MySQL

Journey

```
show databases ;
```

```
create database database_name ;

drop database databasename ;
```

everything is in mind

use dbname to enter

```
create table name ;

truncate table name ;
drop table name ;
```

to see description

```
desc name
```

~~Alter~~ alter table tablename
add column_name datatype ;

] to add column

```
alter table name
drop column name ;
```

] to delete

no need to specify (Just write the name)

```
alter table name
modify column name datatype ;
```

] to modify

Note:- constraints are used to specify rules for data in a table

```
alter table name
modify column datatype constraint ;
```

] to add constraint

```
alter table name
add unique (ID) ;
```

] to make any column unique

```
drop index age ;
```

concept ④ to drop unique

```
alter table name
add constraint uc_person unique ( ID, lname );
```

```
alter table name
drop index uc-person;
```

] to drop unique constraint .

Check constraint on alter table :-

```
alter table name —
add check (Age >= 18);
```

] for single column

↓
or

```
add constraint chk_per check (Age >=18 AND
                              city = 'sand');
```

```
drop check chk-per ;
```

only can be done

Default :-

```
alter table name
alter City set default 'sandness';
```

```
alter table name
alter City drop default ;
```

So far till now we learned to create a database & tables in it.

Now dealing with data values !-

→

show tables ; → to see tables in database inserting values :-

```
insert into table_name ( col1, col2, ........ )
  values ( value1, value2 . ~~~.....);
```

NOTE :- if you're adding values for all the columns of the table :-

```
insert into table-name ~~&~~
  values ( value1, value2, ......)
```

showing table

① select * from table_name ;

② select distinct columnname from tablename;

③ select count (distinct c-name) from t-name;

where keyword :-

where clause is used to filter records
It is used to extract only those records that fulfill a specified condn

NOTE :- The where clause is not only used in SELECT statement, it also used in update, delete etc.

update values in record

```
update table.name
set column.name = value
where column.name = value;
```

if no 'where' then all will be updated

delete record

```
delete from table-name where 'pid = 3;
```

[if no 'where', then all records will be deleted]

creating index :-

```
create unique index index_name
       (optional)
   on table_name(col1, col2, ...);
```

order by :-

```
select * from table-name
order by column.name ;
```

ascending order by default

optional to specify

dsc or asc

update syntax !-

```
update table-name
  set col1 = value1 , col2 = value2 ....
where condition ; (ext. necessary )
```

Delete syntax :-

```
delete from table-name
  where cond^n
```

limit clause syntax !-

```
select * from person        top 3
 limit 3
```

```
-> select * from person
   where age>18 or age>25          where
   limit 3;
```

Concept

```
select min(age) from person;
```

O/P →

min(age)
18

```
select min(age) as chota from person;
```

O/P →

chota
18

MySQL Like operator

└ used in where, to search for a specified pattern

two wildcards (%) & (_)

1)
```
select * from customers
where name like '%or%';
```
└ or must be there

2)
```
Select * from customer
where name like '_r%';
```
└ shows r ~~th~~ must be in second place

In operator

└ shorthand of or conditions in where clause

1)
```
select * from persons
where countery in ('Germany', 'UK', 'USA');
```
└ can be not in also

2)
```
Select * from person
where country in (Select *Country from
                  supplier);
```

B. between op^n:

1) select * from products
where price between 10 and 20
included

2) select * from person ✓
where id between 3 and 6;
→ where fname between 'Hari' and 'Ram';

o/p → Kanha — — —
Hari — — —
⋮
Ram — — —
Ashwin — — —

Aliases
exists for the duration of that query. ✓
'as' keyword is used.

* select fname as "fav person" from person;

NOTE :- single or double quote is required if ~~name~~ alias contains spaces.

join clause

Supported types of joins in MySQL :-

1) inner join
2) left join
3) right join
4) cross join

1) select p.fname as person_name, c.fname as customers_name from person as p inner join customers as c on p.id = c.id;

O/P

person-name	customers.name

No. of ~~[illegible]~~ records depends on minimal table

Union operator :

used to combine the result set of two or more select statements .

rules : • Every SELECT statements within UNION must have the same number of columns

- The columns must also have similar data types.
- The columns in every SELECT statement must be in the same order

```
select col-name(s) from table1
 union
select col.name(s) from table2;
```

group by statement :-

groups rows that have the same values into summary rows .

often used with aggregate functions

count() ,
max() ,
min(),
sum() , Avg()

1) Select count(fname), city from person
group by city
order by count(fname);

Having clause

* The having clause was added to SQL because the where keyword cannot be used with aggregate functions.

```
i select count(fname), city
  from person
  group by city
  having count(fname) = 1 ;
```

Exists operator

→ used to test for the existence of any record in a subquery.

→ return true if the subquery returns one or more records.

```
select fname from person
 where exists( select fname from customers
where age = 26);
```

ANY/ALL operators :-

allows you to perform a comparision b/w a single column value and a range of other values.

ANY op :-

↳ means that the condn will be true if the operation is ~~f~~ true for any of the values in the range.

↳ for all

ALL op ↗