Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID.

```
mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |   1000 | 2024-01-01   |
|          2 |          2 |   1500 | 2024-01-02   |
|          3 |          3 |   1200 | 2024-01-03   |
|          4 |          4 |   1300 | 2024-01-04   |
|          5 |          5 |   2010 | 2024-01-05   |
|          6 |          6 |   1100 | 2024-01-06   |
|          7 |          7 |   1600 | 2024-01-07   |
|          8 |          8 |   1700 | 2024-01-08   |
|          9 |          9 |   1800 | 2024-01-09   |
|         10 |         10 |   1900 | 2024-01-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT S.student_id,SUM(P.amount) FROM
    -> students S JOIN payments P ON
    -> S.student_id=P.student_id
    -> WHERE s.student_id=5;
+------------+---------------+
| student_id | SUM(P.amount) |
+------------+---------------+
|          5 |          2010 |
+------------+---------------+
1 row in set (0.00 sec)
```

2. Write an SQL query to retrieve a list of courses along with the count of students enrolled in each course. Use a JOIN operation between the "Courses" table and the "Enrollments" table.

```
mysql> SELECT C.course_id,COUNT(E.student_id) AS 'Total Enrolled Students' FROM
    -> courses C JOIN enrollments E ON
    -> C.course_id=E.course_id
    -> GROUP BY C.course_id;
+-----------+-------------------------+
| course_id | Total Enrolled Students |
+-----------+-------------------------+
|       101 |                       1 |
|       102 |                       1 |
|       103 |                       1 |
|       104 |                       2 |
|       105 |                       1 |
|       106 |                       1 |
|       107 |                       1 |
|       108 |                       1 |
|       109 |                       1 |
|       110 |                       1 |
+-----------+-------------------------+
10 rows in set (0.00 sec)
```

3. Write an SQL query to find the names of students who have not enrolled in any course. Use a LEFT JOIN between the "Students" table and the "Enrollments" table to identify students without enrollments.

```
mysql> SELECT S.student_id,CONCAT(S.first_name,' ',S.last_name) AS Name,COUNT(E.enrollment_id) AS 'Total Courses Enrolled' FROM
    -> Students S LEFT JOIN enrollments E ON
    -> S.student_id=E.student_id
    -> GROUP BY S.student_id;
+------------+---------------+-----------------------+
| student_id | Name          | Total Courses Enrolled |
+------------+---------------+-----------------------+
|          1 | Alice Smith   |                     1 |
|          2 | Bob Johnson   |                     1 |
|          3 | Carol Williams |                     2 |
|          4 | David Brown   |                     1 |
|          5 | Emma Davis    |                     1 |
|          6 | Frank Miller  |                     1 |
|          7 | Grace Wilson  |                     1 |
|          8 | Henry Moore   |                     1 |
|          9 | Isabel Taylor |                     1 |
|         10 | Jack Anderson |                     1 |
|         11 | John Doe      |                     0 |
+------------+---------------+-----------------------+
11 rows in set (0.00 sec)
```

4. Write an SQL query to retrieve the first name, last name of students, and the names of the courses they are enrolled in. Use JOIN operations between the "Students" table and the "Enrollments" and "Courses" tables.

```
mysql> SELECT
    -> s.first_name, s.last_name, 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;
+------------+-----------+--------------------+
| first_name | last_name | course_name        |
+------------+-----------+--------------------+
| Alice      | Smith     | Mathematics        |
| Bob        | Johnson   | Physics            |
| Carol      | Williams  | Chemistry          |
| Carol      | Williams  | Biology            |
| David      | Brown     | Biology            |
| Emma       | Davis     | English Literature |
| Frank      | Miller    | Computer Science   |
| Grace      | Wilson    | Art History        |
| Henry      | Moore     | Economics          |
| Isabel     | Taylor    | Psychology         |
| Jack       | Anderson  | Sociology          |
+------------+-----------+--------------------+
11 rows in set (0.00 sec)
```

5. Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.

```
mysql> SELECT T.teacher_id,T.first_name,T.last_name,GROUP_CONCAT(C.course_id) AS 'Courses teached' FROM
    -> teacher T JOIN courses C ON
    -> T.teacher_id=C.teacher_id
    -> GROUP BY T.teacher_id;
+------------+------------+-----------+-----------------+
| teacher_id | first_name | last_name | Courses teached |
+------------+------------+-----------+-----------------+
|          1 | John       | Doe       | 101,111         |
|          2 | Jane       | Doe       | 102             |
|          3 | Jim        | Beam      | 103             |
|          4 | Sara       | Conor     | 104             |
|          5 | Luke       | Skywalker | 105             |
|          6 | Leia       | Organa    | 106             |
|          7 | Han        | Solo      | 107             |
|          8 | Anakin     | Skywalker | 108             |
|          9 | Obi-Wan    | Kenobi    | 109             |
|         10 | Yoda       | Master    | 110             |
+------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

6. Retrieve a list of students and their enrollment dates for a specific course. You'll need to join the "Students" table with the "Enrollments" and "Courses" tables.

```
mysql> SELECT
    -> s.first_name, s.last_name, e.enrollment_date
    -> 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 = 'Mathematics';
+------------+-----------+-----------------+
| first_name | last_name | enrollment_date |
+------------+-----------+-----------------+
| Alice      | Smith     | 2024-01-15      |
+------------+-----------+-----------------+
1 row in set (0.00 sec)
```

7. Find the names of students who have not made any payments. Use a LEFT JOIN between the "Students" table and the "Payments" table and filter for students with NULL payment records.

```
mysql> SELECT S.*,P.Amount FROM
    -> students S LEFT JOIN payments P
    -> ON S.student_id=P.student_id;
+------------+------------+-----------+---------------+-------------------------+--------------+--------+
| student_id | first_name | last_name | date_of_birth | email                   | phone_number | Amount |
+------------+------------+-----------+---------------+-------------------------+--------------+--------+
|          1 | Alice      | Smith     | 2001-01-01    | alice.smith@email.com   |   1234567890 |   1000 |
|          2 | Bob        | Johnson   | 2001-02-02    | bob.johnson@email.com   |   2345678901 |   1500 |
|          3 | Carol      | Williams  | 2001-03-03    | carol.williams@email.com|   3456789012 |   1200 |
|          4 | David      | Brown     | 2001-04-04    | david.brown@email.com   |   4567890123 |   1300 |
|          5 | Emma       | Davis     | 2001-05-05    | emma.davis@email.com    |   5678901234 |   2010 |
|          6 | Frank      | Miller    | 2001-06-06    | frank.miller@email.com  |   6789012345 |   1100 |
|          7 | Grace      | Wilson    | 2001-07-07    | grace.wilson@email.com  |   7890123456 |   1600 |
|          8 | Henry      | Moore     | 2001-08-08    | henry.moore@email.com   |   8901234567 |   1700 |
|          9 | Isabel     | Taylor    | 2001-09-09    | isabel.taylor@email.com |   9014345678 |   1800 |
|         10 | Jack       | Anderson  | 2001-10-10    | jack.anderson@email.com |    123456789 |   1900 |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com    |   1234567890 |   NULL |
+------------+------------+-----------+---------------+-------------------------+--------------+--------+
11 rows in set (0.00 sec)
```

8. Write a query to identify courses that have no enrollments. You'll need to use a LEFT JOIN between the "Courses" table and the "Enrollments" table and filter for courses with NULL enrollment records.

```
mysql> SELECT C.course_id,C.course_name,E.enrollment_id FROM
    -> courses C LEFT JOIN enrollments E ON
    -> C.course_id=E.course_id WHERE
    -> E.enrollment_id IS NULL;
+-----------+-------------+---------------+
| course_id | course_name | enrollment_id |
+-----------+-------------+---------------+
|       111 | Probability |          NULL |
+-----------+-------------+---------------+
1 row in set (0.00 sec)
```

9. Identify students who are enrolled in more than one course. Use a self-join on the "Enrollments" table to find students with multiple enrollment records.

```
mysql> SELECT
    -> e1.student_id, COUNT(DISTINCT e1.course_id) AS NumCoursesEnrolled
    -> FROM
    -> Enrollments e1
    -> JOIN
    -> Enrollments e2 ON e1.student_id = e2.student_id
    -> WHERE
    -> e1.course_id <> e2.course_id
    -> GROUP BY
    -> e1.student_id
    -> HAVING
    -> NumCoursesEnrolled > 1;
+------------+--------------------+
| student_id | NumCoursesEnrolled |
+------------+--------------------+
|          3 |                  2 |
+------------+--------------------+
1 row in set (0.00 sec)
```

10. Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.

```
mysql> SELECT T.*,C.course_id FROM
    -> teacher T LEFT JOIN courses C ON
    -> T.teacher_id=C.teacher_id
    -> WHERE C.course_id IS NULL;
+------------+------------+-----------+------------------------+-----------+
| teacher_id | first_name | last_name | email                  | course_id |
+------------+------------+-----------+------------------------+-----------+
|         11 | Manas      | Rustagi   | rustagimanas@yahoo.com |      NULL |
+------------+------------+-----------+------------------------+-----------+
1 row in set (0.00 sec)
```