**SQL Queries and its syntax**

**1.create:**

**database:** create database db\_name;

e.g. *create database geekyworks;*

**table :** create table tb\_name (col\_name datatype, col\_name datatype, …..);

e.g. *create table employee(eid int ,ename varchar(50) ,edpt varchar(50), esal int);*

**2.insert:**

Insert into table\_name (column1, column2, column3, ...)  
values (value1, value2, value3, ...);

e.g. *insert into employee(ename,edpt,esal)*

*values("prathamesh","electrical",15000);*

**3.select:**

select \* from tb\_name; //select all columns

or

select col\_name from tb\_name; //select specific column

e.g. *select \* from employee;*

or

*select ename from employee;*

**4.distinct:**

select distinct column1, column2, ...  
from table\_name; // return the different values from column.

**5. limit and offset:**

select \* from table\_name limit rowcount limit\_count offset\_count;

e.g. *select \* from actor limit 10 offset 10 ;*

//limit: used to return specific no of rows

//offset: used to skip the rows before beginning to return the rows.

**6. where:**

select column1, column2, ...  
from table\_name  
where condition;

e.g. *select \* from city where country\_id=44;*

//The WHERE clause is not only used in SELECT statements, it is also used in UPDATE, DELETE, etc.!

**Operators in The where Clause:**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| = | Equal |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| <> | Not equal. Note: In some versions of SQL this operator may be written as != |
| BETWEEN | Between a certain range |
| LIKE | Search for a pattern |
| IN | To specify multiple possible values for a column |

**7. and, or, not:**

and :

select column1, column2, ...  
 from table\_name  
where condition1 and condition2 and condition3 ...;

e.g. *select \* from city where city='Ahmadnagar' and country\_id=44;*

or:

select column1, column2, ...  
from table\_name  
where condition1 or condition2 or condition3 ...;

e.g. *select \* from city where city='Ahmadnagar' or country\_id=45;*

not:

select column1, column2, ...  
from table\_name  
where not condition;

e.g. *select \* from city where not country\_id=44;*

**8. order by:**

select column1, column2, ...  
from table\_name  
order by column1, column2, ... asc |desc;

e.g. *select \* from city order by city\_id desc;*

**9. update:**

update table\_name

set column1 = value1, column2 = value2, ...

where condition;

e.g. *update actor set first\_name="prathamesh",last\_name="dabhole"*

*where actor\_id=1;*

//The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated.

**10. delete:**

delete from table\_name where condition;

e.g. *delete from employee where ename="vinayak";*

delete from table\_name; //delete all rows in a table without deleting the table.

**11. Aggregate functions:**

min():

select min(column\_name)  
from table\_name  
where condition;

e.g. *select min(country\_id) as min\_country from city;*

max():

select max(column\_name)  
from table\_name  
where condition;

e.g. *select max(country\_id) as max\_country from city;*

count():

select count(column\_name)  
from table\_name  
where condition;

e.g*. select count(\*) from city;*

avg():

select avg(column\_name)  
from table\_name  
where condition;

e.g. *select avg(country\_id) from city;*

sum():

select sum(column\_name)  
from table\_name  
where condition;

e.g. *select sum(country\_id) from city;*

**12. like:**

select column1, column2, ...  
 from table\_name  
where column like pattern;

//There are two wildcards often used in conjunction with the LIKE operator:

The percent sign (%) represents zero, one, or multiple characters

The underscore sign (\_) represents one, single character

e.g. *select \* from city where city like "A%";*

*select \* from city where city like "%a";*

*select \* from film where title like "A%S";*

*select \* from film where title like "\_m%";*

*select \* from film where title like "%rs";*

**13. in operator:**

select column\_name(s)   
from table\_name  
where column\_name in (value1, value2, ...);

e.g.

*select \* from city where city in("Ahmadnagar","aden","adoni");*

**14. between operator:**

select column\_name(s)  
from table\_name  
where column\_name between value1 and value2;

e.g.

*select \* from city where country\_id between 1 and 50 order by country\_id asc;*

//The between operator is inclusive: begin and end values are included.

**15. joins:**

Supported Types of Joins in MySQL

INNER JOIN: Returns records that have matching values in both tables

LEFT JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT JOIN: Returns all records from the right table, and the matched records from the left table

CROSS JOIN: Returns all records from both tables

Inner join:

select column\_name(s)  
from table1  
inner join table2  
on table1.column\_name = table2.column\_name;

e.g*. select p.name,p.email,p.gender,d.name*

*from patient p*

*inner join doctor d*

*on p.doctor\_id=d.id;*

left join:

select column\_name(s)  
from table1  
left join table2  
on table1.column\_name = table2.column\_name;

e.g. *select p.name,p.email,p.gender,d.name*

*from patient p*

*left join doctor d*

*on p.doctor\_id=d.id;*

right join:

select column\_name(s)  
from table1  
right join table2  
on table1.column\_name = table2.column\_name;

e.g. *select p.name,p.email,p.gender,d.name*

*from patient p*

*right join doctor d*

*on p.doctor\_id=d.idz*

cross join:

select column\_name(s)  
from table1  
cross join table2;

e.g. *select p.name,p.email,p.gender,d.name*

*from patient p*

*cross join doctor d;*

**16. having:**

select column\_name(s)  
from table\_name  
where condition  
group by column\_name(s)  
having condition  
order by column\_name(s);

//having vs where

* where is applied on individual rows while having is applied on groups.
* having is applied after grouping while where is applied before grouping.
* having without group by is same as where.

**17. exists, any, all:**

exists:

select column\_name(s)

from table\_name

where exists

(select column\_name from table\_name where condition);

any:

select column\_name(s)  
from table\_name  
where column\_name operator any  
  (select column\_name  
  from table\_name  
  where condition);

e.g. *select \*from patient*

*where id=any*

*(select count(degree) as total from*

*doctor group by degree having total>1);*

all:

select column\_name(s)  
from table\_name  
where column\_name operator all  
  (select column\_name  
  from table\_name  
  where condition);

//exists return true if the subquery returns one or more records or null.

//any operator return true if any of the subquery value meet the condition.

//all operator return true if all of the subquery values meet the condition.

18.