**DAY-10&11&12**

1. **Go through the concept on Group by, Having clause, Union, group by v/s distinct**

ANS:

* GROUP CLAUSE: It is used to group rows that have same values in result set.

Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

ORDER BY column\_name(s);

* HAVING CLAUSE: The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

* UNION: The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION must have the same number of columns

The columns must also have similar data types

The columns in every SELECT statement must also be in the same order

Syntax:

SELECT column\_name(s) FROM table1

UNION

SELECT column\_name(s) FROM table2;

* GROUP BY V/S DISTINCT:

The SELECT DISTINCT statement is used to return only distinct (different) values.Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

Syntax

SELECT DISTINCT column1, column2, ...

FROM table\_name;

If one of them is faster, it's going to be DISTINCT. This is because, although the two are the same, a query optimizer would have to catch the fact that your GROUP BY is not taking advantage of any group members, just their keys.

DISTINCT makes this explicit, so you can get away with a slightly dumber optimizer.

1. **Write a query to demonstrate the use of group by**

**ANS:**

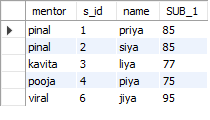
SELECT r.mentor,r.s\_id,s.name,s.SUB\_1

FROM priyanka.stu\_sheet s

inner join priyanka.stu\_records r

on r.s\_id=s.Id

GROUP BY (r.Id);



1. **Find the second highest marks scored by each student along with the student and subject name [Tables: student,subject,marks]**

**ANS: Table1(student)**

CREATE TABLE `priyanka`.`student` (

`id` INT NOT NULL,

`name` VARCHAR(45) NULL,

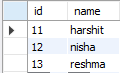
PRIMARY KEY (`id`));

insert into priyanka.student(id,name) values (11,' harshit');

insert into priyanka.student(id,name) values (12,' nisha');

insert into priyanka.student(id,name) values (13,'reshma');

SELECT \* FROM priyanka.student;



**Table2. subject**

CREATE TABLE `priyanka`.`subject` (

`id` INT NOT NULL,

`subjects` VARCHAR(45) NULL,

`stu\_id` INT NULL,

PRIMARY KEY (`id`));

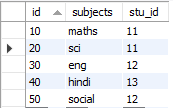
insert into priyanka.subject(id,sub1,sub2) values (10,'maths',11);

insert into priyanka.subject(id,sub1,sub2) values (20,'sci', 11);

insert into priyanka.subject(id,sub1,sub2) values (30, 'eng', 12);

insert into priyanka.subject(id,sub1,sub2) values (40,'hindi', 13);

insert into priyanka.subject(id,sub1,sub2) values (50, 'social', 12);



**TABLE3. marks**

CREATE TABLE `marks` (

`id` int NOT NULL,

`sid` int DEFAULT NULL,

`marks` int DEFAULT NULL,

PRIMARY KEY (`id`)

);

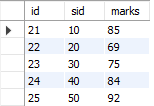
INSERT INTO marks (`id`, `sid `, `marks `) VALUES ('21',10,85);

INSERT INTO marks (`id`, `sid `, `marks `) VALUES ('22',20,69);

INSERT INTO marks (`id`, `sid `, `marks `) VALUES ('23',30,75);

INSERT INTO marks (`id`, `sid `, `marks `) VALUES ('24',40,84);

INSERT INTO marks (`id`, `sid `, `marks `) VALUES ('25',50,92);



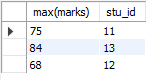
SELECT max(marks),stu\_id

FROM marks

WHERE marks NOT IN (SELECT max(marks)

FROM marks GROUP BY stu\_id)

GROUP BY stu\_id ;



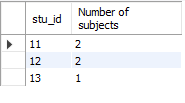
1. **Write a query to demonstrate the use of having clause along with aggregate functions.**

**ANS**: SELECT stu\_id,COUNT(\*) AS "Number of subjects"

FROM priyanka.subject

GROUP BY (stu\_id)

HAVING COUNT(\*)<11;



1. **Write two queries which help us to compare and understand the concept of distinct and group by**

**ANS: GROUP BY:**

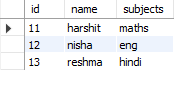
SELECT s.id,s.name,t.subjects

FROM priyanka.student s

inner join priyanka.subject t

on s.id=t.stu\_id

GROUP BY (s.id);



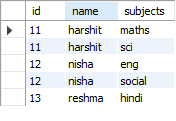
**DISTINCT:**

SELECT distinct (s.id),s.name,t.subjects

FROM priyanka.student s

inner join priyanka.subject t

on s.id=t.stu\_id;



1. **Write Query which includes union keyword**

**ANS:** select student.id from priyanka.student

union

select subject.stu\_id from priyanka.subject



1. **Write a single query which includes group by, having,where ,from ,select,union and order by.**

**ANS:** SELECT id,guide

FROM priyanka.subject

where id>10

GROUP BY (id)

union

SELECT sid,s\_guide

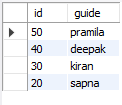
FROM priyanka.marks

where sid>10

GROUP BY (id)

having id<=50

order by (id) DESC;



1. **Write a query which captures duplicate records by the of union**

ANS: select s.id from priyanka.student s

union all

select t.stu\_id from priyanka.subject t;

