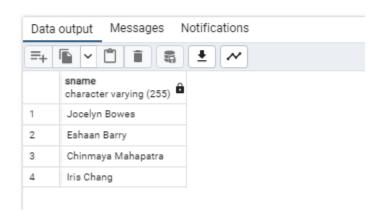# LAB6

## Simran

## 100377444

Q1: Display the name of students who enrolled in at least one section.

ANS: SELECT DISTINCT student.sName

FROM student JOIN std_enrolls_sec

ON std_enrolls_sec.sid = student.sid;

```
SELECT DISTINCT student.sName
FROM student JOIN std_enrolls_sec
ON std_enrolls_sec.sid = student.sid;
```

Data output    Messages    Notifications

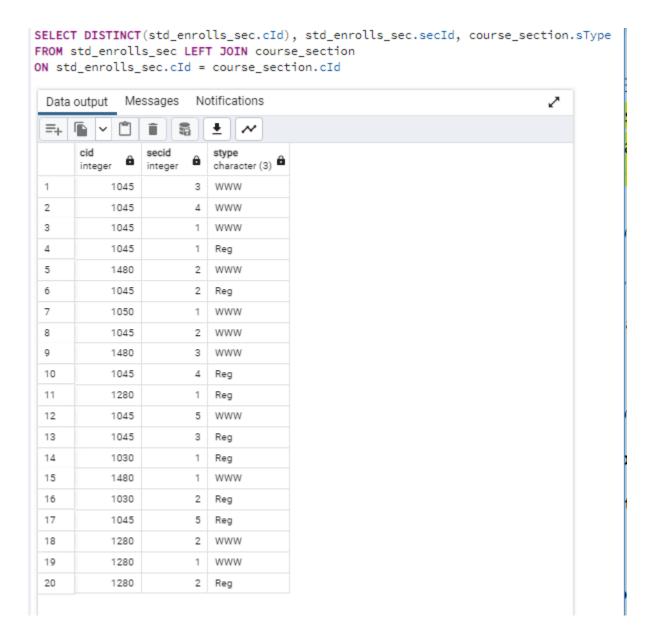| | sname<br>character varying (255) 🔒 |
|---|---|
| 1 | Jocelyn Bowes |
| 2 | Eshaan Barry |
| 3 | Chinmaya Mahapatra |
| 4 | Iris Chang |

Q2: Retrieve Course ID, section ID, and Section Type of the sections that have been enrolled by at least one student

ANS:  SELECT DISTINCT(std_enrolls_sec.cId), std_enrolls_sec.secId, course_section.sType

FROM std_enrolls_sec LEFT JOIN course_section

ON std_enrolls_sec.cId = course_section.cId;

```
SELECT DISTINCT(std_enrolls_sec.cId), std_enrolls_sec.secId, course_section.sType
FROM std_enrolls_sec LEFT JOIN course_section
ON std_enrolls_sec.cId = course_section.cId
```

Data output    Messages    Notifications

| | cid<br>integer | secid<br>integer | stype<br>character (3) |
|---|---|---|---|
| 1 | 1045 | 3 | WWW |
| 2 | 1045 | 4 | WWW |
| 3 | 1045 | 1 | WWW |
| 4 | 1045 | 1 | Reg |
| 5 | 1480 | 2 | WWW |
| 6 | 1045 | 2 | Reg |
| 7 | 1050 | 1 | WWW |
| 8 | 1045 | 2 | WWW |
| 9 | 1480 | 3 | WWW |
| 10 | 1045 | 4 | Reg |
| 11 | 1280 | 1 | Reg |
| 12 | 1045 | 5 | WWW |
| 13 | 1045 | 3 | Reg |
| 14 | 1030 | 1 | Reg |
| 15 | 1480 | 1 | WWW |
| 16 | 1030 | 2 | Reg |
| 17 | 1045 | 5 | Reg |
| 18 | 1280 | 2 | WWW |
| 19 | 1280 | 1 | WWW |
| 20 | 1280 | 2 | Reg |

Q3: Display CIS, SecID, and SType of all sections (no matter if a student has enrolled in that section or not) and the name of students who enrolled in that section.

ANS: SELECT DISTINCT course_section.cId, course_section.secId, course_section.sType, student.sName

FROM std_enrolls_sec, course_section, student

WHERE std_enrolls_sec.cId = course_section.cId

AND std_enrolls_sec.sId = student.sId;

```
SELECT DISTINCT course_section.cId, course_section.secId, course_section.sType, student.sName
FROM std_enrolls_sec, course_section, student
WHERE std_enrolls_sec.cId = course_section.cId
AND std_enrolls_sec.sId = student.sId
```
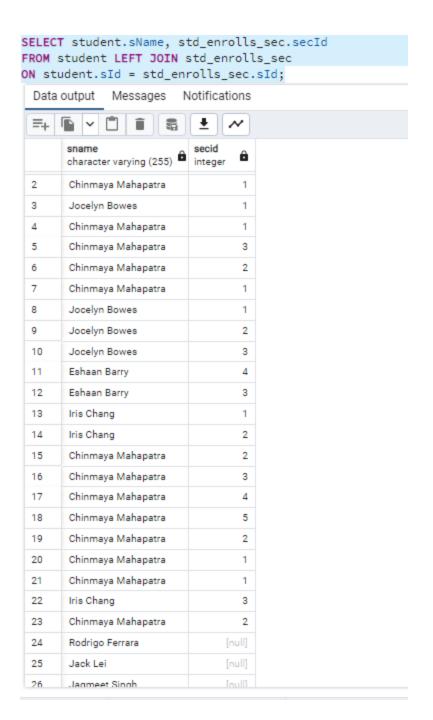
Data output    Messages    Notifications

| | cid integer | secid integer | stype character (3) | sname character varying (255) |
|---|---|---|---|---|
| 1 | 1030 | 1 | Reg | Chinmaya Mahapatra |
| 2 | 1030 | 1 | Reg | Iris Chang |
| 3 | 1030 | 2 | Reg | Chinmaya Mahapatra |
| 4 | 1030 | 2 | Reg | Iris Chang |
| 5 | 1045 | 1 | WWW | Chinmaya Mahapatra |
| 6 | 1045 | 1 | WWW | Eshaan Barry |
| 7 | 1045 | 1 | WWW | Iris Chang |
| 8 | 1045 | 1 | WWW | Jocelyn Bowes |
| 9 | 1045 | 2 | WWW | Chinmaya Mahapatra |
| 10 | 1045 | 2 | WWW | Eshaan Barry |
| 11 | 1045 | 2 | WWW | Iris Chang |
| 12 | 1045 | 2 | WWW | Jocelyn Bowes |
| 13 | 1045 | 3 | WWW | Chinmaya Mahapatra |
| 14 | 1045 | 3 | WWW | Eshaan Barry |
| 15 | 1045 | 3 | WWW | Iris Chang |
| 16 | 1045 | 3 | WWW | Jocelyn Bowes |
| 17 | 1045 | 4 | WWW | Chinmaya Mahapatra |
| 18 | 1045 | 4 | WWW | Eshaan Barry |
| 19 | 1045 | 4 | WWW | Iris Chang |
| 20 | 1045 | 4 | WWW | Jocelyn Bowes |
| 21 | 1045 | 5 | Reg | Chinmaya Mahapatra |
| 22 | 1045 | 5 | Reg | Eshaan Barry |
| 23 | 1045 | 5 | Reg | Iris Chang |
| 24 | 1045 | 5 | Reg | Jocelyn Bowes |

Q4: Display the name of all students(no matter if they have enrolled in any section) and the section number of all sections that they have enrolled.

ANS:  SELECT student.sName, std_enrolls_sec.secId

FROM student LEFT JOIN std_enrolls_sec

ON student.sId = std_enrolls_sec.sId;

```sql
SELECT student.sName, std_enrolls_sec.secId
FROM student LEFT JOIN std_enrolls_sec
ON student.sId = std_enrolls_sec.sId;
```

Data output   Messages   Notifications

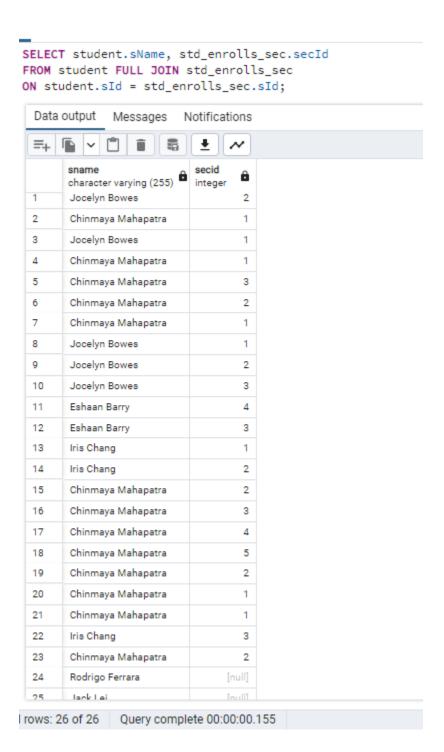| | sname<br>character varying (255) 🔒 | secid<br>integer 🔒 |
|---|---|---|
| 2 | Chinmaya Mahapatra | 1 |
| 3 | Jocelyn Bowes | 1 |
| 4 | Chinmaya Mahapatra | 1 |
| 5 | Chinmaya Mahapatra | 3 |
| 6 | Chinmaya Mahapatra | 2 |
| 7 | Chinmaya Mahapatra | 1 |
| 8 | Jocelyn Bowes | 1 |
| 9 | Jocelyn Bowes | 2 |
| 10 | Jocelyn Bowes | 3 |
| 11 | Eshaan Barry | 4 |
| 12 | Eshaan Barry | 3 |
| 13 | Iris Chang | 1 |
| 14 | Iris Chang | 2 |
| 15 | Chinmaya Mahapatra | 2 |
| 16 | Chinmaya Mahapatra | 3 |
| 17 | Chinmaya Mahapatra | 4 |
| 18 | Chinmaya Mahapatra | 5 |
| 19 | Chinmaya Mahapatra | 2 |
| 20 | Chinmaya Mahapatra | 1 |
| 21 | Chinmaya Mahapatra | 1 |
| 22 | Iris Chang | 3 |
| 23 | Chinmaya Mahapatra | 2 |
| 24 | Rodrigo Ferrara | [null] |
| 25 | Jack Lei | [null] |
| 26 | Jagmeet Singh | [null] |

Q5: Display the name of all students and all sections they have enrolled (no matter if a student has enrolled in any section and no matter if a section was enrolled by any student)

ANS:  SELECT student.sName, std_enrolls_sec.secId

FROM student FULL JOIN std_enrolls_sec

ON student.sId = std_enrolls_sec.sId;

```sql
SELECT student.sName, std_enrolls_sec.secId
FROM student FULL JOIN std_enrolls_sec
ON student.sId = std_enrolls_sec.sId;
```

Data output   Messages   Notifications

| | sname<br>character varying (255) 🔒 | secid<br>integer 🔒 |
|---|---|---|
| 1 | Jocelyn Bowes | 2 |
| 2 | Chinmaya Mahapatra | 1 |
| 3 | Jocelyn Bowes | 1 |
| 4 | Chinmaya Mahapatra | 1 |
| 5 | Chinmaya Mahapatra | 3 |
| 6 | Chinmaya Mahapatra | 2 |
| 7 | Chinmaya Mahapatra | 1 |
| 8 | Jocelyn Bowes | 1 |
| 9 | Jocelyn Bowes | 2 |
| 10 | Jocelyn Bowes | 3 |
| 11 | Eshaan Barry | 4 |
| 12 | Eshaan Barry | 3 |
| 13 | Iris Chang | 1 |
| 14 | Iris Chang | 2 |
| 15 | Chinmaya Mahapatra | 2 |
| 16 | Chinmaya Mahapatra | 3 |
| 17 | Chinmaya Mahapatra | 4 |
| 18 | Chinmaya Mahapatra | 5 |
| 19 | Chinmaya Mahapatra | 2 |
| 20 | Chinmaya Mahapatra | 1 |
| 21 | Chinmaya Mahapatra | 1 |
| 22 | Iris Chang | 3 |
| 23 | Chinmaya Mahapatra | 2 |
| 24 | Rodrigo Ferrara | [null] |
| 25 | Jack Lei | [null] |

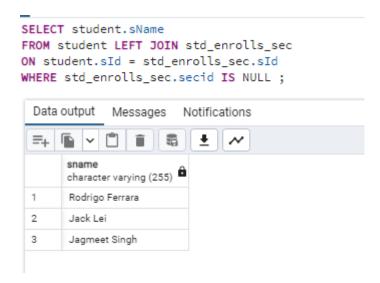I rows: 26 of 26    Query complete 00:00:00.155

Q6: Display the name of student who have not enrolled in any section?

ANS: SELECT student.sName

FROM student LEFT JOIN std_enrolls_sec

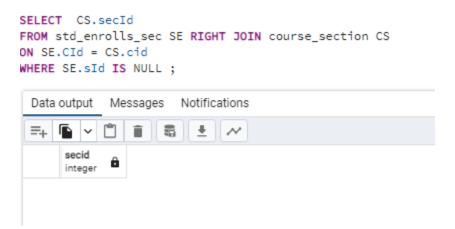ON student.sId = std_enrolls_sec.sId

WHERE std_enrolls_sec.secid IS NULL ;

```
SELECT student.sName
FROM student LEFT JOIN std_enrolls_sec
ON student.sId = std_enrolls_sec.sId
WHERE std_enrolls_sec.secid IS NULL ;
```

Data output   Messages   Notifications

| | sname<br>character varying (255) 🔒 |
|---|---|
| 1 | Rodrigo Ferrara |
| 2 | Jack Lei |
| 3 | Jagmeet Singh |

Q7: Display the name of section which has not be enrolled by any student?

ANS: SELECT CS.secId

FROM std_enrolls_sec  SE RIGHT JOIN course_section CS

ON SE.cId = CS.cId
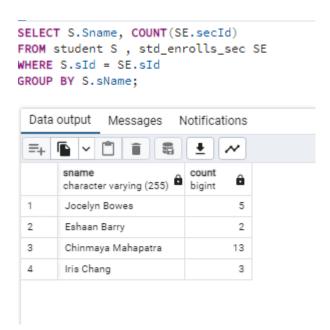
WHERE SE.sId IS NULL;

```
SELECT  CS.secId
FROM std_enrolls_sec SE RIGHT JOIN course_section CS
ON SE.CId = CS.cid
WHERE SE.sId IS NULL ;
```

Data output   Messages   Notifications

| | secid<br>integer 🔒 |
|---|---|
| | |

Q8: Retrieve the name of student and number of sections that they have enrolled

ANS: SELECT S.Sname, COUNT(SE.secId)

FROM student S , std_enrolls_sec SE

WHERE S.sId = SE.sId

GROUP BY S.sName;

```
SELECT S.Sname, COUNT(SE.secId)
FROM student S , std_enrolls_sec SE
WHERE S.sId = SE.sId
GROUP BY S.sName;
```

Data output   Messages   Notifications

| | sname<br>character varying (255) 🔒 | count<br>bigint 🔒 |
|---|---|---|
| 1 | Jocelyn Bowes | 5 |
| 2 | Eshaan Barry | 2 |
| 3 | Chinmaya Mahapatra | 13 |
| 4 | Iris Chang | 3 |

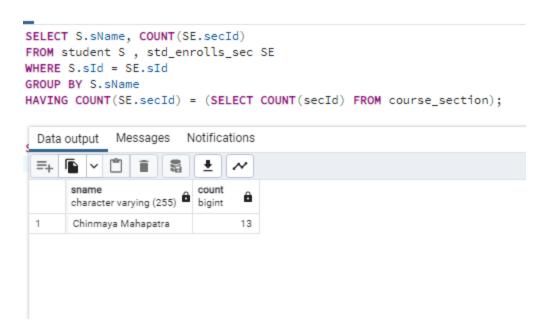Q9: Retrieve the name of students who have taken all courses

ANS: SELECT S.sName

FROM student S , std_enrolls_sec SE

WHERE S.sId = SE.sId

GROUP BY S.sName

HAVING COUNT(SE.secId) = (SELECT COUNT(secId) FROM course_section);

```
SELECT S.sName, COUNT(SE.secId)
FROM student S , std_enrolls_sec SE
WHERE S.sId = SE.sId
GROUP BY S.sName
HAVING COUNT(SE.secId) = (SELECT COUNT(secId) FROM course_section);
```

Data output    Messages    Notifications

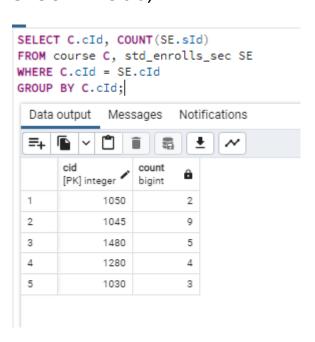| sname character varying (255) | count bigint |
|---|---|
| 1 | Chinmaya Mahapatra | 13 |

Q10: Retrieve the CID of courses and the number of students who enrolled in those sections

ANS: SELECT C.cId, COUNT(SE.sId)

FROM course C, std_enrolls_sec SE

WHERE C.cId = SE.cId

GROUP BY C.cId;

```
SELECT C.cId, COUNT(SE.sId)
FROM course C, std_enrolls_sec SE
WHERE C.cId = SE.cId
GROUP BY C.cId;
```

Data output    Messages    Notifications

| | cid [PK] integer | count bigint |
|---|---|---|
| 1 | 1050 | 2 |
| 2 | 1045 | 9 |
| 3 | 1480 | 5 |
| 4 | 1280 | 4 |
| 5 | 1030 | 3 |

Q11: Retrieve the name of students who are classmates with 'Chinmaya Mahapatra' in all sections that he has enrolled.

ANS: SELECT DISTINCT S.sName

FROM student S, std_enrolls_sec SE

WHERE S.sId = SE.sId
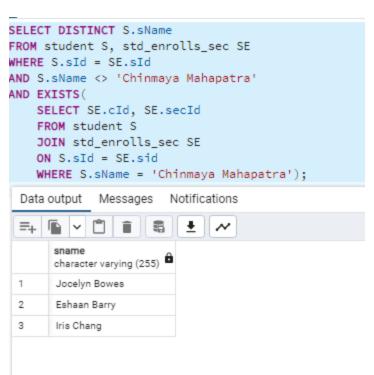
AND S.sName <> 'Chinmaya Mahapatra'

AND EXISTS(

    SELECT SE.cId, SE.secId

    FROM student S

    JOIN std_enrolls_sec SE

    ON S.sId = SE.sid

    WHERE S.sName = 'Chinmaya Mahapatra');

```
SELECT DISTINCT S.sName
FROM student S, std_enrolls_sec SE
WHERE S.sId = SE.sId
AND S.sName <> 'Chinmaya Mahapatra'
AND EXISTS(
    SELECT SE.cId, SE.secId
    FROM student S
    JOIN std_enrolls_sec SE
    ON S.sId = SE.sid
    WHERE S.sName = 'Chinmaya Mahapatra');
```

Data output    Messages    Notifications

| | sname<br>character varying (255) |
|---|---|
| 1 | Jocelyn Bowes |
| 2 | Eshaan Barry |
| 3 | Iris Chang |

Q12: Retrieve the name of all students who enrolled in 1045-001

ANS: SELECT S.sName

FROM student S, std_enrolls_sec SE

WHERE S.sId = SE.sId

AND SE.cId = '1045'

AND SE.secId = '1';

```sql
SELECT S.sName
FROM student S, std_enrolls_sec SE
WHERE S.sId = SE.sId
AND SE.cId = '1045'
AND SE.secId = '1';
```

Data output    Messages    Notifications

| | sname<br>character varying (255) 🔒 |
|---|---|
| 1 | Chinmaya Mahapatra |
| 2 | Jocelyn Bowes |