MySQL QUERIES:

1. **Create a database table student and it should have 5 columns   
    Id , name, branch, address, marks,teacher\_id  
   Insert 7 rows into it**

Ans1) create table testdb.student(

Id mediumint not null auto\_increment,

name varchar(10),

branch varchar(10),

address varchar(40),

marks int,

teacher\_id int,

primary key(Id));

alter table testdb.student modify column branch varchar(25);

insert into testdb.student(Id , name, branch, address, marks,teacher\_id) values(1,'samiksha','comp science','madhya pradesh',10,5);

insert into testdb.student(name, branch, address, marks,teacher\_id) values('sam','biology','mumbai',8,3),

('rohan','zoology','pune',10,1),

('sreejitha','chemistry','kerala',8,2),

('sid','electronics','mumbai',10,4),

('shravani','comp science','mumbai',8,5),

('monica','science','goa',7,2);

1. **Write a query to get the Id of students whose branch is = computer science(CS)**

Ans2) select Id from testdb.student where branch = 'comp science';

1. **Update a new column email into the table and insert values into it for all the students**

Ans3) alter table testdb.student add column email varchar(40);

update testdb.student set email ='samiksha1@gmail.com' where Id=1;

update testdb.student set email ='sam2@gmail.com' where Id=2;

update testdb.student set email ='rohan3@gmail.com' where Id=3;

update testdb.student set email ='sreejitha4@gmail.com' where Id=4;

update testdb.student set email ='sid5@gmail.com' where Id=5;

update testdb.student set email ='shravani6@gmail.com' where Id=6;

update testdb.student set email ='monica7@gmail.com' where Id=7;

1. **Display the name of all the students whose name ends with ‘a’**

Ans4) select name from testdb.student where name like '%a';

1. **Calculate the 5%of total marks of each student.**

Ans5) select name,(5/100)\*marks from testdb.student;

1. **Arrange the student according to their marks in descending order.**

Ans6) select \* from testdb.student order by marks desc;

1. **Select distinct branch and names**

Ans7) select distinct branch from testdb.student;

select distinct name from testdb.student;

1. **Create another table teachers and it should have 4 columns   
    teacher\_Id, name, branch,salary**

**Insert 7 rows in it**

Ans8) create table testdb.teachers(

teacher\_Id int not null auto\_increment,

name varchar(10),

branch varchar(25),

salary int,

primary key(teacher\_Id));

insert into testdb.teachers(teacher\_Id, name, branch,salary) values(1,'Rajesh','zoology',50000);

insert into testdb.teachers(name, branch,salary)

values('Nidhi','chemistry',42000),

('Savita','biology',30000),

('Sanjay','electronics',60000),

('Abhijeet','comp science',50000),

('Ravi','mechanical',54000),

('Sooraj','science',35000);

1. **Select the name of all the students and teachers according to their matching branch.**

Ans9) select teachers.name as teacher,student.name as student

from teachers,student where teachers.branch=student.branch;

select teachers.name as teacher,student.name as student

from teachers inner join student on teachers.branch=student.branch;

1. **Create any 2 stored procedure**

Ans10) delimiter //

create procedure max\_mks()

begin

select max(marks) from testdb.student;

end//

delimiter ;

delimiter $$

create procedure teacher\_list()

begin

select \* from testdb.teachers;

end$$

delimiter ;

1. **Find the min,max and average marks from the student table**

Ans11) select min(marks) from testdb.student;

select max(marks) from testdb.student;

select avg(marks) from testdb.student;

1. **Find the name of the student who has the 2nd highest marks in the batch and also the student who has the least marks**

Ans12) select name,marks from testdb.student where marks =(select max(marks) from testdb.student where marks < (select max(marks) from testdb.student));

select name,marks from testdb.student where marks=(select min(marks) from testdb.student);

1. **Find the average marks of students from each branch**

Ans13) select branch, avg(marks) from testdb.student group by branch;

1. **Calculate the monthly salary of the teachers if their annual salary is given in the table.**

Ans14) select name, salary/12 as monthly\_salary from testdb.teachers;

1. **Find which teacher belongs to which student(Q9)**
2. **Try to find the common data in both the tables**

Ans16) select teachers.name as teacher, student.name as student, teachers.branch

from teachers, student

where teachers.branch=student.branch AND teachers.teacher\_Id=student.teacher\_id;

1. **Try to create trigger where you have to update the column full name when you insert values in first name and last name column**

Ans17) alter table testdb.student rename column name to first\_name;

alter table testdb.student add column last\_name varchar(20);

create table testdb.student\_details(

Id mediumint not null auto\_increment,

full\_name varchar(40),

branch varchar(25),

primary key(Id));

create trigger update\_fullname

after update on student

for each row

insert into student\_details(Id,full\_name,branch)

values(old.Id,concat(old.first\_name,' ',new.last\_name),old.branch);

1. **Find the average salary of teachers according to the respective branch**

Ans18) select branch, avg(salary) from testdb.teachers group by branch;

1. **Update the marks of the students once there 2nd semester exams get over**

Ans19) update testdb.student set marks=150 where Id=1;

update testdb.student set marks=120 where Id=2;

update testdb.student set marks=155 where Id=3;

update testdb.student set marks=147 where Id=4;

update testdb.student set marks=168 where Id=5;

update testdb.student set marks=132 where Id=6;

update testdb.student set marks=156 where Id=7;

1. **Remove all the data from the tables**

Ans20) Delete from table testdb.student;

Delete from table testdb.teachers;