SB101

Day-10

SQL PART-II
DDL, DML, DRL statement

SQL Queries

To see the list of existing databases-

```
SHOW DATABASES;
```

To create a new Database

```
CREATE DATABASE db-name;
```

To use a database

```
USE db-name;
```

To create the table

```
CREATE TABLE table-name(
        column-name data-type(size) constraint,
        column-name data-type(size) constraint,
        ...
        column-name data-type(size) constraint,
);
```

i. For date type, size is not applicable
ii. For int/float/double, size is optional
iii. for char/varchar, size is compulsory
iv. constraint are not compulsory

SQL Queries

To see the list of tables

```
SHOW TABLES;
```

To see the structure of the tables

```
DESC table-name;    or    DESCRIBE table-name;
```

To add a new columns

```
ALTER TABLE table-name ADD column-name data-type(size) constraint;
```

To change the data type/add constraint from a column

```
ALTER TABLE table-name
MODIFY col-name data-type(size) constraint;
```

To change the name of the column

```
ALTER TABLE table-name
CHANGE old-name new-name data-type(size) constraint;
```

SQL Queries

To drop a columns

```
ALTER TABLE table-name DROP column-name;
```

To drop primary key

```
ALTER TABLE table-name DROP PRIMARY KEY;
```

DROP not null using ALTER TABLE

```
ALTER TABLE table-name MODIFY column-name data-type(size) NULL;
```

Dropping unique

```
ALTER TABLE table-name DROP INDEX constraint-name;
```

Tip: use SHOW CREATE TABLE table-name; to get constraint name

Removing default using ALTER TABLE

```
ALTER TABLE table-name ALTER col-name DROP DEFAULT;
```

SQL Queries

To rename the table

```
ALTER TABLE table-name RENAME TO new-name;
```

To add primary key

```
ALTER TABLE table-name ADD PRIMARY KEY (col-name);
```

Applying not null using ALTER TABLE

```
ALTER TABLE table-name MODIFY column-name data-type(size) NOT NULL;
```

Applying unique using ALTER TABLE

```
ALTER TABLE table-name MODIFY col-name data-type(size) UNIQUE;
```

Apply default using ALTER TABLE

```
ALTER TABLE table-name ALTER col-name SET DEFAULT value;
```

SQL Queries

To insert record(s) in table

INSERT INTO table-name VALUES (list-of-values);

1. It is necessary to mention values for all columns, &

2. Order of values and order of columns in the table must be matching.

INSERT INTO table-name (column-list) VALUES (list-of-values)

1. You can mention values for selected columns. Remaining columns will take default value.

2. In query, Order of values and order of columns must be matching yet order of column in table does not play any roll here.

- ❑ Date and String values must be written in the ' or ""
- ❑ Numerical values may or may not be in the ' or ""
- ❑ null value must not be in the ' or "" mark
- ❑ Date format in MySql is YYYY-MM-DD

Multiple records can be added to a table using single insert query.

SQL Queries

To update record from table

```
UPDATE table-name SET col-name = value, col-name = value ....
WHERE Condition;
```

Condition is optional but in absence of condition, all records of the table will be updated which is undesirable

To delete record from table

```
DELETE FROM table-name WHERE condition;
```

Condition is optional but in absence of condition, all records of the table will be deleted; To achieve this functionality use TRUNCATE command

SQL Queries

DQL (Data Query Language) used to fetch/read/retrieve record from the table

```
SELECT projection FROM table-name WHERE condition
GROUP BY col-name HAVING condition
ORDER BY col-name sorting-order, col-name sorting-order;
```

Possible values for projection

* : all columns | column-list: comma separated list of columns to be displayed

DISTINCT(col-name): used to display distinct value of a column

column-name as new-column-name *OR* column-name new-column-name

if new-column-name contains multiple words then put that new-column-name in ""/''

Arithmetic Expression

Tip: If Aliased name is not a multi word then using the quotation is optional.

WHERE clause is used to apply filter on the rows i.e. it is used to define selection. To make condition of WHERE clause we have some operators

> (Greater than)	< (Less than)	>= Greater than or equals to
!= or <> Not equals	= Equal	<= Less than or equals to

Do not use == for equality

SQL Queries

between - and: It is used to fetch the result for a range of values such that both the starting and ending values are inclusive. Mainly it is useful for date and numerical values.
To invert the result, you can use **NOT** operator with between - and

&&/and: used to join two or more conditions such that all the participating conditions are required to be true for the result

||/or: used to join two or more conditions such that any of the participating conditions is required to be true for the result

IN: used to find record that matches a set of values specified in IN clause. you can use **NOT** operator with IN also.

Comparing the **NULL** value

- Never use != or = to compare the value with NULL
- Use **IS NULL** and **IS NOT NULL**

LIKE: It is used for pattern matching

- **%**: Any number of characters (zero or more)
- **_**: fixed number of characters (one _ is for one character)

SQL Queries

ORDER BY: Used to sort the data according to value of a specified column.

- ❑ **The default order of sorting is ascending and for sorting in the descending order we have to use DESC.**
- ❑ **The column used for sorting the records may or may not be part of SELECT clause.**
- ❑ **The aliased column name can be used with the ORDER BY clause**
- ❑ **Keep Caution: Make sure that the ORDER BY clause must be the last clause of the SELECT query & the ORDER BY clause can only be used with the SELECT query.**

Single Row Functions

These functions return one result for every row.

Number function: abs(n), mod(m,n), round(m,n), truncate(m,n), ceil(n), floor(n), greatest(), least()

Character function: upper(), lower(), length(), replace(), concat() & substr()

Date Function: sysdate(), curdate(), now(), date_format(date, format), adddate(date, duration to add)

In MySQL, The index of String is started from 1 unlike programming languages where it starts from 0. For String "MASAI" The letter M is on index 1, A is on index 2 and so on in MySQL.

SQL Queries

Aggregate Functions/Group Functions/Multi-row function

- They are applied on multiple rows together but that produce single result & used to ignore null values
- Total 5 functions: MIN(), MAX(), SUM(), AVG() and COUNT()
  - ❑ Use SUM() and AVG() for numerical data only
  - ❑ Use MIN(), MAX() and COUNT() for all data types
- Aggregate functions cannot be used with the WHERE clause but can be used with SELECT or HAVING clause

Special Function

- ❑ IF(condition, value-IF_true, value_IF_false) function
- ❑ CASE() function

```
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
WHEN condition3 THEN result3
ELSE result;
END;
```

SQL Queries

- To See name of database currently in use

```
SELECT DATABASE();
```

- To Truncate Table

```
TRUNCATE TABLE table-name;
```

- To drop the table

```
DROP TABLE table-name;
```

Difference between TRNUCATE and DROP

TRUNCATE: Delete all the records from the table but table structure will be unaffected

DROP: Delete all the records from the table along with the table structure

- To drop the database

```
DROP DATABASE db-name;
```

SQL Queries

Group By: The main purpose of the group by clause is to group the records/ rows logically.

1. **All non-aggregate columns of the SELECT clause must be mentioned in the GROUP BY clause such that GROUP BY clause can have any additional non-aggregate column**
2. **If select list contains the non-aggregate and aggregate columns then use of GROUP BY is mandatory.**
3. **You must not use aggregate function with GROUP BY, use column name(s) only.**
4. **You must not use column alias with the GROUP BY because SELECT clause is executed after GROUP BY so aliased names are not available at the time of execution of the GROUP BY.**
5. **group by can be used for more than one column also.**
6. **You can use WHERE clause with GROUP BY clause but WHERE clause is used to apply filter condition row wise means first records will be filtered using WHERE condition and then GROUP will be made from the results after executing WHERE. In short; WHERE clause does not apply condition on the result of GROUP BY it apply condition of every record one by one.**

SQL Queries

HAVING CLAUSE: Having allows aggregate functions to be used as filter criteria which cannot be done using WHERE clause i.e. the aggregate functions cannot be used with where clause.

1. **Where is used to filter the content at row level but having is used to filter the content at group level i.e. WHERE clause is to restrict the rows and HAVING is to restrict the GROUP.**
2. **HAVING cannot be written without the GROUP BY clause in the query. HAVING clause should not contain non-aggregate columns which are not present in the GROUP BY clause**

QUERY?