use ==>
this command is use to select a database on which we will going to work...
i.e.... use databaseName;
select ===>
is used to select the columns from the tables ...
i.e.. select columnName from tableName;   select * from tableName;

where===>
is used to apply the condition on the search from the tables...
i.e... select * from tableName where columnName=salleh;

order by===>
is used to order the table rows...
i.e... select * from tableName order by columnName === it will order the rouws by columnName,
      from customers order by customer_id desc = "desc" is used for decending order..
as===>
is used to give the name to specific selected column from the table...
i.e
select first_name,
IMPORTANT== we can also give aliase to the tables... "i.e".. select * from customers c ... where c is the aliase to th
e tables
points,
(points+2)*1 as "discount offers"
from customers
distinct=====>
is used to get the unique data of the columns from the tables..
in=====>
is used to select data on condition ... it is used with where query ... alternat OR operator...
i.e...
select *
from customers
where state in ("va","fa")

between ====>
is used with AND operator... is used to select a range in which we can select things...
i.e...
select *
from customers
where birth_date between "1990-1-1" and "2000-1-1"

like ====>
is used to search a column value precisely....
  i.e...
   where last_name like "b%" ===its mean the lastname must start with b
   where last_name like "%y" ===its mean the lastname must end with y
   where last_name like "_____y" ==== its mean befor "y" there must be 5 characters
     where last_name like "b____y" ==== its mean befor starts with "b" in the middle there will be 4 characters and t
hen at end there will be y
   where last_name like "%b%"=== its mean "b" must be in the string, but can be anywhere...
regexp ===>
is used to match the patterns more precisely...
i.e, where last_name regexp "field$" === the string must end with "field"
    where last_name regexp "^field" === the string must start with "field"
    where last_name regexp "field|rose" === pipe sign(|) is used for multiple searches... it will search both of them
    where last_name regexp "[ri]e" === before "e" the character should be either "r" or "i" ... brackets are used to defi

nde the characters..
    where last_name regexp "[a-z]e" === befor "e" the character could be from "a" to "z" okay...
 is null , is not null =====>
is used to check the tables with null attributes in themmm...
i.e,   where phone is null=== it will select all the rows with "phone" null,
       where phone is not null === it will select all the rows with "phone" numbers...
limit=====>
this one is used for the limiting the selected data from the database...
i.e...
{select *
from customers
limit 3}= it will select the first 3 data from the table...
paginationg===>
it is used for the paginating between different pages ...
i.e,
{
select *
from customers
limit 6,3
}=== the first "6" is used for skiping data and the second "3" is used for the limiting the number of records to be fet
ched

inner joins===>
it is used to select data from another table , based on the unique id or the data present in other tables...
i.e,
we want to get all the orders of a cutomers, with some id...
we will then use this id to look into the table of orders.. where the cutomer_id is equal to the customer_id in the orde
rs
table

{
select o.customer_id,c.first_name,c.last_name,o.order_id
from customers c
inner join orders o
on c.customer_id=o.customer_id
}===
{
select o_i.order_id,o_i.product_id, p.name as product_name, (o_i.quantity*o_i.unit_price) as total_price
from order_items o_i
inner join products p
on o_i.product_id=p.product_id
}
self joining concept in the mysql====>
this concept is used to join the data inside same table...
for example we have a users table in the database...
we have a column which represents to whom should a user give reports to ...
and in that column we are storing the _id of the same table users _id...
then we can join the id of that user and get the data from the same table...
{{
select emp1.employee_id, emp1.first_name,emp2.first_name as reports_to
from sql_hr.employees emp1
inner join sql_hr.employees emp2
on emp1.reports_to=emp2.employee_id
}}
mulitiple tables inner joining concept in mysql====>

{{
select o.order_id,c.first_name,c.last_name,o.order_date,o_s.name
from orders o
inner join customers c
on o.customer_id=c.customer_id
inner join order_statuses o_s
on o.status=o_s.order_status_id
}}

compound join conditions ===>
outer join =====>
is used to get the data from "Left" table or from "Right" table either the condition is satisfyed or not...
outer join is of 2 types>>> "LEFT JOIN" & "RIGHT JOIN"
{{
select
c.customer_id,c.first_name,o.order_id
from customers c
left outer join orders o
on c.customer_id=o.customer_id
}}
multiple tables outer joining concept ====>
{{
select
c.customer_id,c.first_name,o.order_id,sh.name
from customers c
left outer join orders o
on c.customer_id=o.customer_id
left outer join shippers sh
on o.shipper_id=sh.shipper_id
}}

union====>
with union we can combine the results from mulitple queries...
{{
select *,"ACTIVE" as status
from orders
where order_date>="2019-1-30"
union
select *,"ARCHIVED" as status
from orders
where order_date<"2019-1-30";
}}
{{
select *,"BRONZE" as level
from customers
where customers.points<="2000"
union
select *,"PLATINUM" as level
from customers
where customers.points>"2000"
}}
insert into ===>
used to enter a new ROW of data into a table...
{{
insert into customers(

first_name,
last_name,
birth_date,
phone,
address,
city,
state,
points
)
values(
'salleh',
'shah',
default,
default,
'address',
'city',
'pk',
default
)
}}

inserting multiple rows in mysql concept=====>
{{
insert into products(
name,
quantity_in_stock,
unit_price
)
values(
"product1",
3,
40
),
(
"product12",
3,
40
),
(
"product3",
3,
40
)
}}

inserting in hierarchical way in multiple tables concept =====>>>
{{
insert into orders(
order_date,
customer_id
)
values(
'2022-2-17',
1

);

insert into order_items(
product_id,
quantity,
unit_price
)
values(
last_insert_id(),
2,
40
)
}}

last_insert_id() ===>
is used to get the _id of the last inserting row in the table...

is null ====>
is used to determine in mysql if the current column is null or empty
{{
where inv.payment_date is null
where inv.payment_date is not null
}}

create table tableName as =====>>>>
{{
create table payment_users as
select
inv.invoice_id,
inv.number,
c.first_name as client_name,
inv.invoice_total,
inv.payment_date
from sql_invoicing.invoices inv
inner join sql_store.customers c
on inv.client_id=c.customer_id
where inv.payment_date is not null
}}=== it will copy the records from another table and copy them and store the selected data into the new table we cr
eated!

update single row in mysql concept ====>>>
{{
update sql_invoicing.invoices inv
set inv.payment_total=10,inv.payment_date='2022-2-17'
where inv.invoice_id=1
}}

update multiple rows concept in mysql ===>>>
{{
use sql_store;
update customers c
set points=points+1
where birth_date<"1990-1-1"

}}

update with subqueries or nested queried===>
{{
update sql_store.orders o
set comments="Gold customers"
where o.customer_id in (
select c.customer_id
from sql_store.customers c
where c.points>3000
)
}}