SELECT
From
Where - condition clause row wise
Or, And, Not
Between
Distinct
Is, In
Like, %, _
Order By, Asc, Desc
Limit
Aggregate functions- min, max, sum, avg, count
Group By
Having - condition clause group wise
Subqueries
Any/All
Joins

Customer

CustomerID CustomerName Address City PostalCode Country Age

From
Select * from Customer
Select customerID, customerName from Customer

Where
Having is a condition clause. It checks row wise.
Select * from Customer where Country = “Bangladesh”

Or/And/Not
Select * from Customer where Country != “Bangladesh”(not)
Select * from Customer where Country <> “Bangladesh”(another symbol for not)
Select * from Customer where Country = “Bangladesh” And City = “Dhaka”(and)
Select * from Customer where Country = “Bangladesh” OR Country = “India”(and)

Between
Select * from Customer where Age >= 18 AND Age <= 25
We can write,
Select * from Customer where Age Between 18 and 25

Select * from Customer where Age >= 18 AND Age <= 25 not in the region. Then we can write,
Select * from Customer where Age NOT Between 18 and 25

Distinct
Select Distinct Country from Customer (for avoiding repetition, unique value)
Select Distinct Country, Distinct City from Customer

Is/In
Select * from Customer where Address is Null(for Null values we have to use is instead of =)
Select * from Customer where Address is Not Null

Select * from Customer where City = “Dhaka” or City = “Sylhet” or City = “Ctg”
We can write it as,
Select * from Customer where City in (“Dhaka” , “Sylhet”, “Ctg”)

Like
Select * from Customer where Address Like “Dhanmondi”
Its for match with the substrings
Because there are lot of info in address column as block no,flat no,road no, dhanmondi

%(percentage)
It replaces zero or more characters.
%dhanmondi means there can be zero or more characters but it doesnt matter.
It also means dhanmondi is the last word.
But if we use %dhanmondi% then the word can be anywhere in the middle and other characters doesn't matter

If we want to see address from dhanmondi and mohakhali we can write,
Select * from Customer where Address Like “%Dhanmondi%” or Address Like “%mohakhali%”

If we want to find a customer name that starts with A and ends with a we can write,
Select * from Customer where customerName Like “A%a”
So it will work for the name of Ayesha

_ (underscore)
It can ignore one character only.
Select * from Customer where customerName Like “A_a”
So it will work for the name of Ana
Select * from Customer where customerName Like “A__a”
So it will work for the name of Anna

Order by
Select * from Customer Order by Age desc, PostalCode Asc
From this it can order the age. Then if the age is the same then it will use postal code in asc order. That's how the first column will get the priority and then 2nd and then 3rd and on.

Limit
Select * from Customer Order by Age desc Limit 5
That's how we can see the 5 most senior customer

min/max/sum/avg/count
Select min(Age) from Customer
Select max(Age) from Customer
Select sum(Age) from Customer
Select avg(Age) from Customer
Select count(*) from Customer(it will count the total number of customers)

Group by
Select count(*) from Customer group by Country
It will give the no of customers per country

Select count(*),Country from Customer group by Country
It will give the country and then the no of customers per country

Select avg(Age),Country from Customer group by Country
It will give the avg age of each country

Having
Having is a condition clause. It checks group wise. It will check the condition in the entire group. Its normally used for group by or min/max/avg/sum/count informations

Select avg(Age),Country from Customer group by Country Having Count(*)>10
It will give the avg age of a country if that group have more than 10 customers

Subqueries
We use it when the value is unknown.this value usually changes.
Select * from Customer where Age = ?
Here we want to put the max age but it's unknown and everytime it changes the value if a new item inserts. So we use subqueries.
Select * from Customer where Age = (select max(age) from customer)
It will show the max age customer from all the customers

Select * from Customer where (Country,Age) = (Select Country,max(age) from Customer group by country
It will give the max age customer per country. So it will return multiple values. = sign returns one value. So we can't use = sign. So we will use In,
Select * from Customer where (Country,Age) In (Select Country,max(age) from Customer group by country

Lhs and rhs have to match. We can't write (city,age)or(age,country) in lhs and in rhs there is (country,age)

Any/All
Select * from Customer where Country = “Bangladesh” and Age < All(Select Age from Customer where Country = “India”)
First it will go to the bangladesh then in the age group then it will go in the indian age group. It will have to be less than all the Indian customer ages.

Select * from Customer where Country = “Bangladesh” and Age < Any(Select Age from Customer where Country = “India”)
First it will go to the bangladesh then in the age group then it will go in the indian age group. It will have to be less than at least one of the Indian customer age.

Joins
Didn’t give info. Class a bolse pore.

Sequence need to maintain:
1.Select
2.From
3.Where
4.Group by
5.Having
6.Order by

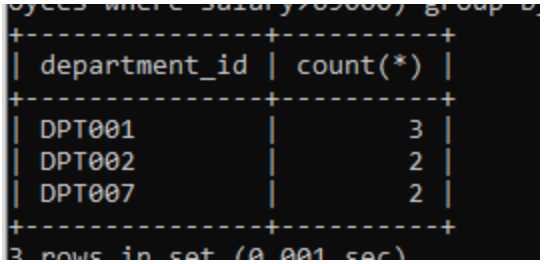
Lab 4 and lab 7 practice
Final = 1.45 - 2 h
Marks will be 40-45 converted to 30
4 questions. All have to answer
Chap = 5,7,8

LAB 3:
4.
select department_id, count(*) from employees where employee_id in (select employee_id from employees where salary>69000) group by department_id ;

```
MariaDB [labhome3]> select department_id, count(*) from employees where employee_id in (select employee_id from employee
s where salary>69000) group by department_id ;
+---------------+----------+
| department_id | count(*) |
+---------------+----------+
| DPT001        |        1 |
| DPT002        |        1 |
| DPT007        |        2 |
+---------------+----------+
3 rows in set (0.000 sec)
```

Here it check all employees so even tho it's grouping by dept still counting only the times it finds emp id on the nested

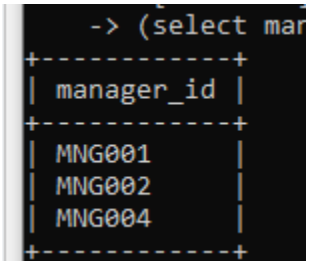
Otherwise;
select department_id, count(*) from employees where department_id in (select department_id from employees where salary>69000) group by department_id ;



```
+---------------+----------+
| department_id | count(*) |
+---------------+----------+
| DPT001        |        3 |
| DPT002        |        2 |
| DPT007        |        2 |
+---------------+----------+
```

checks each time it finds a dept

6.
select distinct manager_id from employees where (manager_id,employee_id) in (select manager_id,employee_id from employees where salary>55000);



```
    -> (select mar
+------------+
| manager_id |
+------------+
| MNG001     |
| MNG002     |
| MNG004     |
+------------+
```