How to connect:

Open cmd in bin folder and give command :

🡺 Mysqld –console

Next Open another command prompt using bin folder and give command:

🡺mysql –u root –p

Sql stands for structured query language which is standard language for accesing and manipulating datbases

In nrml lang it helps us to communicate with DB

**Categories In SQL Commands**

**SQL**

Data Query language Data Definition Data Maniplation Data Control

Insert

Update

delete

Grant

Revoke

Create

Alter

Drop

Select

DQL-It has only one command which accompanies with other sql clauses to compose queries against DB

DDL-It is used to restructure the table.

DML-it is used to manipulate data in the table.

DCL-it is used to control the access within DB

**How to Create a DataBase:**

CREATE DATABASE databasename;

**How to use that database**:

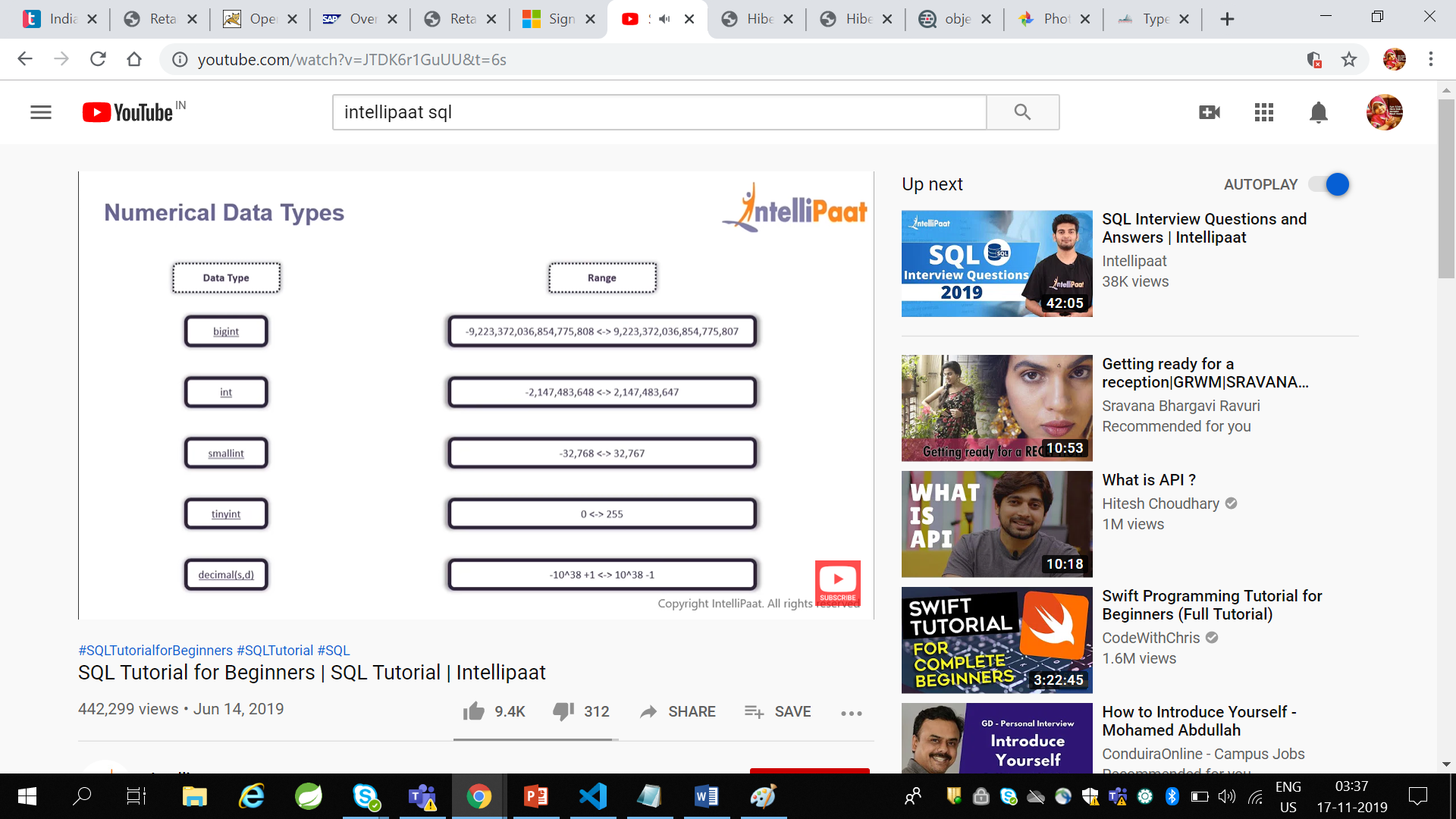
USE databaseName;

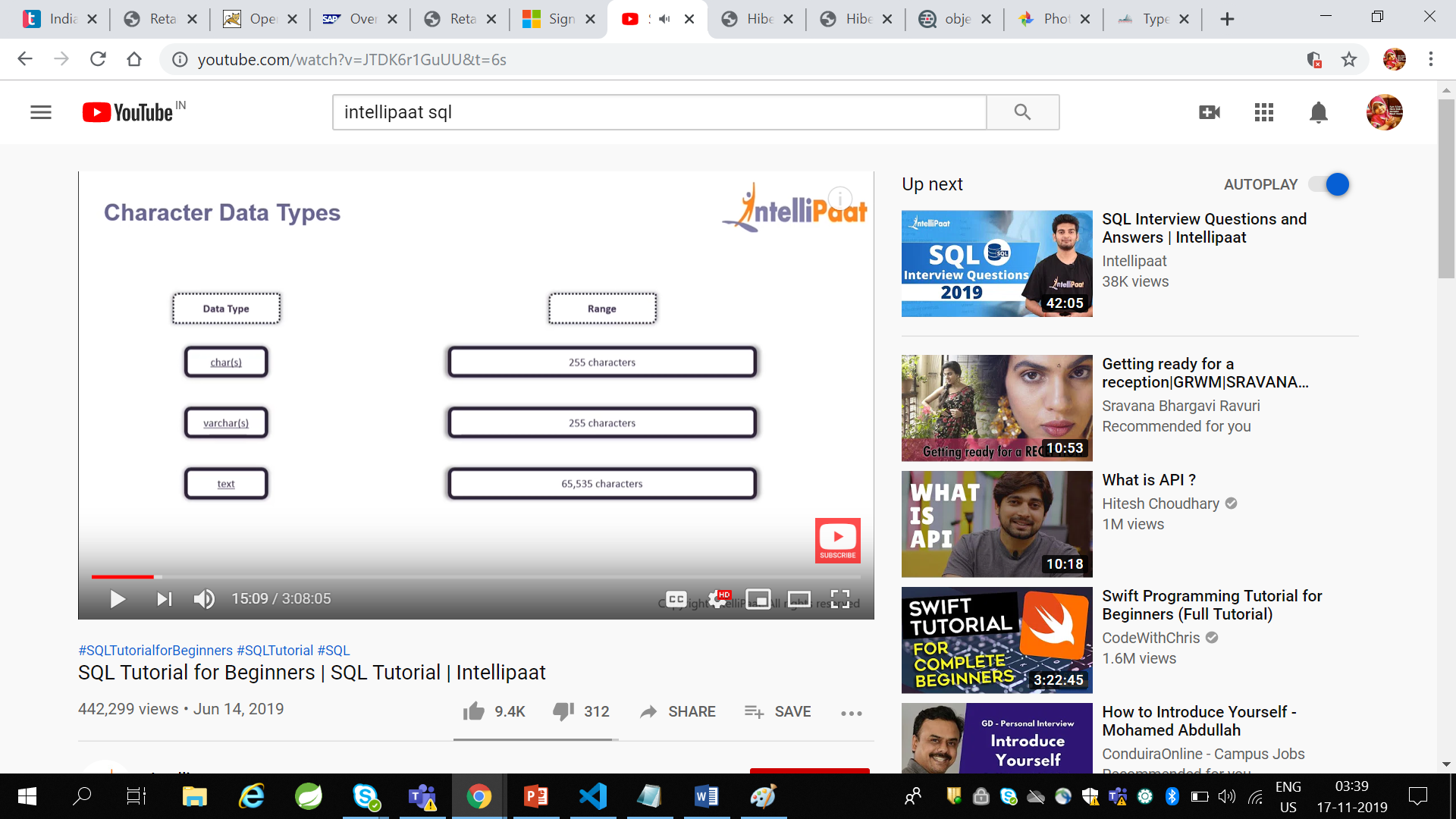
How to **delete database:**

DROP DATABASE databasename;

**DataType in SQL:**

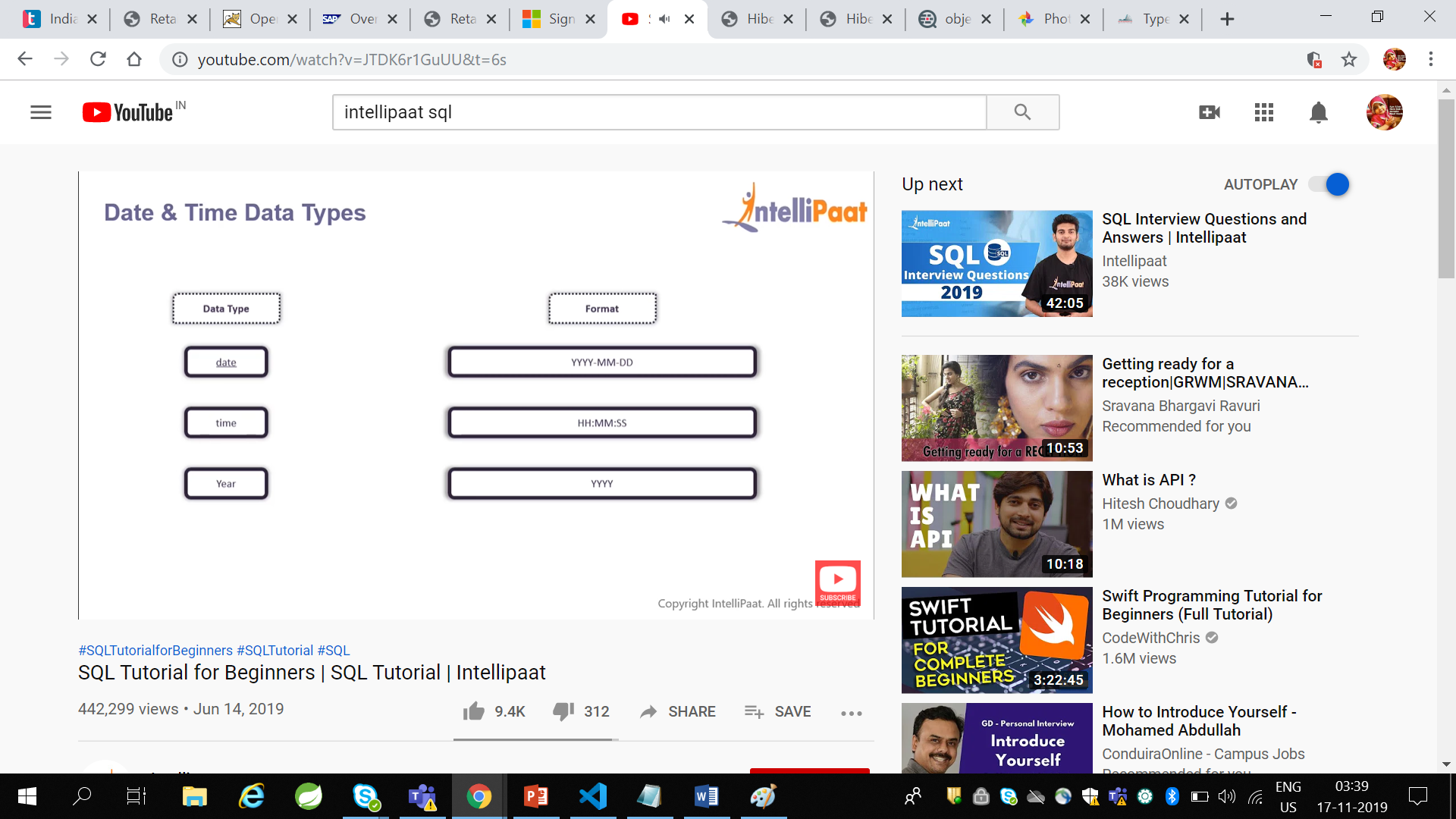
Data type define what type of data a column can hold





When char is used data wastage may happen when we 3 chars if it has capability to take 10 chars

Incase of varchar even it has 10 chars capability if we give 3 chars it assigns 3 chars memory location



**Constraints in SQL:**

Constraints are used to specify rules for data in table

NotNull- No column value is Null

Default-It sets default value to the column when no value is specified

Unique-It ensures that all values of the column are unique

PrimaryKey-it uniquely identifies each record in a table

**Table:**

A Table is DB object which has rows and columns

In SQL rows are known as RECORDS(it gives complete info of an entity) ,columns are FIELDS(which provides specific info of data).

**How to create a Table:**

CREATE TABLE table\_name(

Create table employee(

e\_id int not null,

e\_name varchar(20)

e\_age int,

e\_salary int,

primary key(e\_id)

);

Column1 datatype,

Column2 datatype,

Column3 datatype,

……

column datatype,

PRIMARY KEY(column\_x);

🡪A table cannot have more than 1 primary key

**How to INSERT records into Table**

INSERT INTO table\_name VALUES(value1,value2…..valueN);

**How to SELECT values from Table**

SELECT column1,column2,column3,…..columnN FROM tablename;

SELECT \*FROM tablename -🡪extract complete data from table

**How to Select distinct value from Table**

SELECT DISTINCT columnname FROM tablename

SELECT DISTINCT e\_gender FROM employee

**How to show columns of table**

Show columns from table\_name;

**How to add new column to a table**

ALTER TABLE table\_name ADD column\_name datatype;

**Where Clause Syntax:**

SELECT column1,column2,….columnN FROM tableName WHERE condition;

SELECT \* FROM employee WHERE e\_salary>10000;

**AND Operator:**

SELECT column1,column2,…column FROM tableName WHERE condition1 AND condition2……. AND condition;

SELECT \* FROM employee WHERE e\_gender=’male’ AND e\_salary>10000;

**OR Operator:**

SELECT column1,column2,…column FROM tableName WHERE condition1 OR condition2……. OR condition;

SELECT \* FROM employee WHERE e\_gender=’male’ OR e\_salary>10000;

**NOT Operator:**

SELECT column1,column2,…column FROM tableName WHERE NOT condition1 ;

SELECT \* FROM employee WHERE NOT e\_gender=’male’ ;

**Like Operator:**

Like Operator is used to extract records where a particular pattern is present

Like is used as conjunction ib between wild card characters

Wild card characters are:

Percentage symbol 🡪Represents zero , one or multiple characers

underscore symbol🡪Represents a single character

SELECT columns FROM tablename WHERE column\_N LIKE ‘\_XXXX%’;

SELECT \* FROM Employee WHERE e\_name ‘j%’; o/p:john,james,jaya

SELECT \* FROM employee WHERE e\_age ‘3\_’; o/p:30,33,35

**BETWEEN Operator :**

SELECT col\_list FROM tablename WHERE column\_N BETWEEN val1 AND val2;

SELECT \* FROM employee WHERE e\_age BETWEEN 30 AND 34;

SELECT \* FROM employee WHERE e\_salary 90000 AND 120000;

**Aggregrate Functions inSQL:**

**1) MIN()**

**2) MAX()**

**3) COUNT()**

**4) SUM()**

**5) AVG()**

**1.MIN()**

SELECT MIN(column\_name) FROM table\_name;

SELECT MIN(e\_age) FROM employee;

**2.MAX()**

SELECT MAX(column\_name) FROM table\_name;

SELECT MAX(e\_age) FROM employee;

**3.Count():**It returns the no. of rows that matches a specific criteria

SELECT COUNT(\*)FROM table\_name WHERE condition;

SELECT COUNT(\*) FROM employee WHERE e\_gender=’male’;

**4.SUM():**It gives total sum of a numeric column

SELECT SUM(column\_name) FROM table\_name;

SELECT SUM(e\_salaries) FROM employee;

**5.AVG():**It gives total AVG of a numeric column

SELECT AVG(column\_name) FROM table\_name;

SELECT AVG(e\_age) FROM employee;

**String Functions in SQL:**

**1.LTRIM function: TO remove spaces on left side we use this.**

SELECT LTRIM(“ hello”);

o/p:hello

2.lower and upper function

SELECT LOWER(“HELLO WORLD”);

o/p:hello world

SELECT UPPER(“hello world”);

o/p:HELLO WORLD

3.REVERSE function:

SELECT REVERSE(“hello”);

o/p:olleh

4.SUBSTRING Function():

SELECT SUBSTRING(“hello world”,7,5);

o/p:world

ORDER BY:By default it has a sorting of ascending order

SELECT column\_list FROM table\_name ORDER BY columns ASC|DESC

SELECT \* FROM employee ORDER BY e\_salary;

o/p:9000,11000,54000,77000

SELECT \* FROM employee ORDER BY e\_salary DESC;

o/p:77000,54000,11000,9000

TOP clause: it is to get top N columns

SELECT TOP x column\_list FROM table\_name;

SELECT TOP 3 \*FROM employee ;

SELECT TOP 3 \*FROM employee ORDER BY e\_age DESC;

GROUP BY :

It is used to get aggregrate result with respect to group

SELECT column\_list FROM table\_name WHERE condition GROUP BY colnames ORDER BY colnames;

SELECT column\_list FROM table\_name GROUP BY colnames;

SELECT avg(e\_salary),e\_gender from employee GROUP BY e\_gender;

SELECT AVG(e\_age),e\_dept FROM employee GROUP BY e\_dept ORDER BY AVG(e\_age) DESC;;

mysql> select avg(b\_count),b\_name from bottle group by b\_name order by avg(b\_count) desc;

o/p:

| avg(b\_count) | b\_name |

+--------------+--------------------+

| 9.6667 | Atlasware |

| 7.5000 | Tupperware |

Having Clause:

It is used in GROUP BY instead of WHERE like when we having some conditions.

Having clause is used in combination with GROUP BY

To impose conditions on group

SELECT column\_name(s) FROM table\_name WHERE condition GROUP BY column\_name(s) HAVING condition

ORDER BY column\_name(s)

mysql> select avg(b\_count),b\_name from bottle group by b\_name having avg(b\_count) >8 order by avg(b\_count) desc;

o/p:

| avg(b\_count) | b\_name |

+--------------+-------------------+

| 30.0000 | bisleri |

| 9.6667 | Atlasware |

**UPDATE Statement:**

Its used to modify the existing records in a table.

UPDATE table\_name SET col1=val1, col2=val2….[WHERE condition]

update bottle set b\_count=20 where b\_id=1;

**DELETE Statement:**

Delete statement is used to delete existing records in the table.

DELETE from table\_name where [ condition]

**TRUNCATE Statement:**

Truncate statement deletes all the data inside the table;

TRUNCATE TABLE table\_name;

TRUNCATE TABLE bottle;

**Inner joins:**

It returns records that have matching values in both the tables.It is also known as simple join.

SELECT columns FROM table1 INNER JOIN table2 ON table1.column\_x=table2.column\_y;

mysql> select student.s\_name,student.s\_bottle,bottle.b\_name,bottle.b\_colour from bottle INNER JOIN student on

-> bottle.b\_name=student.s\_bottle;

| s\_name | s\_bottle | b\_name | b\_colour |

+------------+------------+------------+----------+

| khairu | tupperware | Tupperware | blue |

| Mahesh | tupperware | Tupperware | blue |

| khairu | tupperware | Tupperware | pink |

| Mahesh | tupperware | Tupperware | pink |

| Sahithi | Atlasware | Atlasware | grey |

| Rahul | Atlasware | Atlasware | grey |

| Sahithi | Atlasware | Atlasware | pink |

| Rahul | Atlasware | Atlasware | pink |

| Sahithi | Atlasware | Atlasware | blue |

| Rahul | Atlasware | Atlasware | blue |

| khairu | tupperware | tupperware | green |

| Mahesh | tupperware | tupperware | green |

| Sree lekha | sports | sports | pink |

| Seetha | Sports | sports | pink |

| Reddy | sports | sports | pink |

| Sree lekha | sports | sports | blue |

| Seetha | Sports | sports | blue |

| Reddy | sports | sports | blue |

| Pooja | Bisleri | bisleri | blue |

| Pooja | Bisleri | bisleri | pink |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Left Join:**

Left join returns all the record from the left table and matched records from the right table

SELECT columns FROM table1 LEFT JOIN table2 ON table1.column\_x=table2.column\_y;

select bottle.b\_name,bottle.b\_colour,student.s\_name,student.s\_bottle from bottle LEFT JOIN student ON bottle.b\_name=student.s\_bottle;

**Right Join:**

Right join returns all the records from right table and matched records from left table

SELECT columns FROM table1 RIGHT JOIN table2 ON table1.column\_x=table2.column\_y;

mysql> select student.s\_name,student.s\_bottle,bottle.b\_name,bottle.b\_colour from student

-> RIGHT JOIN bottle

-> ON student.s\_bottle=bottle.b\_name;

**Full Join:**

It returns all rows from the Left table and the Right table with NULL values in place where the join condition is not met

SELECT columns FROM table1 FULL JOIN table2 ON table1.column\_x=table2.column\_y;

mysql> select student.s\_name,student.s\_bottle,bottle.b\_name,bottle.b\_colour from bottle

-> full JOIN student ON bottle.b\_name=student.s\_bottle;

**Update Using Join:**

mysql> update student join bottle on student.s\_bottle=bottle.b\_name set bottle.b\_name="Dr.copper" where bottle.b\_id=5;