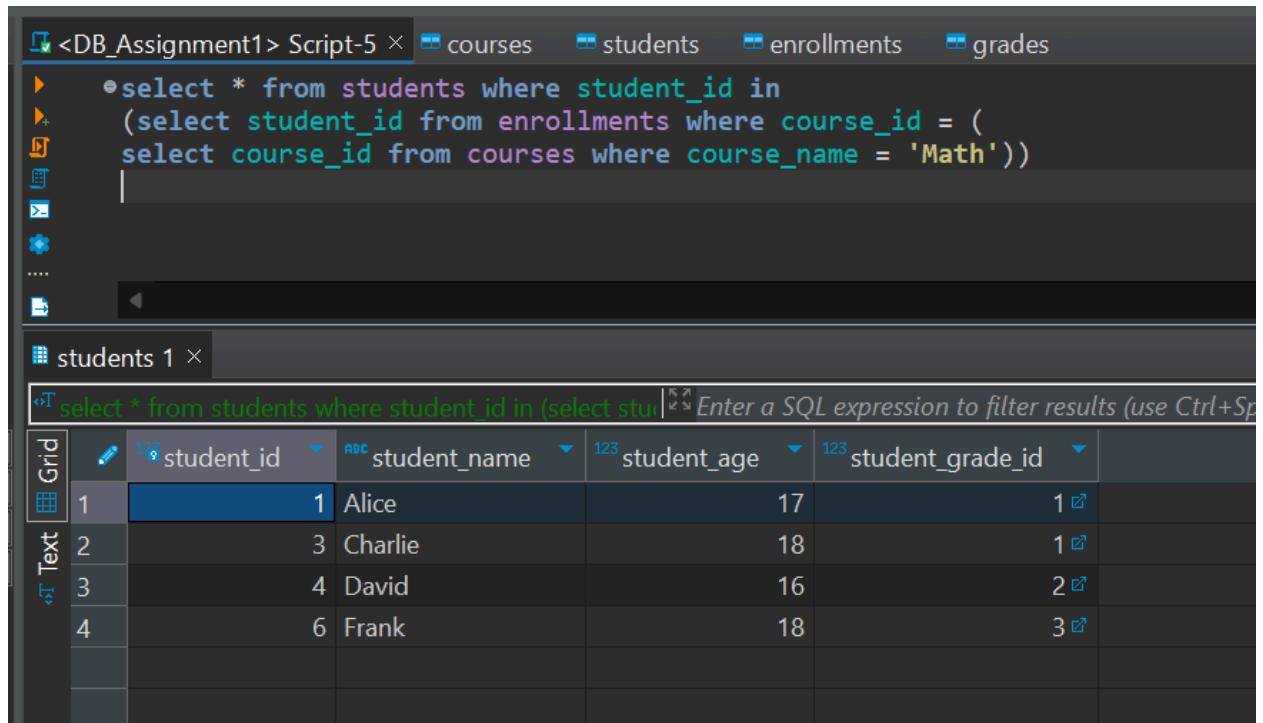


Q1. Find all students enrolled in the Math course.

```
select * from students where student_id in
(select student_id from enrollments where course_id =
(select course_id from courses where course_name = 'Math'))
```

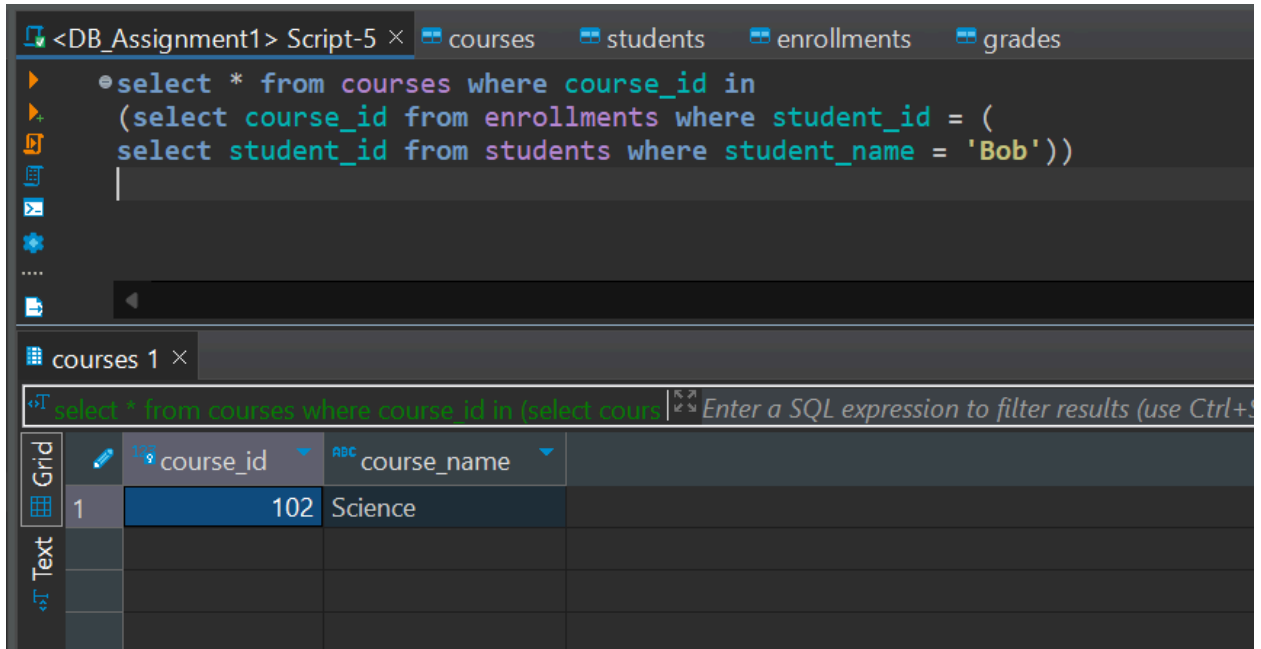


The screenshot shows a database management tool interface. At the top, there are tabs for '<DB\_Assignment1> Script-5', 'courses', 'students', 'enrollments', and 'grades'. The 'Script-5' tab is active, displaying the SQL query:   
`select * from students where student_id in  
(select student_id from enrollments where course_id = (  
select course_id from courses where course_name = 'Math'))`  
Below the query editor, there is a tab labeled 'students 1'. Below this tab, a text input field contains the same SQL query. Below the text input, a table displays the results of the query. The table has five columns: 'student\_id', 'student\_name', 'student\_age', 'student\_grade\_id', and an empty column. The results show four rows of data: (1, Alice, 17, 1), (3, Charlie, 18, 1), (4, David, 16, 2), and (6, Frank, 18, 3). The first row is highlighted in blue.

	student_id	student_name	student_age	student_grade_id	
1	1	Alice	17	1	
2	3	Charlie	18	1	
3	4	David	16	2	
4	6	Frank	18	3	

Q2. List all courses taken by students named Bob.

```
select * from courses where course_id in
(select course_id from enrollments where student_id = (
select student_id from students where student_name = 'Bob'));
```



The screenshot shows a database IDE with a script editor and a results pane. The script editor contains the following SQL query:

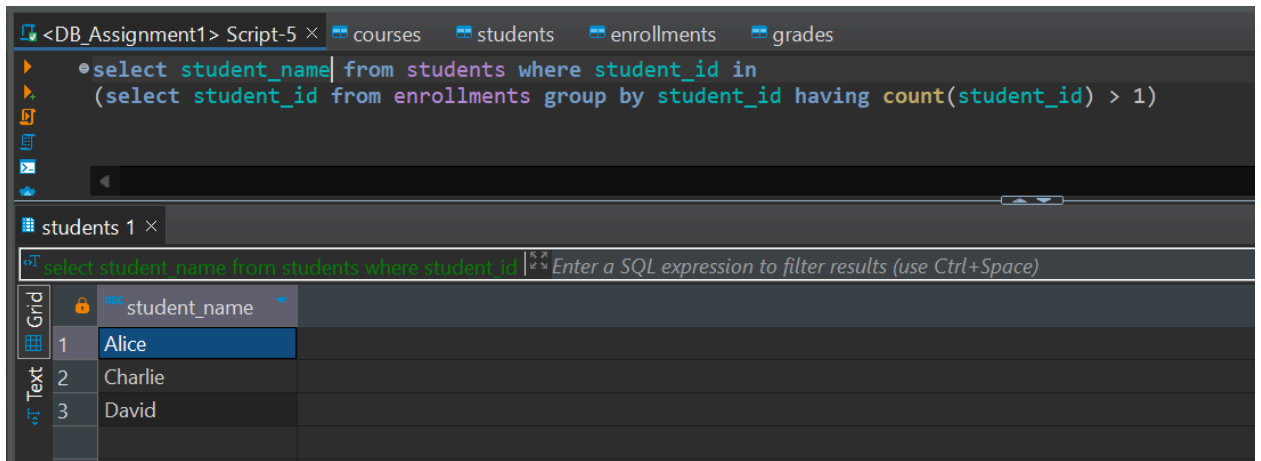
```
select * from courses where course_id in
(select course_id from enrollments where student_id = (
select student_id from students where student_name = 'Bob'))
```

The results pane shows a table with the following data:

course_id	course_name
102	Science

Q3. Find the names of students who are enrolled in more than one course.

```
select student_name from students where student_id in
(select student_id from enrollments group by student_id having count(student_id) > 1);
```



The screenshot shows a database IDE with a script editor and a results pane. The script editor contains the following SQL query:

```
select student_name from students where student_id in
(select student_id from enrollments group by student_id having count(student_id) > 1)
```

The results pane shows a table with the following data:

student_name
Alice
Charlie
David

Q4. List all students who are in Grade A (grade\_id = 1).

```
SELECT * from students where  
student_grade_id = (select grade_id from grades where grade_name = 'A');
```

The screenshot shows a database IDE with a script editor and a results pane. The script editor contains the following SQL query:

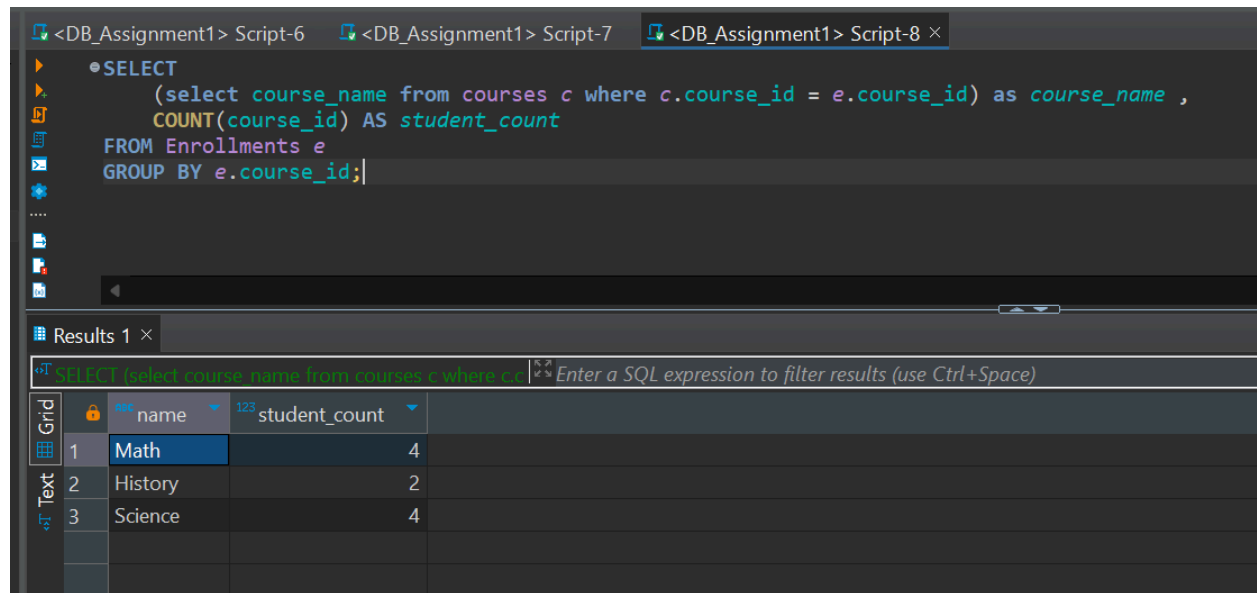
```
SELECT * from students where  
student_grade_id = (select grade_id from grades where grade_name = 'A');
```

The results pane, titled "students 1", displays the results of the query. It shows a table with 5 columns: student\_id, student\_name, student\_age, student\_grade\_id, and an empty column. The results are as follows:

	student_id	student_name	student_age	student_grade_id	
1	1	Alice	17	1	
2	3	Charlie	18	1	
3	5	Eve	17	1	
4	8	Henry	16	1	

Q5. Find the number of students enrolled in each course.

```
SELECT
    (select course_name from courses c where c.course_id = e.course_id) as course_name ,
    COUNT(course_id) AS student_count
FROM Enrollments e
GROUP BY e.course_id;
```



The screenshot shows a database IDE with three tabs: <DB\_Assignment1> Script-6, <DB\_Assignment1> Script-7, and <DB\_Assignment1> Script-8. The active tab, Script-8, contains the following SQL query:

```
SELECT
    (select course_name from courses c where c.course_id = e.course_id) as course_name ,
    COUNT(course_id) AS student_count
FROM Enrollments e
GROUP BY e.course_id;
```

Below the query editor, the 'Results' pane shows the output of the query. It displays a table with two columns: 'name' and 'student\_count'. The table contains three rows of data:

	name	student_count
1	Math	4
2	History	2
3	Science	4

Q6. Retrieve the course with the highest number of enrollments.

```
select course_name from courses where course_id = (select max(course_id) from enrollments);
```

The screenshot shows a database IDE interface. At the top, there are tabs for database objects: `<DB_Assignment1> Script-5`, `courses`, `students`, `enrollments`, and `grades`. The main editor displays the following SQL query:

```
select course_name from courses where course_id =  
(select max(course_id) from enrollments) ;
```

Below the editor, a results pane titled `courses 1` is visible. It shows a table with the following data:

	course_name
1	History

On the left side of the results pane, there are icons for `Grid` and `Text`. The `Grid` icon is selected. A tooltip is visible over the `course_name` header, displaying the text: `Enter a SQL expression to filter results (use Ctrl+Space)`.

Q7. List students who are enrolled in all available courses.

```
select student_name
from Students
WHERE
    student_id IN (
        select student_id
        from Enrollments
        GROUP BY
            student_id
        HAVING
            COUNT(course_id) = (SELECT
                COUNT(course_id)
            FROM
                Courses));
```

The screenshot shows a database IDE with three tabs: "<DB\_Assignment1> Script-6", "<DB\_Assignment1> Script-7", and "<DB\_Assignment1> Script-8". The active tab is "Script-8", which contains the SQL query from the previous block. Below the editor, there is a results pane titled "students 1 ×". It shows a table with the column "student\_name" and a filter bar with the text "SELECT student\_name FROM Students WHERE student\_id". The table is currently empty, and the filter bar has a placeholder text "Enter a SQL expression to filter results (use Ctrl+Space)".

student_name
--------------

Q8. Find students who are not enrolled in any courses.

```
SELECT
    student_name
FROM
    Students
WHERE
    student_id NOT IN (
        SELECT
            student_id
        FROM
            Enrollments
        GROUP BY
            student_id
        HAVING
            COUNT(course_id) > 0
    );
```

The screenshot shows a database IDE with three tabs: "<DB\_Assignment1> Script-6", "<DB\_Assignment1> Script-7", and "<DB\_Assignment1> Script-8". The active tab, Script-8, contains the SQL query from the previous block. Below the editor, a results pane titled "students 1" shows the output of the query. The results are displayed in a table with two columns: "student\_name" and an empty column. The first three rows contain the names "Henry", "Ivy", and "Jack".

	student_name	
1	Henry	
2	Ivy	
3	Jack	

Q9. Retrieve the average age of students enrolled in the Science course.

```
SELECT
    AVG(student_age) as Avg_Age
FROM
    Students
WHERE
    student_id IN (
        SELECT
            student_id
        FROM
            Enrollments
        WHERE
            course_id = (
                SELECT
                    course_id
                FROM
                    courses
                WHERE
                    course_name = 'Science'
            )
    );
```



<DB\_Assignment1> Script-6<DB\_Assignment1> Script-7<DB\_Assignment1> Script-8 ×

•SELECT  
AVG(student\_age) as Avg\_Age  
FROM  
Students  
WHERE  
student\_id IN (  
SELECT  
student\_id  
FROM  
Enrollments  
WHERE  
course\_id = (  
SELECT  
course\_id  
FROM  
courses  
WHERE  
course\_name = 'Science'  
)  
);

⚙️  
⋮  
📄  
🔍  
🔧  
🔗  
🔑  
🔒  
🔥  
🔧  
🔗  
🔑  
🔒  
🔥

Results 1 ×

SELECT AVG(student\_age) as Avg\_Age FROM Student

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

Grid

1

avg\_age

16.5

Text

Q10. Find the grade of students enrolled in the History course.

```
SELECT
  student_name,
  (
    Select
      grade_name
    from
      grades G
    where
      S.student_grade_id = G.grade_id
  ) as Grade
from
  Students S
where
  student_id in (
    select
      student_id
    from
      Enrollments E
    where
      E.course_id = (
        Select
          course_id
        from
          courses
        where
          course_name = 'History'
      )
  )
)
```

<DB\_Assignment1> Script-6

<DB\_Assignment1> Script-7

<DB\_Assignment1> Script-8 ×

• SELECT

student\_name,

(

Select

grade\_name

from

grades G

where

S.student\_grade\_id = G.grade\_id

) as Grade

from

Students S

where

student\_id in (

select

student\_id

from

Enrollments E

where

E.course\_id = (

Select

course\_id

from

courses

where

course\_name = 'History'

)

)

students 1 ×

SELECT student\_name, ( Select grade name from gra

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

	student_name	grade	
1	Charlie	A	
2	Grace	B	

ERD of book store.

