**Select Statement**

Used to select data from database. The data returned is stored in a result-table called result set.

To list all the columns of a table use \* along with select statement.

Eg. select \* from products

To list only the selected columns , mention the columns name along with select statement

Eg : select productName, price from products.

**Select Distinct**

The SELECT DISTINCT statement is used to return only distinct (different) values.

Eg : select distinct productName from products

One productName will appear only once. Unique productName will be listed

**Order By keyword**

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

**Example 1 : Ascending order**

select \* from products order by productName

Above query is same as

select \* from products order by productName asc

**Example 2 : Descending order**

select \* from products order by productName desc

**Insert into keyword**

The INSERT INTO statement is used to insert new records in a table.

It is possible to write the INSERT INTO statement in two ways.

The first way specifies both the column names and the values to be inserted:

INSERT INTO *table\_name* (*column1*, *column2*, *column3*, ...)

VALUES (*value1*, *value2*, *value3*, ...);

Ex : insert into products (productName, price) values ("Radio",2000);

Note :

You can skip column which is allowed to take null or default values.

In the above example productId was auto increment, so productId column has been skipped in the query

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table.

The INSERT INTO syntax would be as follows:

INSERT INTO *table\_name*

VALUES (*value1*, *value2*, *value3*, ...);

**Ex :**

insert into products values (11,"Cable",150);

In this kind of query all the values for columns must be entered, with same order as the columns in the table.

**Null value**

A field with a null value is a field with no value.

## **How to Test for NULL Values?**

We will have to use the IS NULL and IS NOT NULL operators instead.

To select records with null values use IS NULL

To select records without null values use IS NOT NULL

Example : IS NULL

select \* from products where price is null

Above query will list the products whose price is null

Example : IS NOT NULL

select \* from products where price is not null

Above query will list the products whose price is not equal to null

**Update query :**

The UPDATE statement is used to modify the existing records in a table.

Syntax :

UPDATE *table\_name*

SET *column1* = *value1*, *column2* = *value2*, ...

WHERE *condition*;

Ex :

update products

set productName = "black salt",price = 25

where productId = 1;

**Delete statement**

The DELETE statement is used to delete existing records in a table.

DELETE Syntax

DELETE FROM *table\_name*

WHERE *condition*;

Ex: delete from products where productId=1

## Delete All Records :

It is possible to delete all rows in a table without deleting the table.

DELETE FROM *table\_name*;

Or

DELETE \* FROM *table\_name*;

**Select Top clause**

The SELECT TOP clause is used to specify the number of records to return.

SELECT TOP *number*|*percent* *column\_name(s)*

FROM *table\_name*

WHERE *condition*;

Note : top keyword doesn’t exist in mysql

Or

SELECT *column\_name(s)*

FROM *table\_name*

WHERE ROWNUM <= *number*;

Note : rownum keyword doesn’t exist in mysql

Or

Using LIMIT keyword

>

SELECT productName

FROM products

limit 3

Note : if you want top 3 sorted records with respect to product name

>

SELECT productName

FROM products

order by productName

limit 3