ASSIGNMENT 2:

Create queries:

```sql
-- Students table
CREATE TABLE Students (
student_id INT PRIMARY KEY,
student_name VARCHAR(50),
student_age INT,
student_grade_id INT,
FOREIGN KEY (student_grade_id) REFERENCES Grades(grade_id)
);
-- Grades table
CREATE TABLE Grades (
grade_id INT PRIMARY KEY,
grade_name VARCHAR(10)
);
-- Courses table
CREATE TABLE Courses (
course_id INT PRIMARY KEY,
course_name VARCHAR(50)
);
-- Enrollments table
CREATE TABLE Enrollments (
enrollment_id INT PRIMARY KEY,
student_id INT,
course_id INT,
enrollment_date DATE,
FOREIGN KEY (student_id) REFERENCES Students(student_id),
FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

Insert queries:

```sql
-- Insert into Grades table
INSERT INTO Grades (grade_id, grade_name) VALUES
(1, 'A'),
(2, 'B'),
(3, 'C');
-- Insert into Courses table
INSERT INTO Courses (course_id, course_name) VALUES
(101, 'Math'),
(102, 'Science'),
(103, 'History');
-- Insert into Students table
INSERT INTO Students (student_id, student_name, student_age, student_grade_id) VALUES
(1, 'Alice', 17, 1),
(2, 'Bob', 16, 2),
(3, 'Charlie', 18, 1),
(4, 'David', 16, 2),
(5, 'Eve', 17, 1),
(6, 'Frank', 18, 3),
(7, 'Grace', 17, 2),
(8, 'Henry', 16, 1),
(9, 'Ivy', 18, 2),
(10, 'Jack', 17, 3);
-- Insert into Enrollments table
INSERT INTO Enrollments (enrollment_id, student_id, course_id, enrollment_date)
VALUES
(1, 1, 101, '2023-09-01'),
(2, 1, 102, '2023-09-01'),
(3, 2, 102, '2023-09-01'),
(4, 3, 101, '2023-09-01'),
(5, 3, 103, '2023-09-01'),
(6, 4, 101, '2023-09-01'),
(7, 4, 102, '2023-09-01'),
(8, 5, 102, '2023-09-01'),
(9, 6, 101, '2023-09-01'),
(10, 7, 103, '2023-09-01');
```

ER Diagram:

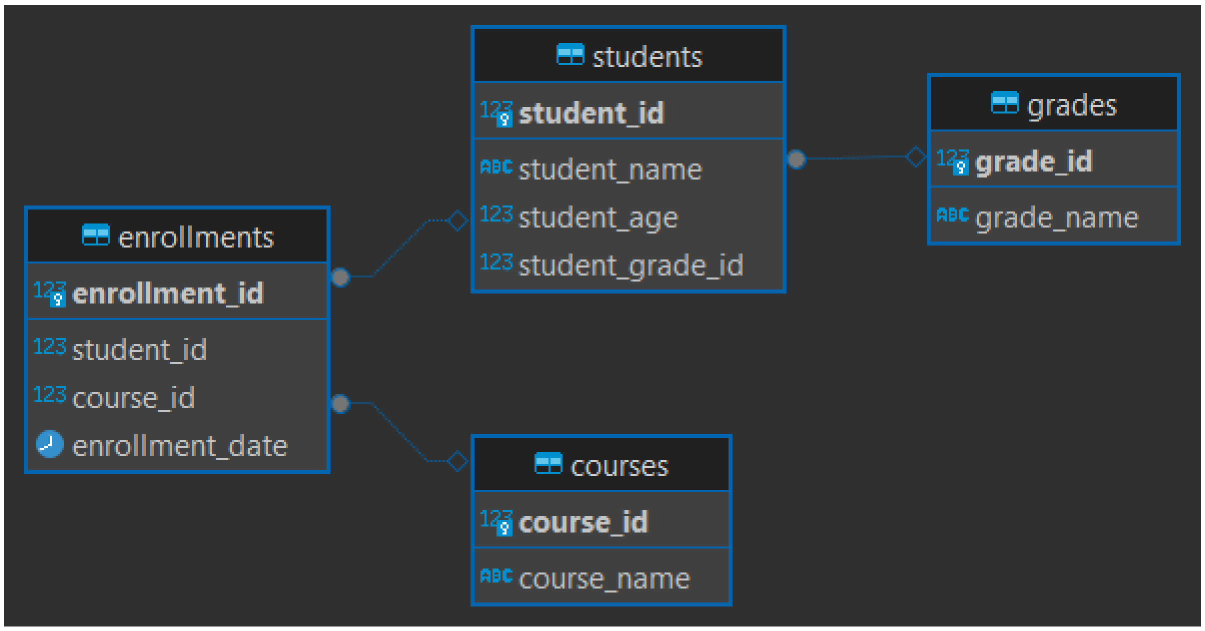
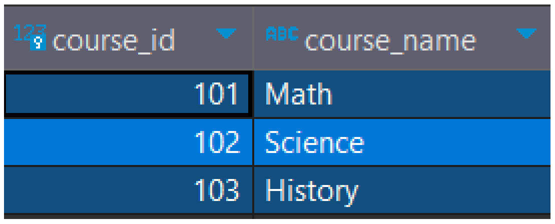


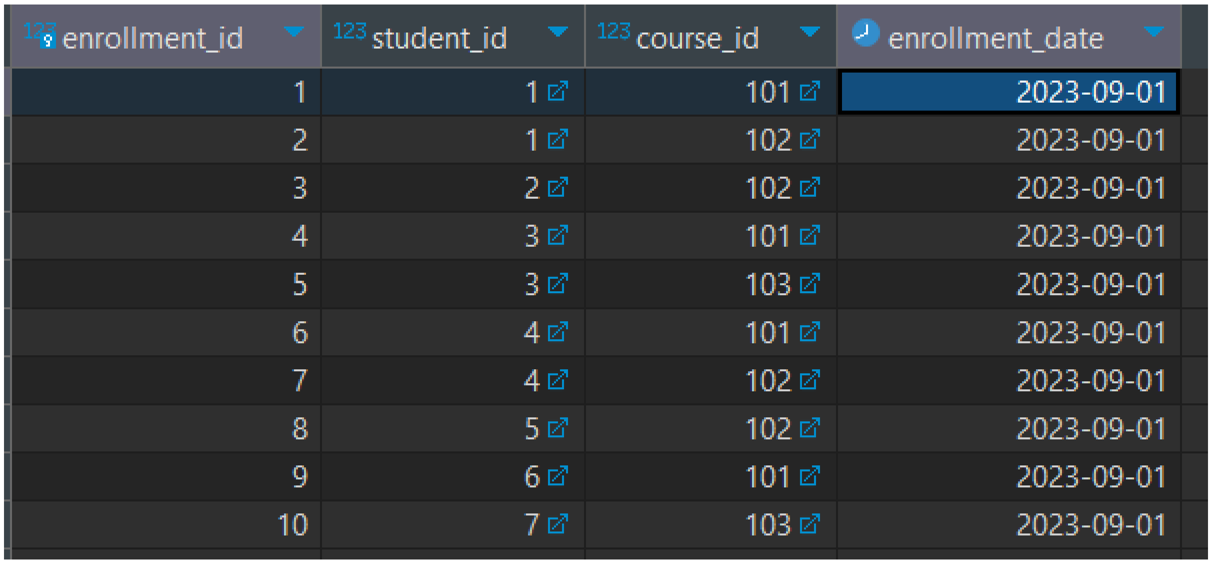
Table Data:

Courses



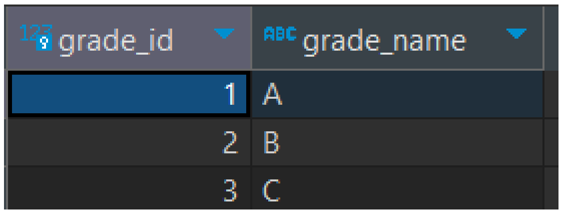
course_id	course_name
101	Math
102	Science
103	History

Enrollments



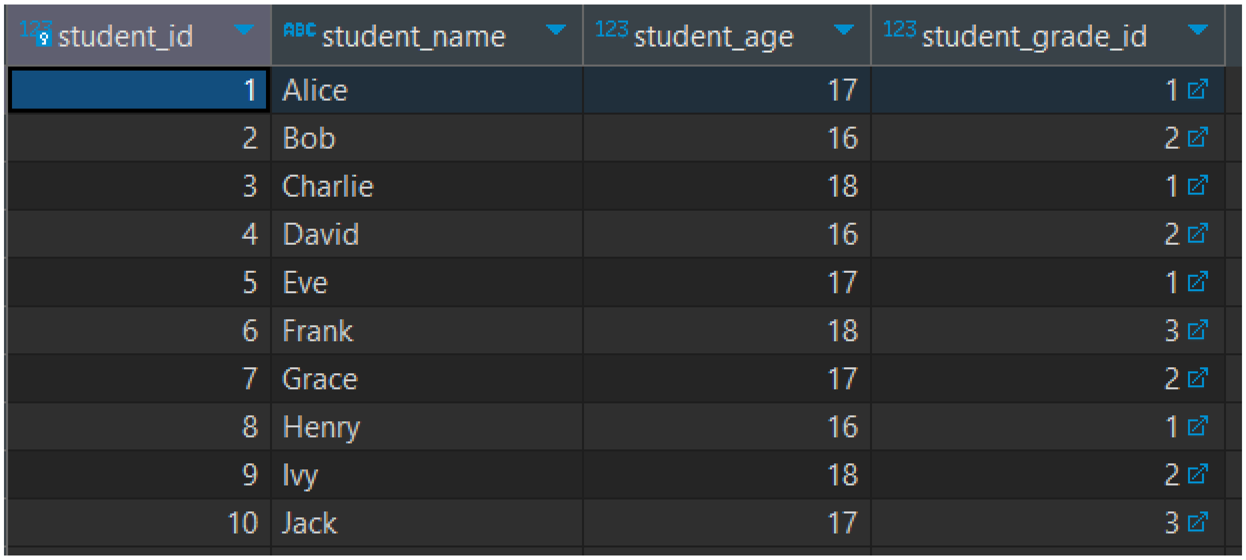
enrollment_id	student_id	course_id	enrollment_date
1	1	101	2023-09-01
2	1	102	2023-09-01
3	2	102	2023-09-01
4	3	101	2023-09-01
5	3	103	2023-09-01
6	4	101	2023-09-01
7	4	102	2023-09-01
8	5	102	2023-09-01
9	6	101	2023-09-01
10	7	103	2023-09-01

Grades



grade_id	grade_name
1	A
2	B
3	C

Students



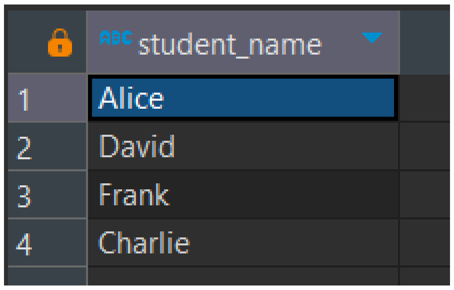
student_id	student_name	student_age	student_grade_id
1	Alice	17	1
2	Bob	16	2
3	Charlie	18	1
4	David	16	2
5	Eve	17	1
6	Frank	18	3
7	Grace	17	2
8	Henry	16	1
9	Ivy	18	2
10	Jack	17	3

Questions:

1. Find all students enrolled in the Math course.

```
select s.student_name
from students s
where s.student_id in (
 select e.student_id
 from enrollments e
 where e.course_id in (
        select c.course_id
        from courses c
        where c.course_name = 'Math'));
```

Output:

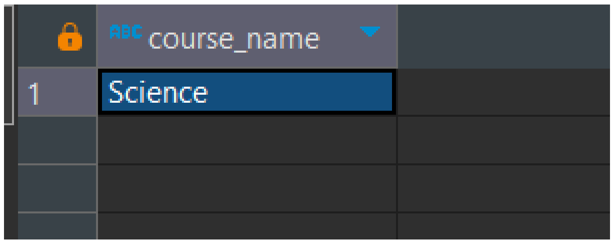


	student_name
1	Alice
2	David
3	Frank
4	Charlie

2. List all courses taken by students named Bob.

```
select c.course_name
from courses c
where c.course_id = (
 select e.course_id
 from enrollments e
 where e.student_id = (
              select s.student_id
              from students s
              where s.student_name ='Bob' ));
```

Output:

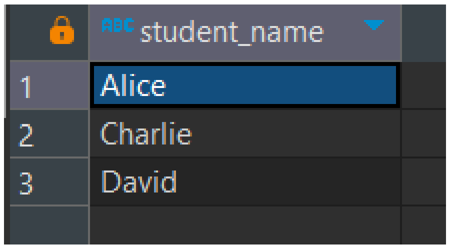


```
  course_name
1 Science
```

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 student_id
 having count(student_id)>1);
```

Output:

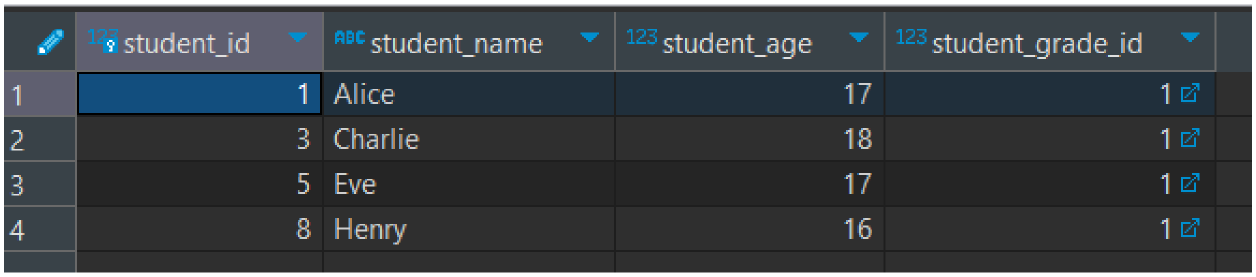


```
  student_name
1 Alice
2 Charlie
3 David
```

4. List all students who are in Grade A (grade_id = 1)

```
select *
from students s
where s.student_grade_id =1;
```

Output:

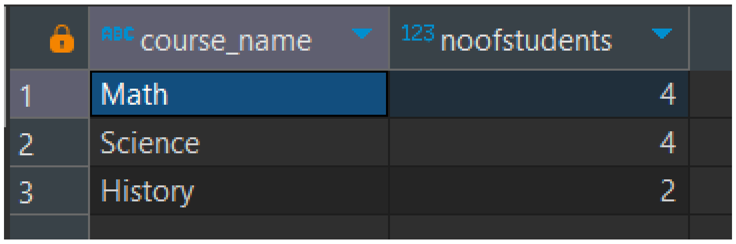


	student_id	student_name	student_age	student_grade_id
1	1	Alice	17	1
2	3	Charlie	18	1
3	5	Eve	17	1
4	8	Henry	16	1

5. Find the number of students enrolled in each course

```
SELECT c.course_name,
       (SELECT COUNT(e.student_id)
        FROM enrollments e
        WHERE e.course_id = c.course_id) AS noOfStudents
FROM courses c;
```

Output:

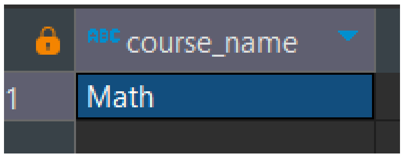


	course_name	noofstudents
1	Math	4
2	Science	4
3	History	2

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

```
SELECT c.course_name
FROM courses c
WHERE c.course_id = (
    SELECT e.course_id
    FROM enrollments e
    GROUP BY e.course_id
    ORDER BY COUNT(e.student_id) DESC
    LIMIT 1
);
```

Output:



course_name
Math

7. List students who are enrolled in all available courses

```
SELECT s.student_name
FROM students s
WHERE (SELECT COUNT(DISTINCT e.course_id)
       FROM enrollments e
       WHERE e.student_id = s.student_id) = (SELECT COUNT(course_id) FROM courses);
```

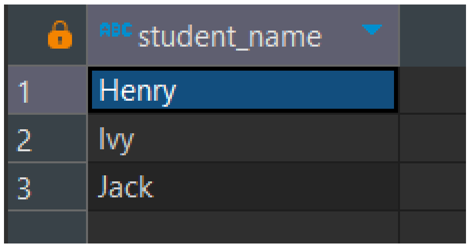
Output:

Empty

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

```
SELECT s.student_name
FROM students s
WHERE s.student_id NOT IN (
    SELECT e.student_id
    FROM enrollments e
);
```

Output:

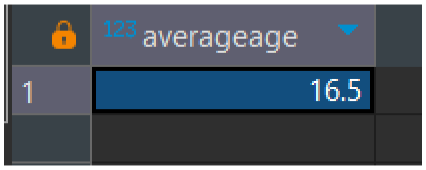


student_name
Henry
Ivy
Jack

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

```
SELECT AVG(s.student_age) as AverageAge
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')
);
```

Output:

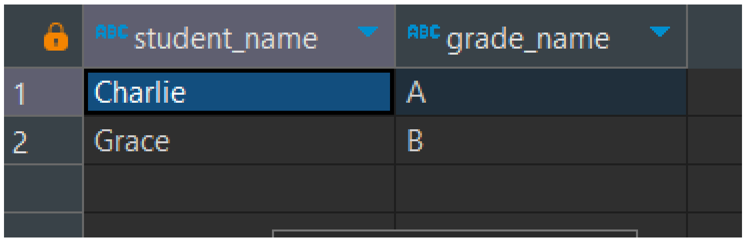


	averageage
1	16.5

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

```sql
SELECT
    s.student_name,
    (SELECT g.grade_name
     FROM grades g
     WHERE g.grade_id = s.student_grade_id) AS grade_name
FROM students s
WHERE s.student_id IN (
    SELECT e.student_id
    FROM enrollments e
    WHERE e.course_id = (SELECT course_id FROM courses WHERE course_name = 'History')
);
```

Output:



	student_name	grade_name
1	Charlie	A
2	Grace	B