Attendance → 5

HW → 5

Lab exam → 10

Project → 20

Wil be converted in 30

Data → comes from source, unorganised, data becomes fact if you do not use it, management system converts data into information, to build relationship between data we have to use RDMS. RDMS is a concept. MySQL is a language, using this we will RDMS

**Lab\_01**

**Command →**

sudo /opt/lampp/lampp start

Linux /opt/lampp/bin/mysql -h localhost -u root -p

Windows mysql -h localhost -u root -p

show databases;

create database cse370\_lab\_bad;

use cse370\_lab\_bad;

create table lab\_grades( —> then press enter to add column

create table lab\_grades(

-> student\_id char(4),

-> name varchar(30),

-> major char(3),

-> section char(1),

-> days\_present int,

-> project\_marks double,

-> cgpa decimal(3,2), -> submission\_date date);

describe lab\_grades;

show tables;

insert into lab\_grades values → press enter

select \* from lab\_grades;

alter table lab\_grades add atten\_marks int;

ALTER TABLE lab\_grades change COLUMN atten\_marks attendance\_marks int;

delete from lab\_grades where name = 'Farhana'; → you can use or, and to put extra condition

update lab\_grades set major = "CSE" where name = 'Arafat';

Select name, day\_present/15\*5 as attendance\_marks from lab\_grades where name = ‘Aarafat';

select upper(name) from lab\_grades;

Select distinct major from lab\_grades;

select \* from lab\_grades order by name asc, student\_id desc;

Select \* from lab\_grades where project\_marks between 15 and 17;

Update lab\_grades set major = ‘CSE';

Update lab\_grades set major = ‘ECE' where student\_id = ‘s001';

Update lab\_grades set major = ‘ECE' where student\_id in (‘s001' , ‘S004');

Select name from lab\_grades where submitted\_date <'2018-09-01';

Select name,major from lab\_grades where name like ‘a%';

Select name,major from lab\_grades where name like ‘%a%a%';

Student\_id char(4)

Student\_id = hedar name

Char = data type

4 = length

decimal(3,2) → decimal er left side e 3 length value and right side e 2 length er value deoa jabe

Double = floating

Date data type manipulation in sql

Alter and update for manipulating original table

Select to get a temporary table

Alter korle describe deoa lagebe

First e trable er description and select\* from lab\_grade korte hobe

**Lab 2**

* select max(cgpa) from lab\_grades;
* Select min(cgpa) as min\_cgpa from lab\_grades;
* select count(\*) from lab\_grades;
* select major, count(\*) from lab\_grades group by major;
* select max(submission\_date) from lab\_grades;
* select major, min(project\_marks) from lab\_grades group by major;
* select major, count(\*) from lab\_grades where project\_marks > 18 group by major;
* select major, count(\*) from lab\_grades where project\_marks > 18 group by major **having** major = 'CSE';
* select major, avg(project\_marks), count(\*) from lab\_grades where project\_marks > 17 group by major having count(\*)>1;
* select name from lab\_grades where cgpa = (select max(cgpa) from lab\_grades);
* select name from lab\_grades where project\_marks = (select min(project\_marks) from lab\_grades);
* select major, name, days\_present from lab\_grades where (major, days\_present) in (select major, min(days\_present) from lab\_grades group by major);
* select name from lab\_grades where major = 'CSE' and cgpa > any (select cgpa from lab\_grades where major = 'CS');
* select name from lab\_grades where major = 'CSE' and cgpa > all (select cgpa from lab\_grades where major = 'CS');
* select distinct major from lab\_grades l1 where exists (select \* from lab\_grades l2 where l2.major = l1.major and l2.cgpa<3.7);
* Select count(8) from lab\_grades where projec\_marks = (select max(project\_marks) from lab\_grades);
* select count(\*) from lab\_grades l1 where not exists (select \* from lab\_grades l2 where l1.student\_id != l2.student\_id and l2.project\_marks>l1.project\_marks);

**Lab 3**

1. Select student.student\_name, grade\_report.grade from student inner join grade\_report on grade\_report on student.student\_number = grade\_report.student\_number
2. Left join → left side er table ta ke priority dibe, right join → right ke priority dibe, inner join → duita table er common item niye table
3. Select s.student\_name, gp.grade, sc.course\_name from student s, grade\_report gp, section sc where student
4. Select c.customer\_id, c.customer\_name, c.customer\_city, a.account\_number from (customer c inner join depositor d on c.customer\_id = d.customer\_id) inner join account a on d.account\_number = a.account\_number –TASK 3
5. Select c.customer\_id, c.customer\_name, c.customer\_city, a.account\_number from customer c, depositor d, account a where c.customer\_id = d.customer\_id and d.account\_number = a.account\_number; → **it will only work for inner join – TASK 3**
6. Select c.customer\_name, c.customer\_city, a.account\_number, a.balance, a.branch\_name from customer c, depositor d, account a where c.customer\_id = d.customer\_id and d.account\_number = a.account\_number; –. TASK 5
7. Select c.customer\_name, c.customer\_city, l.branch\_name from customer c, borrower b, loan l where c.customer\_id = b.customer\_id and b.loan\_number = l.loan\_number and l.branch\_name=’Perryridge’; →**task 6**
8. Sequence
   1. Select
   2. from
   3. where
   4. and/or
   5. group by
   6. having
   7. order by
   8. Limit
9. select branch\_name, avg(balance) from account group by branch\_name having avg(balance) >= 700; → task 8
10. Select c.customer\_name from customer c, depositor d, account a where c.customer\_id = d.customer\_id and d.account\_number and a.branch\_name = Mianus
11. Select c.customer\_name from customer c, depositor d, account a where c.customer\_id = d.customer\_id and d.account\_number = a.account\_number and a.branch\_name = 'Mianus'

and customer\_name in

(select c.customer\_name from customer c, borrower b, loan l where c.customer\_id = b.customer\_id and b.loan\_number = l.loan\_number and l.branch\_name = 'Mianus')

1. Inner join, outer join, left join, right join