a table.

□ The syntax of INSERT statement is:

→ □ **INSERT**

□ **VALUES**

□ In INSERT clause, we specify the name of the table that we want to add row to.

□ INTO is optional and also column_list is optional.

□ In the VALUES clauses we specify the values to be inserted.

□ If column_list is not specified then we must specify all column values.

□ If column_list is specified then column values must be in the same order as in the column list. We can omit the columns that have default values, accept null values or automatically generated.

□ To insert a null value we can use NULL keyword.

□ To insert default value or generate a value for an auto increment column we can use DEFAULT keyword.

□ We can use SET with insert statement also.

Insert and Errors:

1. If we **violate primary key** then we get:

→ ERROR 1062 (23000):

2. If the **column name is wrong** then we get:

→ ERROR 1054 (42S22):

3. If the **number of columns and their values don't match** we get:

   ERROR 1136 (21S01): Column count doesn't match value count at row 1.

4. If we **don't include the columns that need data** then we get:

   ERROR 1364 (HY000): Field 'rno' doesn't have a default value

5. If we insert multiple records and any of the rows would have violated any constraint then none of the rows would be inserted.

INSERT IGNORE Statement:

□ If we use INSERT IGNORE statement, then the rows that cause the errors are ignored and the remaining rows are inserted into the table.

□ The syntax of INSERT IGNORE statement is:

→ □ **INSERT**

□ **VALUES (expr_1 [, expr_2] ...)**

INSERT ON DUPLICATE KEY UPDATE:

- □ Using this option if the new row we INSERT causes a duplicate value in the UNIQUE or PRIMARY index, then MySQL performs an update to the old row based on the new values.
- □ The syntax is:
  - □ **INSERT INTO table(column_list)**
  - □ **VALUES(value_list)**
  - □ **ON DUPLICATE KEY UPDATE column_1 = new_value_1, column_2 = new_value_2, …;**

UPDATE Statement:

- □ UPDATE statement is used to modify one or more rows in a table.
- □ The syntax is:
  - → □ **UPDATE table_name**
  - □ **SET**
  - □ **[WHERE**
- □ In the SET clause we name each column and its new value.
- □ In the WHERE clause we specify the conditions that must be met for a row to be updated.
- □ If the WHERE clause is omitted then all rows would be updated.

UPDATE with ORDER BY and LIMIT:

- □ UPDATE statement can also be used with Order by and Limit clauses.
- □ The syntax is:
  - □ **UPDATE table_name**
  - □ **SET column_name_1 = expression_1 [, column_name_2 = expression_2]…**
  - □ **[WHERE search_condition]**
  - □ **[ORDER BY expression [<u>ASC</u> | DESC]]**
  - □ **[LIMIT number_rows]**
- □ ORDER BY is used with LIMIT to sort the records and limiting the number of records to be updated.

DELETE Statement:

- □ DELETE statement is used to delete one or more rows from the table specified in the DELETE clause.
- □ The syntax is:
  - → □ **DELETE**
  - □ **[WHERE**
- □ In the WHERE clause we specify the conditions that must be met for a row to be deleted.
- □ A foreign key constraint may prevent us from deleting a row. In such case we can only delete the row if we delete all child rows for that row first.

DELETE with ORDER BY and LIMIT:

- □ DELETE statement can also be used with Order by and Limit clauses.
- □ The syntax is:
  - □ **DELETE from table_name**
  - □ **[WHERE search_condition]**
  - □ **[ORDER BY expression [ASC | DESC]]**
  - □ **[LIMIT number_rows]**
- □ ORDER BY is used with LIMIT to sort the records and limiting the number of records to be deleted.

REPLACE Statement:

- → □ Replace statement is used to
- □ The syntax of REPLACE statement is:
  - □ **REPLACE [INTO] table_name [(column_list)]**
  - □ **VALUES (expr_1 [, expr_2] …)**
  - □ **[expr_1 [, expr_2] …]…**
- □ If table contains unique valued index and if we try to INSERT a record containing a key value that already exist , a duplication-key violation error occurs and the row is not inserted.
- □ What if we want the new record to take priority over existing one?
- □ We can remove the existing record with DELETE and then use INSERT to add the new record.
- □ REPLACE is like INSERT except that it deletes old record when duplicate unique key value is present in a new record.
- □ Advantage of REPLACE instead of DELETE and INSERT is that REPLACE is performed as single atomic operation.
- □ Note: if new record is not duplicate then it behaves like INSERT.

Module 6 MCQ:

Consider following table description for Q1 to Q6

```
Create Table: CREATE TABLE `st11` (
  `rno` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`rno`)
)
```

Q1) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 () values ();
```

Options:

A. error
B. it inserts 1 record (Null, Null)
C. it inserts 1 record (1, Null)
D. it inserts 1 record (0, Null)

Solution:

Q2) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 set rno = 10;
```

Options:

A. error
B. it inserts 1 record (10, 10)
C. it inserts 1 record (1, Null)
D. it inserts 1 record (10, Null)

Solution:

3) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 (10, 'kamal');
```

Options:

A. error
B. it inserts 1 record (10, 'kamal')
C. it inserts 1 record (1, Null)
D. it inserts 1 record ('kamal', 10)

Solution:

Q4) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 value('kamal', 20);
```

Options:

A. it inserts 1 record (20, 'kamal')
B. it inserts 1 record ('kamal', 20)
C. error.
D. it inserts 1 record (20, null)

Solution:

Q5) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 values(10, 'kamal', 11, 'vimal');
```

Options:

A. it inserts 2 records (10, ‘kamal’) and (11, ‘vimal’)
B. it inserts 1 record (10, ‘kamal’)
C. it inserts 1 record (11, ‘vimal’)
D. error.

Solution:

Q6) What is the result when following statement is executed assuming table does not have any row?

```
insert into st11 values(10, 'kamal'), (10, 'vimal');
```

Options:

A. it inserts 2 records (10, ‘kamal’) and (11, ‘vimal’)
B. it inserts 1 record (10, ‘kamal’)
C. it inserts 1 record (10, ‘vimal’)
D. error.

Solution:

Q7) How can you change the primary key value of a row?

Options:

A. You cannot change the primary key value.
B. Change it with a simple UPDATE statement.
C. The row must be removed with a DELETE and reentered with an INSERT KEY.
D. This is only possible using UPDATE PRIMARY KEY statement.

Solution:

Q8) Wrong statement about UPDATE keyword?

Options:

A. If WHERE clause in missing in statement the all records will be updated.
B. Only one record can be updated at a time using WHERE clause
C. Multiple records can be updated at a time using WHERE clause
D. None is wrong statement

Solution:

Q9) What will be the result?

```
UPDATE employees
SET salary=salary * 1.1;
```

Options:

A. The statement will fail because there is no WHERE clause to restrict the rows affected.

B. The first row in the table will be updated.

C. There will be an error if any row has its SALARY column NULL.

D. Every row will have SALARY incremented by 10 percent, unless SALARY was NULL.

Solution:

Q10) What is the result of following assuming table as 4 records in emp table:

```
delete * from emp;
```

Options:

A. table will have 0 records.

B. error.

C. table will have 4 records as delete did not match any rows.

D. table will have 1 blank record.

Solution:

Q11) Which of these commands will remove every row in a table while keeping its structure intact? Select two

Options:

A. A DELETE command with no WHERE clause

B. A DROP TABLE command

C. A TRUNCATE command

D. An UPDATE command, setting every column to NULL and with no WHERE clause

Solution: