select \* from datafiles.ESD;

ALTER TABLE datafiles.esd

RENAME COLUMN `Annual Salary` TO `Annual\_Salary`;

#where

select Job\_Title,Department,Age,Annual\_Salary from datafiles.esd where Job\_Title='Sr. Analyst';

#and

select Job\_Title,Department,Age,Annual\_Salary,Country from datafiles.esd where Country='china' or Country="United States";

#or

select Job\_Title,Department,Age,Annual\_Salary from datafiles.esd where Job\_Title='Sr. Analyst' and City="Seattle";

#not

select Job\_Title,Department,Age,Annual\_Salary from datafiles.esd where not Job\_Title='Sr. Analyst';

#like

select Job\_Title,Annual\_Salary from datafiles.esd where Job\_Title like "%eer";

select Job\_Title,Annual\_Salary from datafiles.esd where Job\_Title like "Man%";

select Job\_Title,Annual\_Salary from datafiles.esd where Job\_Title like "%ana%";

#order by dont use where clause

select \* from datafiles.esd order by Annual\_Salary asc;

select Full\_Name,Job\_Title,Annual\_Salary from datafiles.esd order by Annual\_Salary desc;

#limit

select Full\_Name,Job\_Title,Annual\_Salary from datafiles.esd order by Annual\_Salary desc limit 5;

# i want 1 value after first or 2 values after first

select Full\_Name,Job\_Title,Annual\_Salary from datafiles.esd order by Annual\_Salary desc limit 1,2;

#BETWEEN

select \* from datafiles.esd where Age between 30 and 35;

#in

select \* from datafiles.esd where Department in ("IT","Finance");

#not in

select \* from datafiles.esd where Department not in ("IT","Finance","Sales");

#string functions

select \* from datafiles.esd;

#concatination of 2 or more columns

select concat(Job\_Title ,"-",Department) as Designation from datafiles.esd;

#concat\_ws

select concat\_ws(" - ",Full\_Name,Job\_Title,Department) as Emp\_Name\_Desig from datafiles.esd;

#length of characters/ numbers in a datafield

select length(Annual\_Salary) as total\_Digits from datafiles.esd;

select upper(Full\_Name) as Upper\_Case from datafiles.esd;

select lower(Full\_Name) as Upper\_Case from datafiles.esd;

select left(Full\_Name,4) as Upper\_Case from datafiles.esd;

select right(Full\_Name,3) as Upper\_Case from datafiles.esd;

select mid(Full\_Name,3,5) as Upper\_Case from datafiles.esd; #(columnname,startindex,number of character)

#Aggregate Functions

select \* from datafiles.order\_details;

select sum(quantityOrdered) as Total\_Quantity from datafiles.order\_details;

select avg(quantityOrdered) as Avg\_Quantity from datafiles.order\_details;

select count(quantityOrdered) as Total\_orders from datafiles.order\_details;

select min(quantityOrdered) as Min\_order from datafiles.order\_details;

select max(quantityOrdered) as max\_order from datafiles.order\_details;

select truncate(priceEach,0) as Real\_Price from datafiles.order\_details;

select ceil(priceEach) as ceil\_Price from datafiles.order\_details;

select floor(priceEach) as floor\_Price from datafiles.order\_details;

select \* from datafiles.orders;

#date functions

select date(orderDate) as Date from datafiles.orders;

select time(orderDate) as Date from datafiles.orders;

select day(orderDate) as Date from datafiles.orders;

select month(orderDate) as Date from datafiles.orders;

select year(orderDate) as Date from datafiles.orders;

select hour(orderDate) as Date from datafiles.orders;

select minute(orderDate) as Date from datafiles.orders;

select monthname(orderDate) as Date from datafiles.orders;

select dayname(orderDate) as Date from datafiles.orders;

select datediff(shippedDate,orderDate) as Date from datafiles.orders;

select \* from datafiles.products;

select productName,productline,

case

when quantityInStock < 1000 then "need more production"

else "More for now"

end as Production\_details

from datafiles.products;

#multiple cases

select \* from datafiles.order\_details;

select quantityOrdered,priceEach,

case

when quantityOrdered <=30 then "less orders"

when quantityOrdered =40 then "High orders"

else "average orders"

end as Detail\_order

from datafiles.order\_details;

#Group by

select \* from datafiles.esd;

select Department,count(EEID) as Depart\_wiseEmp from datafiles.esd group by Department;

select Gender,count(EEID) as Depart\_wiseEmp from datafiles.esd group by Gender;

select \* from datafiles.products;

select productLine,count(productCode) from datafiles.products group by productLine order by count(productCode);

# having clause is used with aggregate function where we cannot use where clause

select \* from datafiles.esd;

select Department,count(EEID) from datafiles.esd group by Department having count(EEID)>130;

select \* from datafiles.products;

select productLiTne,sum(quantityInStock) from datafiles.products group by productLine having sum(quantityInStock)>5000;

# Joins in Mysql

select \* from datafiles.products;

select \* from datafiles.order\_details;

#inner joins, joins on the basis of common values

select products.productName,order\_details.quantityOrdered from datafiles.products

inner join datafiles.order\_details

on products.productCode = order\_details.productCode;

select products.productName,sum(order\_details.quantityOrdered) from datafiles.products

inner join datafiles.order\_details

on products.productCode = order\_details.productCode

group by products.productName;

#left join left column all data right column common data

select products.productName,order\_details.quantityOrdered from datafiles.products

left join datafiles.order\_details

on products.productCode = order\_details.productCode;

#right join

select products.productName,order\_details.quantityOrdered from datafiles.products

right join datafiles.order\_details

on products.productCode= order\_details.productCode;

# full join

select products.productName, order\_details.quantityOrdered from datafiles.products

cross join datafiles.order\_details

on products.productCode = order\_details.productCode;

select \* from datafiles.products cross join datafiles.order\_details;