SHEET NO - 02

Name : Raksha Pahariya
Roll No : 2021CSB029
Subject : DBMS Lab

CREATING TABLES

```
CREATE TABLE depts (
   deptcode varchar(4) PRIMARY KEY,
   deptname varchar(50) NOT NULL
  );

CREATE TABLE students (
   rollno numeric(8) PRIMARY KEY,
   name varchar(30),
   bdate date CHECK (bdate < '2021-01-01'),
   deptcode char(4) REFERENCES depts (deptcode) ON DELETE CASCADE,
   hostel numeric CHECK (hostel < 10),
   parent_inc numeric(8, 1)
  );

CREATE TABLE faculty (
   fac_code varchar(8) PRIMARY KEY,
   fac_name varchar(30) NOT NULL,
   fac_dept varchar(5) REFERENCES depts (deptcode)
  );

CREATE TABLE crs_offrd (
   crs_code varchar(10) PRIMARY KEY,
   crs_name varchar(35) NOT NULL,
   crs_credits numeric(2, 1),
   crs_fac_cd varchar(8) REFERENCES faculty (fac_code)
  );
```

```
CREATE TABLE crs_regd (
   crs_rollno numeric(8) REFERENCES students (rollno),
   crs_cd char(10) REFERENCES crs_offrd (crs_code),
   marks numeric(5, 2),
   PRIMARY KEY (crs_rollno, crs_cd)
  );
```

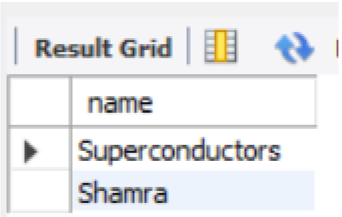
Assignment - 04

A . Retrieve the name of the students whose name starts with ‘S’ and contains ‘r’ as the second last character.

QUERY :

```
select name
from students
where name like 'S%r_';
```

OUTPUT :



name
Superconductors
Shamra

EXPLANATION :

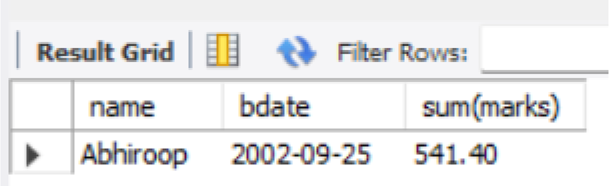
- ***S%: Matches any name that starts with 'S'.***
- ***r_: Matches any name where the second last character is 'r', and the underscore _ is a wildcard that matches any single character.***
- ***like doesn't work with char(10), but works with varchar(10)***

B . Retrieve the name of the youngest student(s) from the ‘CST’ department along with the total marks obtained by him (them).

QUERY :

```
select name, bdate, sum(marks)
from students, crs_regd
where crs_rollno = rollno
and deptcode = 'CST'
group by rollno
having bdate = ( select max(bdate)
   from students
   where deptcode = 'CST');
```

OUTPUT :



name	bdate	sum(marks)
Abhiroop	2002-09-25	541.40

EXPLANATION :

- *It filters the results to include only students in the 'CST' department using the deptcode = 'CST' condition.*
- *It groups the results by the rollno.*
- *It uses the HAVING clause to further filter the results. The subquery in the HAVING clause finds the maximum bdate for students in the 'CST' department, and the HAVING condition checks if the student's bdate matches this maximum bdate.*

C . Find the age of all the students.

QUERY :

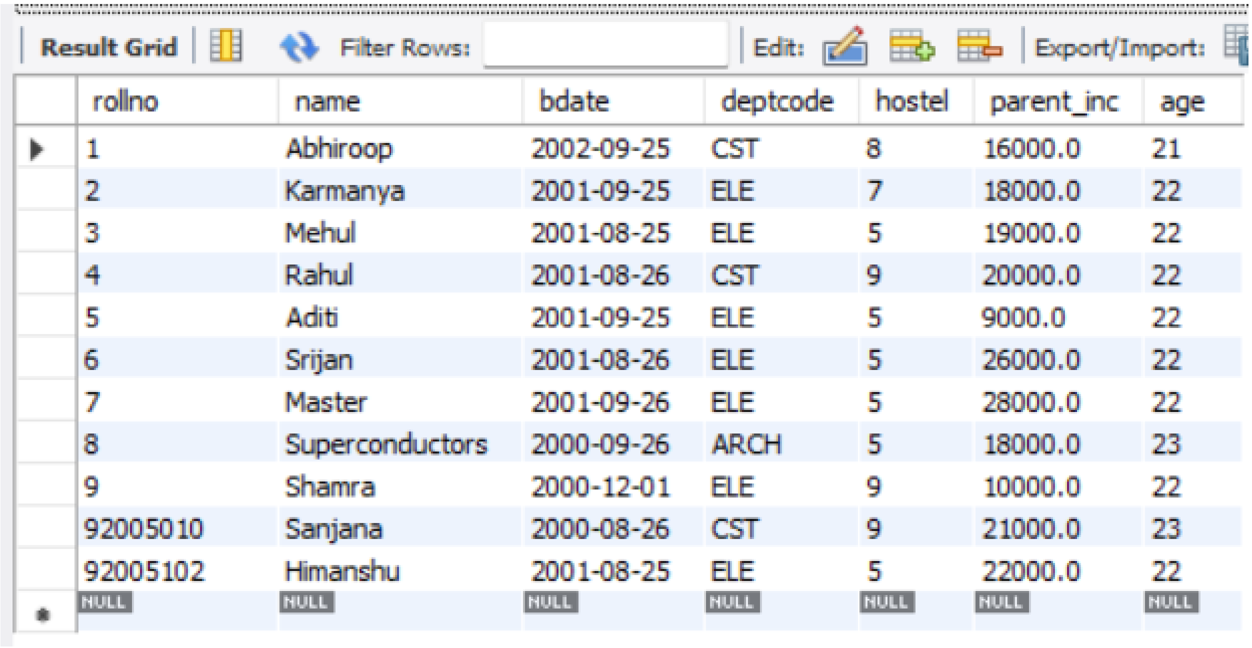
1. *alter table students add age numeric;*

2. *update students*
```
   set age = year(now()) - year(bdate)
   where month(now()) > month(bdate)
   or (month(now()) = month(bdate)
   and (date(now()) > date(bdate)
   or date(now()) = date(bdate)));
```

```
3.  update students
        set age = year(now()) - year(bdate) - 1
        where month(now()) < month(bdate)
        or (month(now()) = month(bdate)
        and date(now()) < date(bdate));

4.  select * from students;
```

OUTPUT :



rollno	name	bdate	deptcode	hostel	parent_inc	age
1	Abhiroop	2002-09-25	CST	8	16000.0	21
2	Karmanya	2001-09-25	ELE	7	18000.0	22
3	Mehul	2001-08-25	ELE	5	19000.0	22
4	Rahul	2001-08-26	CST	9	20000.0	22
5	Aditi	2001-09-25	ELE	5	9000.0	22
6	Srijan	2001-08-26	ELE	5	26000.0	22
7	Master	2001-09-26	ELE	5	28000.0	22
8	Superconductors	2000-09-26	ARCH	5	18000.0	23
9	Shamra	2000-12-01	ELE	9	10000.0	22
92005010	Sanjana	2000-08-26	CST	9	21000.0	23
92005102	Himanshu	2001-08-25	ELE	5	22000.0	22
NULL	NULL	NULL	NULL	NULL	NULL	NULL

EXPLANATION :

- *adds a new column named "age" to the "students" table with a data type of NUMERIC*
- *UPDATE statement calculates the age of each student in the "students" table based on their birthdate ("bdate") and updates the "age" column with the calculated value.*
- *If his/her birth month was before the current month or his/her birth month is ame as current month but the birthdate was before the current date or his/her birth month and birth date both are same as the current month and date, I subtracted the birth year from the current year as it means that he completed his birthday in the current year(2nd command in query).*

- *If his/her birth month is after the current month or his/her birth month is the same as the current month but the birthdate is after the current date, I subtracted the birth year from the current year and then subtracted 1 from it as he has not completed his birthday in the current year yet(2nd command in query).*

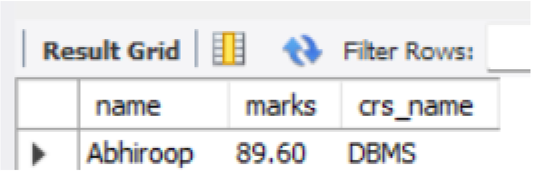
Assignment - 05

A . Retrieve the name of the student(s) who obtained second highest marks in ‘DBMS’.

QUERY :

```
select name, marks, crs_name
from students, crs_regd, crs_offrd
where rollno = crs_rollno
and crs_cd = crs_code
and crs_name = 'DBMS'
and marks = (select max(marks)
                 from crs_regd, crs_offrd
                 where crs_cd = crs_code
                 and crs_name = 'DBMS'
                 and marks < (select max(marks)
                                  from crs_regd, crs_offrd
                                  where crs_cd = crs_code
                                  and crs_name = 'DBMS'));
```

OUTPUT :



name	marks	crs_name
Abhiroop	89.60	DBMS

EXPLANATION :

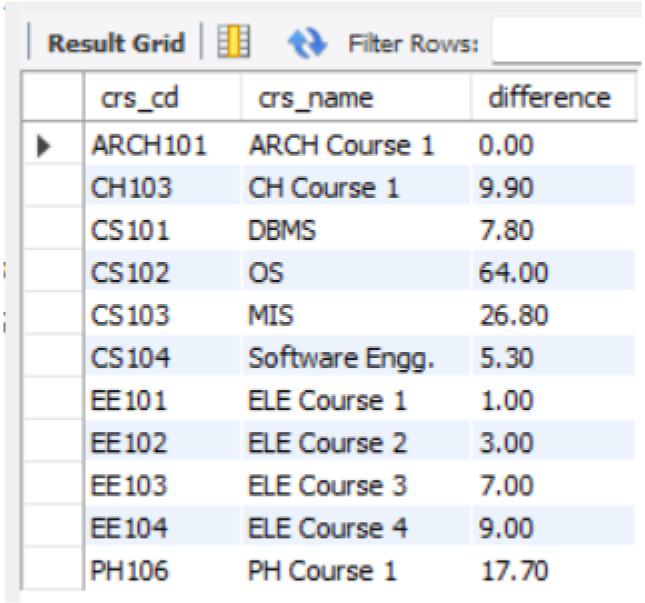
This query gives us a table having two columns: name of each student and the total number of courses taken up by the student.

B . Find out the differences between highest and lowest marks obtained in each subject.

QUERY :

```
select crs_cd, crs_name, max(marks)-min(marks) as difference
from crs_regd, crs_offrd
where crs_cd = crs_code
group by(crs_cd)
order by(crs_cd);
```

OUTPUT :



crs_cd	crs_name	difference
ARCH101	ARCH Course 1	0.00
CH103	CH Course 1	9.90
CS101	DBMS	7.80
CS102	OS	64.00
CS103	MIS	26.80
CS104	Software Engg.	5.30
EE101	ELE Course 1	1.00
EE102	ELE Course 2	3.00
EE103	ELE Course 3	7.00
EE104	ELE Course 4	9.00
PH106	PH Course 1	17.70

EXPLANATION :

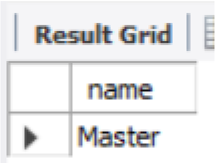
This query gives us a table having three columns: name of students, department of the student , course taken up by the student and department of the faculty teaching the course taken up by the student.

C . Assuming the existence of several interdepartmental courses, retrieve the name of the student(s) who is(are) studying under at least one faculty from each department.

QUERY :

```
select name
from students, crs_regd, crs_offrd, faculty
where rollno = crs_rollno and crs_cd = crs_code and crs_fac_cd = fac_code
group by(name)
having count(distinct fac_dept) = (select count(distinct fac_dept)
from faculty);
```

OUTPUT :



```
name
Master
```

EXPLANATION :

- *COUNT(DISTINCT fac_dept) calculates the count of distinct department codes associated with the faculty members who taught the courses taken by the students.*
- *(SELECT COUNT(DISTINCT fac_dept) FROM faculty) calculates the total count of distinct department codes in the "faculty" table.*

<u>D . Assuming the existence of several interdepartmental courses, retrieve the name of the student(s) who is(are) studying under the faculties only from his(their) own department.</u>

QUERY :

```
1. select name, count(distinct crs_cd) as courses_count
      from students, crs_regd
      where crs_rollno = rollno
      group by(rollno);

2. select name, deptcode, crs_cd, fac_dept from students, crs_regd,
      crs_offrd, faculty where crs_rollno = rollno and crs_cd = crs_code and
      crs_fac_cd = fac_code;

3. select name, count(deptcode) as
```

```
crs_taught_by_fac_of_dept_same_as_student_count
from (
select name, deptcode, crs_cd, fac_dept
from students, crs_regd, crs_offrd, faculty
where crs_rollno = rollno
and crs_cd = crs_code
and crs_fac_cd = fac_code
) as tbl1
where deptcode = fac_dept
group by(name);
```

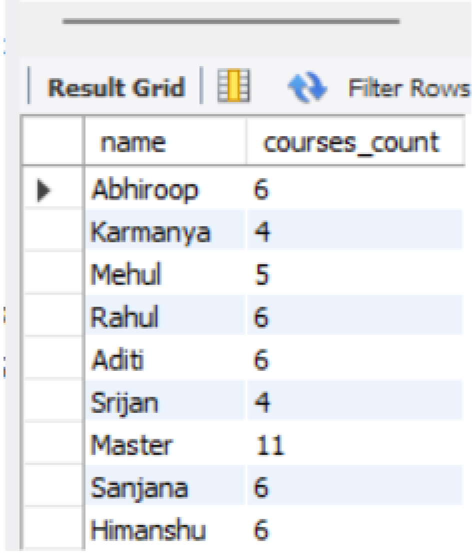
4.
   ```
   select s1.name, s1.crs_taught_by_fac_of_dept_same_as_student_count,
       s2.courses_count
       from (
       select name, count(deptcode) as
       crs_taught_by_fac_of_dept_same_as_student_count
       from (
       select name, deptcode, crs_cd, fac_dept
       from students, crs_regd, crs_offrd, faculty
       where crs_rollno = rollno
       and crs_cd = crs_code and crs_fac_cd = fac_code
       ) as tbl1
       where deptcode = fac_dept
       group by(name)
       ) as s1,
       (
       select name, count(distinct crs_cd) as courses_count
       from students, crs_regd
       where crs_rollno = rollno
       group by(rollno)
       ) as s2
       where s1.name = s2.name;
   ```

5.
   ```
   select name
       from (
       select s1.name,
       s1.crs_taught_by_fac_of_dept_same_as_student_count,
   ```

```
s2.courses_count
from (
select name, count(deptcode) as
crs_taught_by_fac_of_dept_same_as_student_count
from (
select name, deptcode, crs_cd, fac_dept
from students, crs_regd, crs_offrd, faculty
where crs_rollno = rollno
and crs_cd = crs_code and crs_fac_cd = fac_code
) as tbl1
where deptcode = fac_dept
group by(name)
) as s1,
(
select name, count(distinct crs_cd) as courses_count from students,
crs_regd where crs_rollno = rollno group by(rollno)
) as s2
where s1.name = s2.name
) as temp_table
where crs_taught_by_fac_of_dept_same_as_student_count =
courses_count;
```

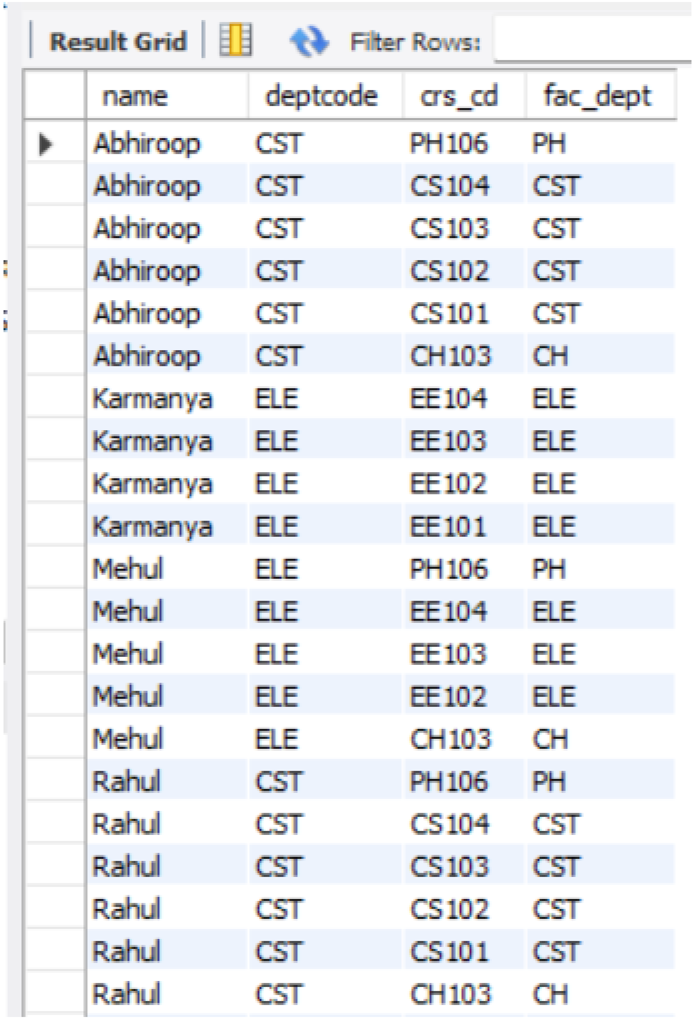
OUTPUT :

1.

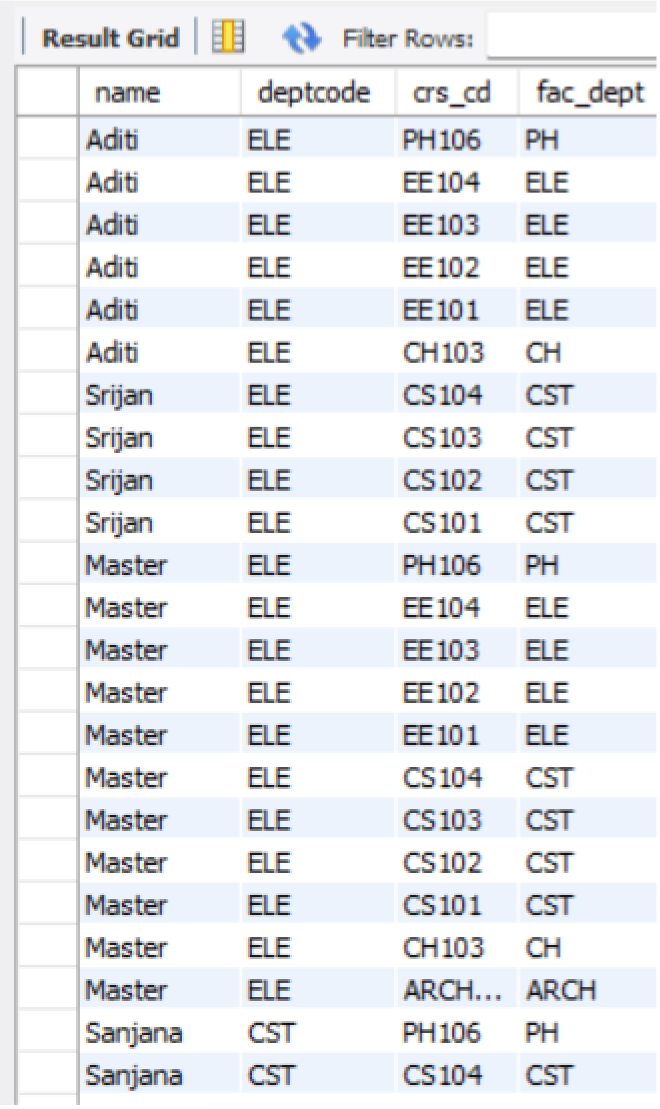


name	courses_count
Abhiroop	6
Karmanya	4
Mehul	5
Rahul	6
Aditi	6
Srijan	4
Master	11
Sanjana	6
Himanshu	6

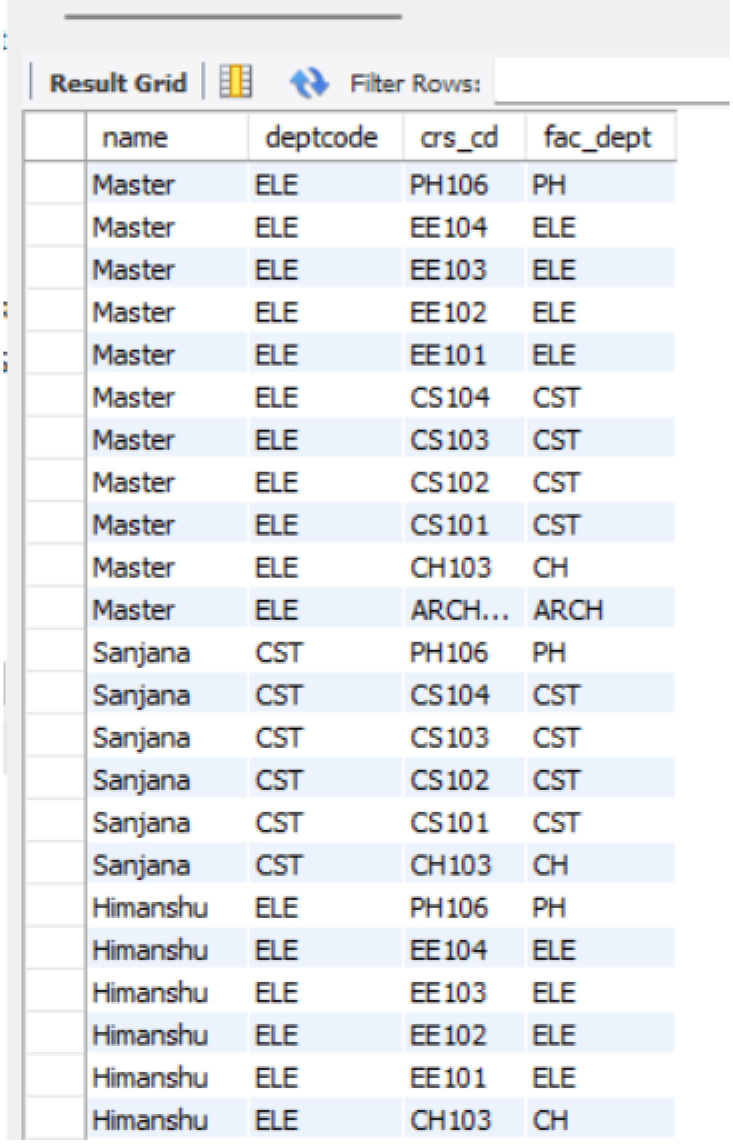
2.



name	deptcode	crs_cd	fac_dept
Abhiroop	CST	PH106	PH
Abhiroop	CST	CS104	CST
Abhiroop	CST	CS103	CST
Abhiroop	CST	CS102	CST
Abhiroop	CST	CS101	CST
Abhiroop	CST	CH103	CH
Karmanya	ELE	EE104	ELE
Karmanya	ELE	EE103	ELE
Karmanya	ELE	EE102	ELE
Karmanya	ELE	EE101	ELE
Mehul	ELE	PH106	PH
Mehul	ELE	EE104	ELE
Mehul	ELE	EE103	ELE
Mehul	ELE	EE102	ELE
Mehul	ELE	CH103	CH
Rahul	CST	PH106	PH
Rahul	CST	CS104	CST
Rahul	CST	CS103	CST
Rahul	CST	CS102	CST
Rahul	CST	CS101	CST
Rahul	CST	CH103	CH

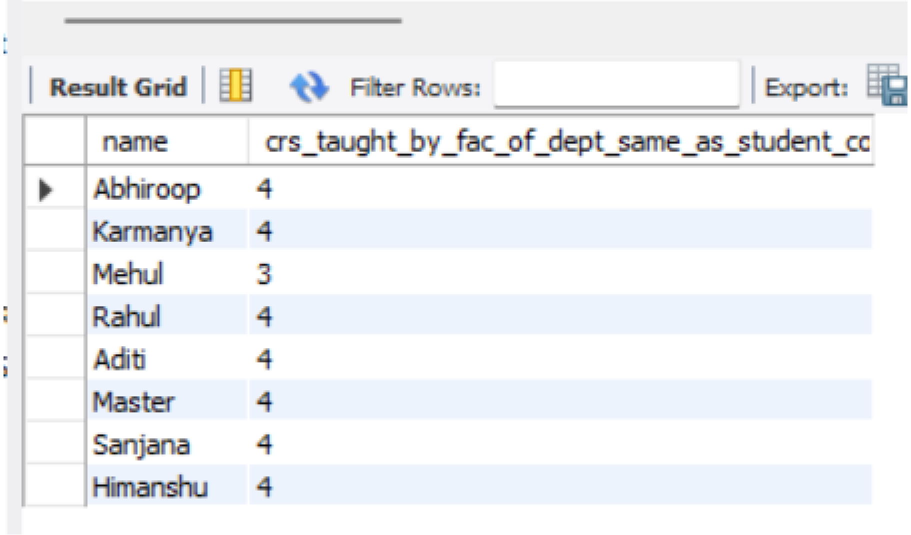


name	deptcode	crs_cd	fac_dept
Aditi	ELE	PH106	PH
Aditi	ELE	EE104	ELE
Aditi	ELE	EE103	ELE
Aditi	ELE	EE102	ELE
Aditi	ELE	EE101	ELE
Aditi	ELE	CH103	CH
Srijan	ELE	CS104	CST
Srijan	ELE	CS103	CST
Srijan	ELE	CS102	CST
Srijan	ELE	CS101	CST
Master	ELE	PH106	PH
Master	ELE	EE104	ELE
Master	ELE	EE103	ELE
Master	ELE	EE102	ELE
Master	ELE	EE101	ELE
Master	ELE	CS104	CST
Master	ELE	CS103	CST
Master	ELE	CS102	CST
Master	ELE	CS101	CST
Master	ELE	CH103	CH
Master	ELE	ARCH...	ARCH
Sanjana	CST	PH106	PH
Sanjana	CST	CS104	CST



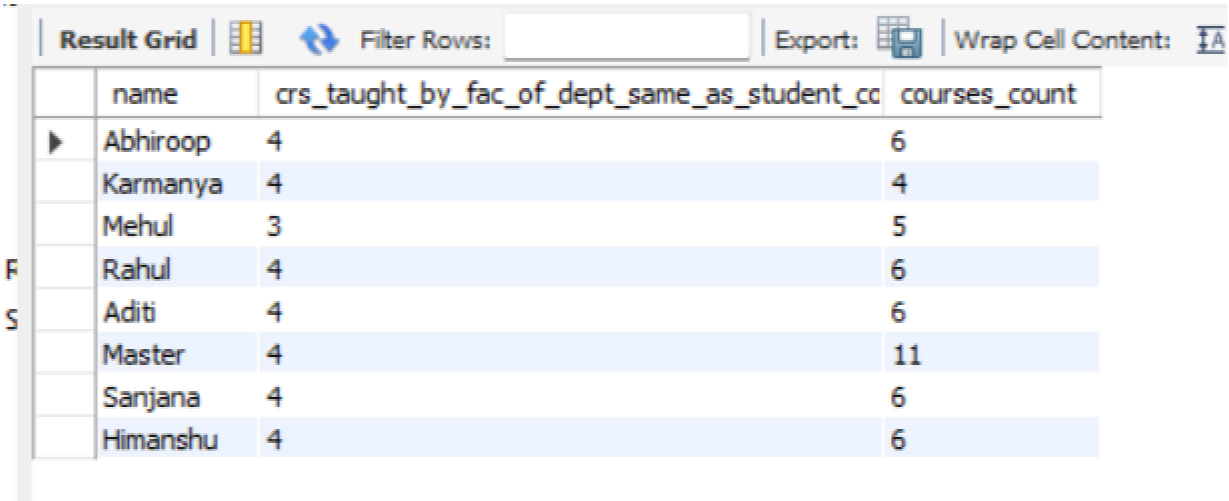
name	deptcode	crs_cd	fac_dept
Master	ELE	PH106	PH
Master	ELE	EE104	ELE
Master	ELE	EE103	ELE
Master	ELE	EE102	ELE
Master	ELE	EE101	ELE
Master	ELE	CS104	CST
Master	ELE	CS103	CST
Master	ELE	CS102	CST
Master	ELE	CS101	CST
Master	ELE	CH103	CH
Master	ELE	ARCH...	ARCH
Sanjana	CST	PH106	PH
Sanjana	CST	CS104	CST
Sanjana	CST	CS103	CST
Sanjana	CST	CS102	CST
Sanjana	CST	CS101	CST
Sanjana	CST	CH103	CH
Himanshu	ELE	PH106	PH
Himanshu	ELE	EE104	ELE
Himanshu	ELE	EE103	ELE
Himanshu	ELE	EE102	ELE
Himanshu	ELE	EE101	ELE
Himanshu	ELE	CH103	CH

3.



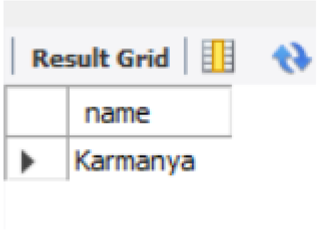
name	crs_taught_by_fac_of_dept_same_as_student_cc
Abhiroop	4
Karmanya	4
Mehul	3
Rahul	4
Aditi	4
Master	4
Sanjana	4
Himanshu	4

4.



name	crs_taught_by_fac_of_dept_same_as_student_cc	courses_count
Abhiroop	4	6
Karmanya	4	4
Mehul	3	5
Rahul	4	6
Aditi	4	6
Master	4	11
Sanjana	4	6
Himanshu	4	6

5.



name
Karmanya

EXPLANATION :

- *We use the tables 1 and 3 shown in the output and combine them in this query*

- *Finally, we compare the total number of courses taken by each student to the total number of courses taken up by the student which are taught by the faculty whose department is the same as that of the student and give the names of the students who study under faculty from their own department (students who have values of the above compared columns as equal).*

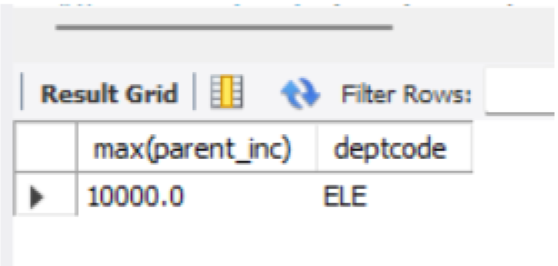
Assignment - 06

A . Display the highest parent incomes, in descending order, for each department excluding ‘ARCH’ such that only those highest parent incomes will appear that are below 12,000

QUERY :

```
select max(parent_inc), deptcode
from students
where parent_inc < 12000 and deptcode != 'ARCH'
group by(deptcode)
order by(max(parent_inc)) desc;
```

OUTPUT :



max(parent_inc)	deptcode
10000.0	ELE

EXPLANATION :

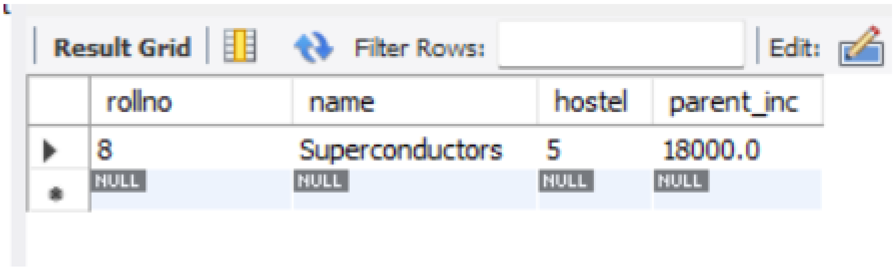
- *This query retrieves the highest parent incomes for each department except 'ARCH'.*
- *It filters out rows where the parent income is below 12,000 and the department is not 'ARCH'.*
- *The GROUP BY clause groups the results by the "deptcode" column, so you get the maximum parent income for each department.*
- *Finally, it orders the results in descending order of the maximum parent income.*

B . Retrieve the fifth highest parent income for hostel number 5.

QUERY :

```
select s1.rollno, s1.name, hostel, s1.parent_inc
from students s1
where hostel = 5 and 4 = (select count(s2.parent_inc)
from students s2
where s2.hostel = s1.hostel
and s2.parent_inc > s1.parent_inc);
```

OUTPUT :



rollno	name	hostel	parent_inc
8	Superconductors	5	18000.0
NULL	NULL	NULL	NULL

EXPLANATION :

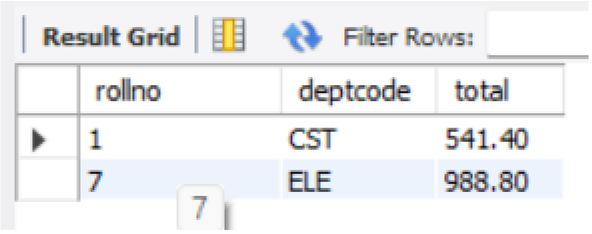
- *This query is designed to find the fifth highest parent income for students in hostel number 5.*
- *It uses a subquery to count how many students in the same hostel have a higher parent income than the current student (s1).*
- *If there are exactly four students with higher parent incomes, it means the current student has the fifth highest parent income in the hostel.*
- *The query retrieves the roll number, name, hostel, and parent income of students who meet this condition.*

C . Find the roll number of the students from each department who obtained highest total marks in their own department.

QUERY :

```
select rollno, deptcode, total
from (
select rollno, deptcode, sum(marks) as total
from students, crs_regd
where rollno = crs_rollno
group by(rollno)
) as tbl1
where total = (select max(total)
from (
select rollno, deptcode, sum(marks) as total
from students, crs_regd
where rollno = crs_rollno
group by(rollno)
) as tbl2
where tbl1.deptcode = tbl2.deptcode
group by(deptcode))
order by(rollno);
```

OUTPUT :



rollno	deptcode	total
1	CST	541.40
7	ELE	988.80

EXPLANATION :

- *This query finds the roll numbers of students from each department who obtained the highest total marks in their own department.*
- *It uses two levels of subqueries to achieve this. The inner subquery calculates the total marks for each student by joining the "students" and "crs_regd" tables and grouping by roll number.*
- *The outer subquery compares the total marks of each student with the maximum total marks within their department.*
- *Finally, the query orders the results by roll number.*