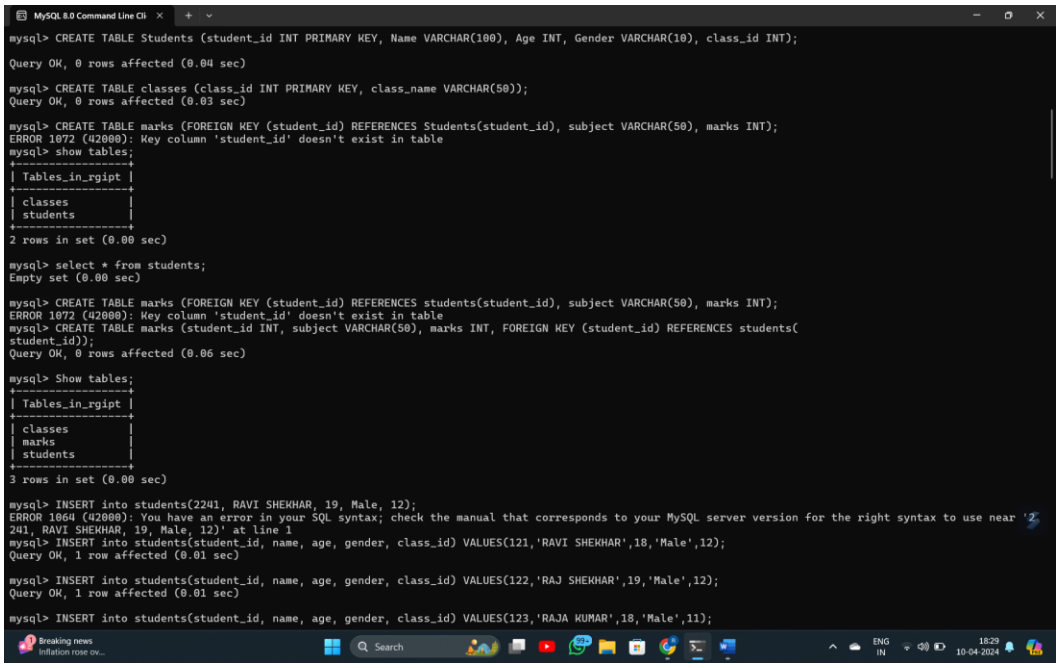
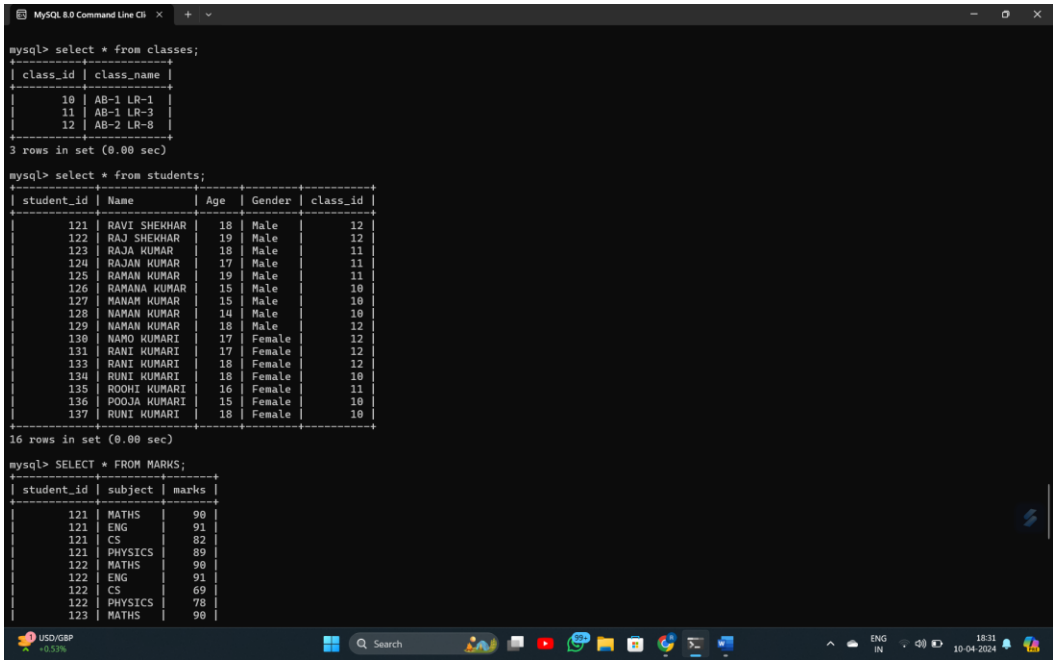
DBMS LAB

Name – RAVI SHEKHAR

Roll No. – 22CS3075



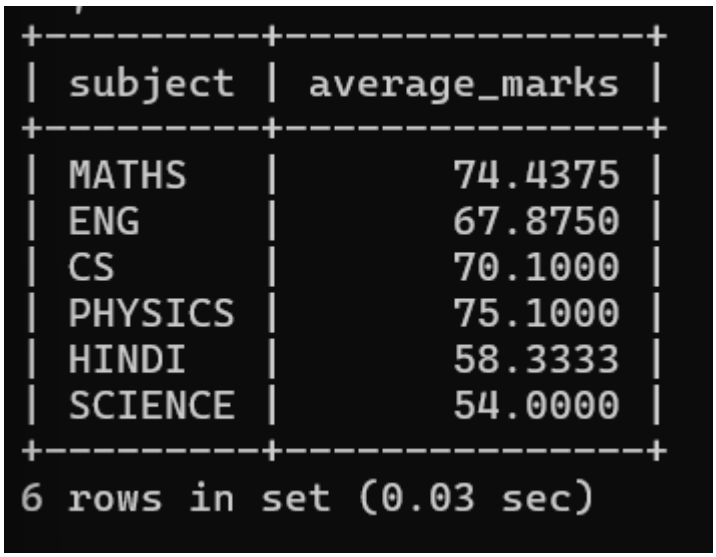
```
MySQL 8.0 Command Line Cli
mysql> CREATE TABLE Students (student_id INT PRIMARY KEY, Name VARCHAR(100), Age INT, Gender VARCHAR(10), class_id INT);

Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE classes (class_id INT PRIMARY KEY, class_name VARCHAR(50));
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE marks (FOREIGN KEY (student_id) REFERENCES Students(student_id), subject VARCHAR(50), marks INT);
ERROR 1072 (42000): Key column 'student_id' doesn't exist in table
mysql> show tables;
+-----------------+
| Tables_in_rgipt |
+-----------------+
| classes         |
| students        |
+-----------------+
2 rows in set (0.00 sec)

mysql> select * from students;
Empty set (0.00 sec)

mysql> CREATE TABLE marks (FOREIGN KEY (student_id) REFERENCES students(student_id), subject VARCHAR(50), marks INT);
ERROR 1072 (42000): Key column 'student_id' doesn't exist in table
mysql> CREATE TABLE marks (student_id INT, subject VARCHAR(50), marks INT, FOREIGN KEY (student_id) REFERENCES students(
student_id));
Query OK, 0 rows affected (0.06 sec)

mysql> Show tables;
+-----------------+
| Tables_in_rgipt |
+-----------------+
| classes         |
| marks           |
| students        |
+-----------------+
3 rows in set (0.00 sec)

mysql> INSERT into students(2241, RAVI SHEKHAR, 19, Male, 12);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '2
241, RAVI SHEKHAR, 19, Male, 12)' at line 1
mysql> INSERT into students(student_id, name, age, gender, class_id) VALUES(121,'RAVI SHEKHAR',18,'Male',12);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT into students(student_id, name, age, gender, class_id) VALUES(122,'RAJ SHEKHAR',19,'Male',12);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT into students(student_id, name, age, gender, class_id) VALUES(123,'RAJA KUMAR',18,'Male',11);
```



```
MySQL 8.0 Command Line Cli
mysql> select * from classes;
+----------+------------+
| class_id | class_name |
+----------+------------+
|       10 | AB-1 LR-1  |
|       11 | AB-1 LR-3  |
|       12 | AB-2 LR-8  |
+----------+------------+
3 rows in set (0.00 sec)

mysql> select * from students;
+------------+---------------+------+--------+----------+
| student_id | Name          | Age  | Gender | class_id |
+------------+---------------+------+--------+----------+
|        121 | RAVI SHEKHAR  |   18 | Male   |       12 |
|        122 | RAJ SHEKHAR   |   19 | Male   |       12 |
|        123 | RAJA KUMAR    |   18 | Male   |       11 |
|        124 | RAJAN KUMAR   |   17 | Male   |       11 |
|        125 | RAMAN KUMAR   |   19 | Male   |       11 |
|        126 | RAMANA KUMAR  |   15 | Male   |       10 |
|        127 | MANAM KUMAR   |   15 | Male   |       10 |
|        128 | NAMAN KUMAR   |   14 | Male   |       10 |
|        129 | NAMAN KUMAR   |   18 | Male   |       12 |
|        130 | NAMO KUMARI   |   17 | Female |       12 |
|        131 | RANI KUMARI   |   17 | Female |       12 |
|        133 | RANI KUMARI   |   18 | Female |       12 |
|        134 | RUNI KUMARI   |   18 | Female |       10 |
|        135 | ROOHI KUMARI  |   16 | Female |       11 |
|        136 | POOJA KUMARI  |   15 | Female |       10 |
|        137 | RUNI KUMARI   |   18 | Female |       10 |
+------------+---------------+------+--------+----------+
16 rows in set (0.00 sec)

mysql> SELECT * FROM MARKS;
+------------+---------+-------+
| student_id | subject | marks |
+------------+---------+-------+
|        121 | MATHS   |    90 |
|        121 | ENG     |    91 |
|        121 | CS      |    82 |
|        121 | PHYSICS |    89 |
|        122 | MATHS   |    90 |
|        122 | ENG     |    91 |
|        122 | CS      |    69 |
|        122 | PHYSICS |    78 |
|        123 | MATHS   |    90 |
```

1.

Select subject, AVG(marks) AS average_marks FROM marks GROUP BY subject;

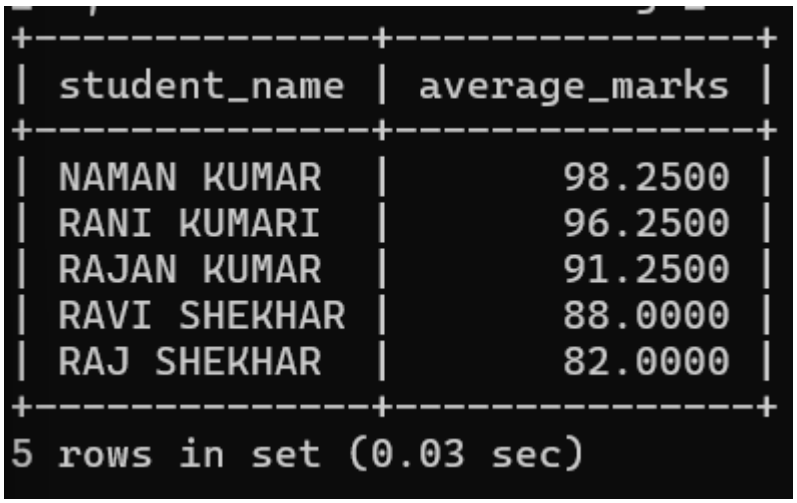


subject	average_marks
MATHS	74.4375
ENG	67.8750
CS	70.1000
PHYSICS	75.1000
HINDI	58.3333
SCIENCE	54.0000

6 rows in set (0.03 sec)

2.

SELECT name, AVG(marks) AS average_marks FROM marks GROUP BY name ORDER BY average_marks DESC LIMIT 5;



student_name	average_marks
NAMAN KUMAR	98.2500
RANI KUMARI	96.2500
RAJAN KUMAR	91.2500
RAVI SHEKHAR	88.0000
RAJ SHEKHAR	82.0000

5 rows in set (0.03 sec)

3.

SELECT c.class_name, COUNT(s.student_id) AS total_students FROM classes c LEFT JOIN students s ON c.class_id = s.class_id GROUP BY c.class_name;

```
+------------+----------------+
| class_name | total_students |
+------------+----------------+
| AB-1 LR-1  |              6 |
| AB-1 LR-3  |              4 |
| AB-2 LR-8  |              6 |
+------------+----------------+
3 rows in set (0.02 sec)
```

4.

SELECT gender, COUNT(*) AS total_students, (COUNT(*) * 100.0) / (SELECT COUNT(*) FROM students) AS percentage FROM students GROUP BY gender;

```
+--------+----------------+------------+
| gender | total_students | percentage |
+--------+----------------+------------+
| Male   |              9 |   56.25000 |
| Female |              7 |   43.75000 |
+--------+----------------+------------+
2 rows in set (0.01 sec)
```

5.

SELECT c.class_name FROM classes c JOIN students s ON c.class_id = s.class_id JOIN marks m ON s.student_id = m.student_id GROUP BY c.class_name ORDER BY AVG(m.marks) DESC LIMIT 1;

```
+------------+
| class_name |
+------------+
| AB-2 LR-8  |
+------------+
1 row in set (0.00 sec)
```

6.

SELECT c.class_name, (SUM(CASE WHEN m.marks >= 60 THEN 1 ELSE 0 END) * 100.0) / COUNT(m.marks) AS pass_percentage FROM classes c JOIN students s ON c.class_id = s.class_id JOIN marks m ON s.student_id = m.student_id GROUP BY c.class_name;

```
+------------+-----------------+
| class_name | pass_percentage |
+------------+-----------------+
| AB-2 LR-8  |        87.50000 |
| AB-1 LR-3  |        75.00000 |
| AB-1 LR-1  |        45.83333 |
+------------+-----------------+
3 rows in set (0.00 sec)
```

7.

SELECT s.name AS student_name, SUM(m.marks) AS total_marks

FROM students s

JOIN marks m ON s.student_id = m.student_id

GROUP BY s.student_id, s.name

ORDER BY total_marks DESC

LIMIT 1;

```
    -> LIMIT 1;
+--------------+-------------+
| student_name | total_marks |
+--------------+-------------+
| NAMAN KUMAR  |         393 |
+--------------+-------------+
1 row in set (0.00 sec)
```

8.

```
SELECT subject
FROM marks
WHERE marks > 90
GROUP BY subject
ORDER BY COUNT(*) DESC
LIMIT 1;
```

```
+---------+
| subject |
+---------+
| ENG     |
+---------+
1 row in set (0.00 sec)
```

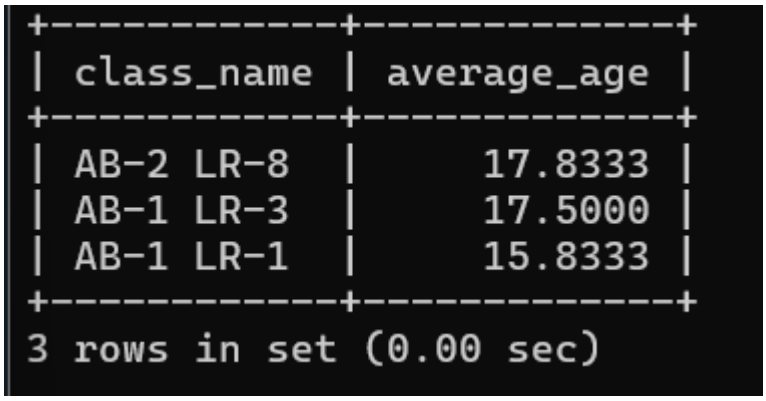
9.

```
SELECT c.class_name,
(COUNT(CASE WHEN s.gender = 'female' THEN 1 ELSE NULL END) * 100.0) /
COUNT(*) AS female_percentage
FROM classes c
JOIN students s ON c.class_id = s.class_id
GROUP BY c.class_name
ORDER BY female_percentage DESC
  LIMIT 1;
```

```
+------------+--------------------+
| class_name | female_percentage  |
+------------+--------------------+
| AB-2 LR-8  |           50.00000 |
+------------+--------------------+
1 row in set (0.00 sec)
```

10.

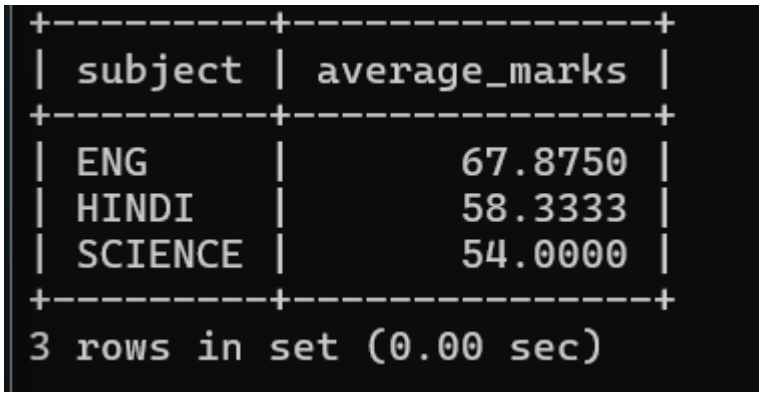
```
SELECT c.class_name, AVG(s.age) AS average_age
FROM classes c
JOIN students s ON c.class_id = s.class_id
GROUP BY c.class_name;
```



```
+------------+-------------+
| class_name | average_age |
+------------+-------------+
| AB-2 LR-8  |     17.8333 |
| AB-1 LR-3  |     17.5000 |
| AB-1 LR-1  |     15.8333 |
+------------+-------------+
3 rows in set (0.00 sec)
```

11.

```
SELECT subject, AVG(marks) AS average_marks
FROM marks
GROUP BY subject
HAVING AVG(marks) < (SELECT AVG(marks) FROM marks);
```



```
+---------+---------------+
| subject | average_marks |
+---------+---------------+
| ENG     |       67.8750 |
| HINDI   |       58.3333 |
| SCIENCE |       54.0000 |
+---------+---------------+
3 rows in set (0.00 sec)
```

12.

```
SELECT s.name AS student_name
FROM students s
JOIN marks m ON s.student_id = m.student_id
JOIN (
   SELECT class_id, AVG(marks) AS average_marks
   FROM students s
   JOIN marks m ON s.student_id = m.student_id
   GROUP BY class_id
   ORDER BY average_marks ASC
   LIMIT 1
) lowest_avg_class ON s.class_id = lowest_avg_class.class_id
GROUP BY s.student_id, s.name
HAVING AVG(m.marks) = (
   SELECT MAX(avg_marks)
   FROM (
      SELECT AVG(marks) AS avg_marks
      FROM marks
      GROUP BY student_id
   ) AS avg_per_student
);
```