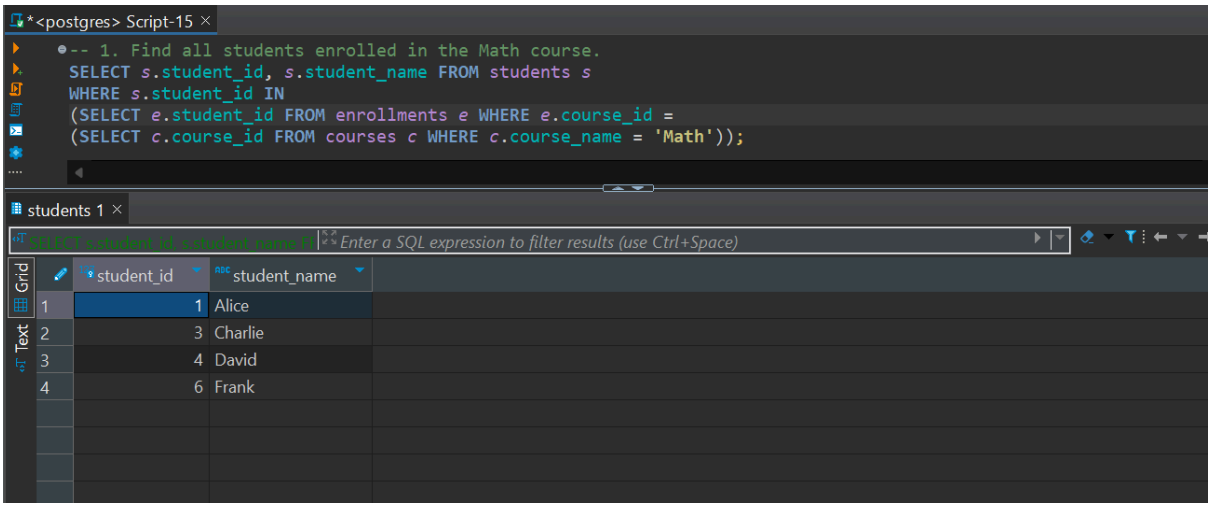
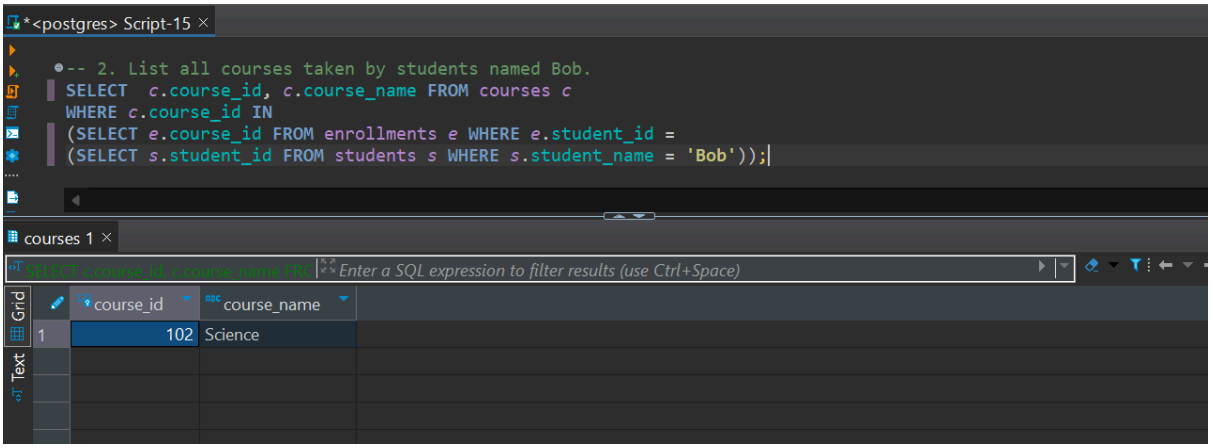
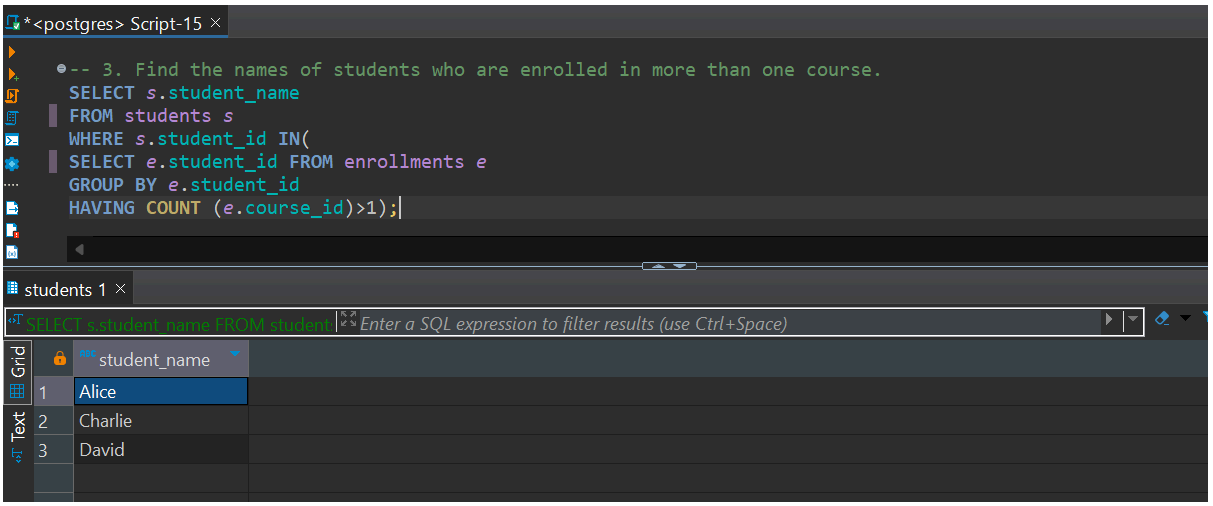
Q1.


*<postgres> Script-15
-- 1. Find all students enrolled in the Math course.
SELECT s.student_id, s.student_name FROM students s
WHERE s.student_id IN
(SELECT e.student_id FROM enrollments e WHERE e.course_id =
(SELECT c.course_id FROM courses c WHERE c.course_name = 'Math'));
students 1
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
student_id
student_name
1 1 Alice
2 3 Charlie
3 4 David
4 6 Frank

Q2.


*<postgres> Script-15
-- 2. List all courses taken by students named Bob.
SELECT c.course_id, c.course_name FROM courses c
WHERE c.course_id IN
(SELECT e.course_id FROM enrollments e WHERE e.student_id =
(SELECT s.student_id FROM students s WHERE s.student_name = 'Bob'));
courses 1
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
course_id
course_name
1 102 Science

Q3.


*<postgres> Script-15
-- 3. Find the names of students who are enrolled in more than one course.
SELECT s.student_name
FROM students s
WHERE s.student_id IN(
SELECT e.student_id FROM enrollments e
GROUP BY e.student_id
HAVING COUNT (e.course_id)>1);
students 1
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
student_name
1 Alice
2 Charlie
3 David

Q4.

```sql
-- 4. List all students who are in Grade A (grade_id = 1).
SELECT s.student_name FROM students s WHERE s.student_grade_id IN
(SELECT g.grade_id from grades g WHERE grade_name='A');
```

students 1

Enter a SQL expression to filter results (use Ctrl+Space)

	student_name
1	Alice
2	Charlie
3	Eve
4	Henry

Q5.

```sql
-- 5. Find the number of students enrolled in each course.
SELECT e.course_id,COUNT(e.student_id) FROM enrollments e
GROUP BY e.course_id
ORDER BY e.course_id ASC;
```

enrollments 1

Enter a SQL expression to filter results (use Ctrl+Space)

	course_id	count
1	101	4
2	102	4
3	103	2

Q6.

*<postgres> Script-15

```sql
-- 6. Retrieve the course with the highest number of enrollments.
SELECT e.course_id FROM enrollments e
GROUP BY e.course_id
HAVING COUNT(e.student_id) = (
    SELECT MAX(student_count)
    FROM (
        SELECT COUNT(student_id) AS student_count
        FROM enrollments
        GROUP BY course_id
    ) AS max_counts
);
```

enrollments 1

Enter a SQL expression to filter results (use Ctrl+Space)

	course_id
1	101
2	102

Q7.

```sql
-- 7. List students who are enrolled in all available courses.
SELECT e.student_id FROM enrollments e
GROUP BY (e.student_id)
HAVING COUNT(e.course_id)=
(SELECT COUNT(c.course_id) FROM courses c)
```

enrollments 1

Enter a SQL expression to filter results (use Ctrl+Space)

student_id

Q8.

```sql
-- 8. Find students who are not enrolled in any courses.
SELECT s.student_id, s.student_name FROM students s
WHERE s.student_id NOT IN
(SELECT e.student_id FROM enrollments e);
```

students 1

Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_name
1	8	Henry
2	9	Ivy
3	10	Jack

Q9.

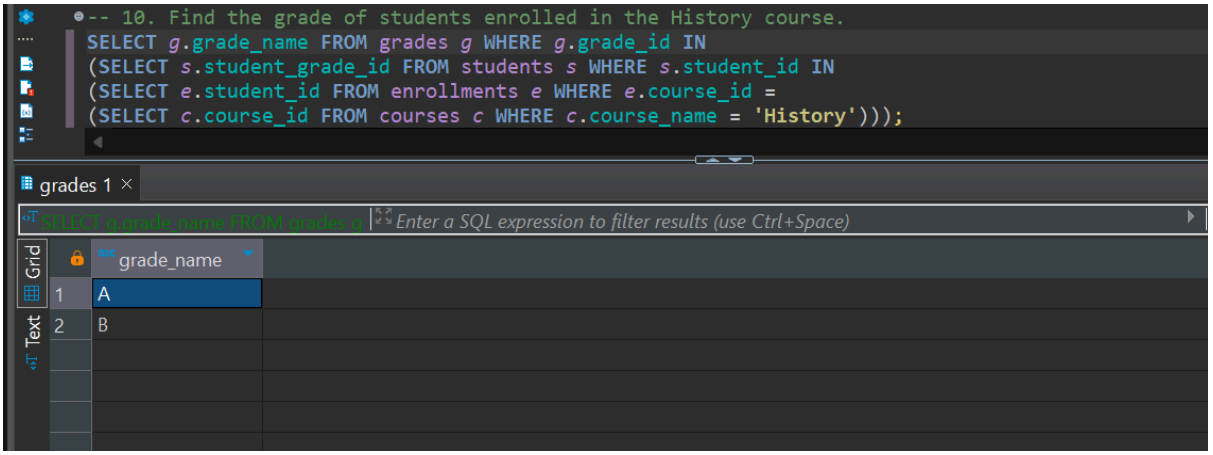
```sql
-- 9. Retrieve the average age of students enrolled in the Science course.
SELECT AVG(s.student_age) as average_age FROM students s WHERE s.student_id IN
(SELECT e.student_id FROM enrollments e WHERE e.course_id =
(SELECT c.course_id FROM courses c WHERE c.course_name = 'Science'));
```

Results 1

Enter a SQL expression to filter results (use Ctrl+Space)

	average_age
1	16.5

Q10.


-- 10. Find the grade of students enrolled in the History course.
SELECT g.grade_name FROM grades g WHERE g.grade_id IN
(SELECT s.student_grade_id FROM students s WHERE s.student_id IN
(SELECT e.student_id FROM enrollments e WHERE e.course_id =
(SELECT c.course_id FROM courses c WHERE c.course_name = 'History')));
grades 1
SELECT g.grade_name FROM grades g
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
grade_name
1
A
2
B