Summary – Day 3

SELECT:

The SELECT statement in MySQL is used to fetch data from one or more tables. We can retrieve records of all fields or specified fields that match specified criteria using this statement. It can also work with various scripting languages such as PHP, Ruby, and many more.

Syntax:

It is the most commonly used SQL query. The general syntax of this statement to fetch data from tables are as follows:

```
SELECT field_name1, field_name 2,… field_nameN
FROM table_name1, table_name2...
[WHERE condition]
[GROUP BY field_name(s)]
[HAVING condition]
[ORDER BY field_name(s)]
[OFFSET M ][LIMIT N];
```

LIMIT:

The LIMIT clause is used in the SELECT statement to constrain the number of rows to return. The LIMIT clause accepts one or two arguments. The values of both arguments must be zero or positive integers.

Syntax:

```
SELECT
    select_list
FROM
    table_name
LIMIT [offset,] row_count;
```

In this syntax:

- The offset specifies the offset of the first row to return. The offset of the first row is 0, not 1.
- The row_count specifies the maximum number of rows to return.

DISTINCT:

When querying data from a table, you may get duplicate rows. To remove these duplicate rows, you use the DISTINCT clause in the SELECT statement.

Syntax:

```
SELECT DISTINCT
    select_list
FROM
    table_name
WHERE
    search_condition
ORDER BY
    sort_expression;
```

- In this syntax, you specify one or more columns that you want to select distinct values after the SELECT DISTINCT keywords.
- If you specify one column, the DISTINCT clause will evaluate the uniqueness of rows based on the values of that column.
- However, if you specify two or more columns, the DISTINCT clause will use the values of these columns to evaluate the uniqueness of the rows.

WHERE:

MySQL WHERE Clause is used with SELECT, INSERT, UPDATE and DELETE clause to filter the results. It specifies a specific position where you have to do the operation.

Syntax:

```
WHERE conditions;
```

There can be one condition or many conditions written with AND/OR clauses.

AND/OR:

In MySQL, you can use AND & OR condition both together with the SELECT, INSERT, UPDATE and DELETE statement. While combine these conditions, you must be aware where to use round brackets so that the database know the order to evaluate each condition.

Syntax:

```
WHERE condition1
AND condition2
...
OR condition_n;
```

IN:

The IN operator allows you to determine if a value matches any value in a list of values.

Syntax:

```
value IN (value1, value2, value3,...)
```

NOT IN:

The NOT operator negates the IN operator

Syntax:

```
value NOT IN (value1, value2, value2)
```

The NOT IN operator returns one if the value doesn't equal any value in the list. Otherwise, it returns 0.

BETWEEN:

The BETWEEN operator is a logical operator that specifies whether a value is in a range or not.

Syntax:

```
value BETWEEN low AND high;
```

EXISTS:

The EXISTS operator in MySQL is a type of Boolean operator which returns the true or false result. It is used in combination with a subquery and checks the existence of data in a subquery. It means if a subquery returns any record, this operator returns true. Otherwise, it will return false. The true value is always represented numeric value 1, and the false value represents 0. We can use it with SELECT, UPDATE, DELETE, INSERT statement.

```
SELECT col_names
FROM tab_name
WHERE [NOT] EXISTS (
   SELECT col_names
   FROM tab_name
   WHERE condition
);
```

ISNULL:

The ISNULL function takes one argument and tests whether that argument is NULL or not. The ISNULL function returns 1 if the argument is NULL, otherwise, it returns 0.

Syntax:

```
ISNULL(expr)
```

IS NOT NULL:

The IS NOT NULL operator is used to test for non-empty values (NOT NULL values).

Syntax:

```
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

Wild Cards:

The wildcards in MySQL are characters that allow us to search complex data from the table very easily and quickly. It works with string by substituting one or more characters and produce the result after matching the string into the table.

MySQL uses wildcards in conjunction with the LIKE or NOT LIKE comparison operators. The LIKE operator works with the WHERE clause to find the result for a specified pattern in a column.

Types of wildcards:

%	This symbol indicates zero or more characters.
_	This symbol is used to match exactly one(single) character.

Syntax:

```
SELECT * FROM tab_name WHERE column_name LIKE 'X%';
OR,
SELECT * FROM tab_name WHERE column_name LIKE '%X';
OR,
SELECT * FROM tab_name WHERE column_name LIKE 'X%X';
```

```
SELECT * FROM tab_name WHERE column_name LIKE 'X_';
OR,
SELECT * FROM tab_name WHERE column_name LIKE '_X';
```

ORDER BY:

When you use the SELECT statement to query data from a table, the order of rows in the result set is unspecified. To sort the rows in the result set, you add the ORDER BY clause to the SELECT statement.

Syntax:

```
SELECT
   select_list
FROM
   table_name
ORDER BY
   column1 [ASC|DESC],
   column2 [ASC|DESC],
   ...;
```

In this syntax, you specify the one or more columns that you want to sort after the ORDER BY clause.

The ASC stands for ascending and the DESC stands for descending. You use ASC to sort the result set in ascending order and DESC to sort the result set in descending order respectively.