Select Into Select

--\_\_\_\_\_ Simple \_\_\_\_\_

select \* from employees

--\_\_\_\_\_\_\_\_\_ Select into SElect \_\_\_\_\_\_\_\_\_\_\_

-- Jo Colum 2nd Select (Sub Query) kaa Under hon gaa ---> wohi Parent Select me hon gaa

select \* from (

select employee\_name,annual\_salary from employees

) as emp

**Fiydaa kya?**

Yaa krna saa Achaa to hum simple (Select colu\_1,2 from table ) kr laing ??

**Fiydaa**

select

departement,

salary,

salary/2 halfslary

from employee1

-- I need those Employee jin ke Half Salary ---> 35000 sa kam ho

--\_\_\_\_\_\_\_\_ 1st Way \_\_\_\_\_\_\_\_

select

departement,

salary,

salary/2 halfslary

from employee1

where salary/2 < 35000

--\_\_\_\_\_\_\_\_ error (can not used Aliyses Name in Where Condition) \_\_\_\_\_\_\_\_

select

departement,

salary,

salary/2 halfsalary

from employee1

where halfsalary < 35000 --- error bec -> halfSalary is not the part of table

--\_\_\_\_\_\_\_\_ 2nd Way \_\_\_\_\_\_\_\_ by using SubQuery inner query will become the part of outer Query

select

departement,

salary,

halfSalary

from (

select

departement,

salary,

salary/2 halfSalary

from employee1

) as emp

where halfSalary < 35000

**Find the names of employees who earn more than the average salary ?**

**overAllAvergeInCompany Emplyee -🡪 Saaa 🡪 zydaa kamana wala Employee**

--\_\_\_\_\_\_\_\_\_ 1st Way \_\_\_\_\_\_\_\_\_\_\_\_

SELECT employee\_name

FROM employees

WHERE (annual\_salary/12) > (SELECT AVG((annual\_salary/12)) FROM employees);

--\_\_\_\_\_\_\_\_\_\_ 2nd Way \_\_\_\_\_\_\_\_\_\_\_\_\_

SELECT

employee\_name

FROM (

select

employee\_name,

annual\_salary,

annual\_salary/12 month\_salary,

(SELECT AVG((annual\_salary/12)) FROM employees) as averageSalaryofAllEmployee

from employees

) as emp

WHERE (month\_salary) > (averageSalaryofAllEmployee);

**Example:**

create table orders(

id int primary key,

product varchar(25),

buyprice int,

salePrice int

)

insert into orders values (1,'Apple',80 , 85);

insert into orders values (2,'Mango',180 , 200);

insert into orders values (3,'Orange',100 , 125);

insert into orders values (4,'Apple',100 , 105);

insert into orders values (5,'Mango',100 , 95);

insert into orders values (4,'Apple',100 , 105);

insert into orders values (5,'Mango',100 , 95);

insert into orders values (6,'Apple',120 , 100);

insert into orders values (7,'Mango',110 , 95);

insert into orders values (8,'Orange',100 , 125);

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1st Way \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

select

product,

sum(buyprice),

sum(salePrice) ,

(sum(salePrice) - sum(buyprice)),

case

when (sum(salePrice)) - (sum(buyprice)) > 0 then 'Profit'

when (sum(salePrice)) - (sum(buyprice)) = 0 then 'Good'

else 'Lost'

end

from orders

group by product

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2nd way \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- overall products ke buyAmon , cell Amount nakal gy --------> takaa me profite nkaaal skon

select

product,

tSalePrice,

tBuyPrice,

lossProfit,

case

when lossProfit > 0 then 'Profit'

when lossProfit = 0 then 'Good'

else 'Lost'

end

from (

select

product,

sum(buyprice) as tSalePrice,

sum(salePrice) as tBuyPrice,

(sum(salePrice) - sum(buyprice)) as lossProfit

from orders

group by product

) as o

**Sub query Reduce Upper Query Complicity.**