

Q1.

The screenshot shows a PostgreSQL query editor with a query window titled "students-data" and a query tab "<postgres> query". The query is as follows:

```
SELECT s.student_id, s.student_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE c.course_name = 'Math';
```

Below the query editor, a results window titled "students 1" shows the results of the query. The results are displayed in a grid with columns "student_id" and "student_name". The results are:

student_id	student_name
1	Alice
3	Charlie
4	David
6	Frank

Q2.

The screenshot shows a PostgreSQL query editor with a query window titled "students-data" and a query tab "<postgres> query". The query is as follows:

```
SELECT c.course_id, c.course_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE s.student_name = 'Bob';
```

Below the query editor, a results window titled "courses 1" shows the results of the query. The results are displayed in a grid with columns "course_id" and "course_name". The results are:

course_id	course_name
102	Science

Q3.

students-data | *<postgres> query X

```
SELECT s.student_id, s.student_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
GROUP BY s.student_id, s.student_name
HAVING COUNT(e.course_id) > 1;
```

students 1 X

SELECT s.student_id, s.student_name | Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_name
1	4	David
2	3	Charlie
3	1	Alice

Refresh Save Cancel Export data 200 3

Q4.

students-data | *<postgres> query X

```
SELECT s.student_id, s.student_name
FROM Students s
WHERE s.student_grade_id = 1;
```

students 1 X

SELECT s.student_id, s.student_name | Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_name
1	1	Alice
2	3	Charlie
3	5	Eve
4	8	Henry

Q5.

students-data *postgres> query x

```

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS student_count
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name;

```

courses 1 x

SELECT c.course_id, c.course_name, CO Enter a SQL expression to filter results (use Ctrl+Space)

	course_id	course_name	student_count
1	101	Math	4
2	103	History	2
3	102	Science	4

Q6.

students-data *postgres> query x

```

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS enrollment_count
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name
ORDER BY enrollment_count DESC
LIMIT 1;

```

courses 1 x

SELECT c.course_id, c.course_name, CO Enter a SQL expression to filter results (use Ctrl+Space)

	course_id	course_name	enrollment_count
1	101	Math	4

Q7.

students-data * <postgres> query x

```
SELECT s.student_id, s.student_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
GROUP BY s.student_id, s.student_name
HAVING COUNT(DISTINCT e.course_id) = (SELECT COUNT(course_id) FROM Courses);
```

students 1 x

SELECT s.student_id, s.student_name | Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_name

Q8.

students-data * <postgres> query x

```
SELECT s.student_id, s.student_name
FROM Students s
LEFT JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.enrollment_id IS NULL;
```

students 1 x

SELECT s.student_id, s.student_name | Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_name
1	10	Jack
2	8	Henry
3	9	Ivy

Q9.

students-data * <postgres> query x

```

SELECT s.student_id, s.student_name
FROM Students s
LEFT JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.enrollment_id IS NULL;
SELECT AVG(s.student_age) AS average_age
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE c.course_name = 'Science';

```

Results 1 x

SELECT AVG(s.student_age) AS average_age

Grid	average_age
1	16.5

Q10.

students-data * <postgres> query x

```

SELECT s.student_id, s.student_name, g.grade_name
FROM Students s
JOIN Grades g ON s.student_grade_id = g.grade_id
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE c.course_name = 'History';

```

students(+) 1 x

SELECT s.student_id, s.student_name, g.grade_name

Grid	student_id	student_name	grade_name
1	3	Charlie	A
2	7	Grace	B