(Data Analysis with Complex Queries)

Internship Name: CODTECH SQL Internship

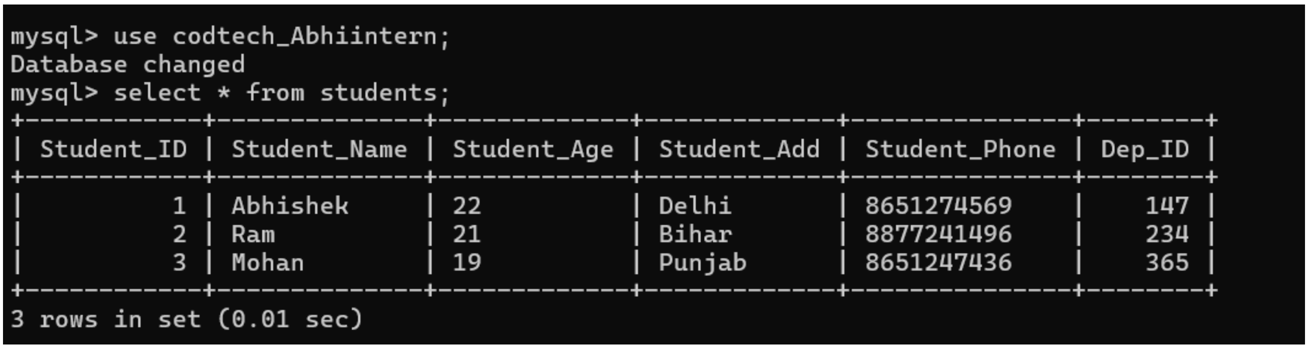
Task: 2

Week 02

Submitted by: Abhishek Kumar Malakar

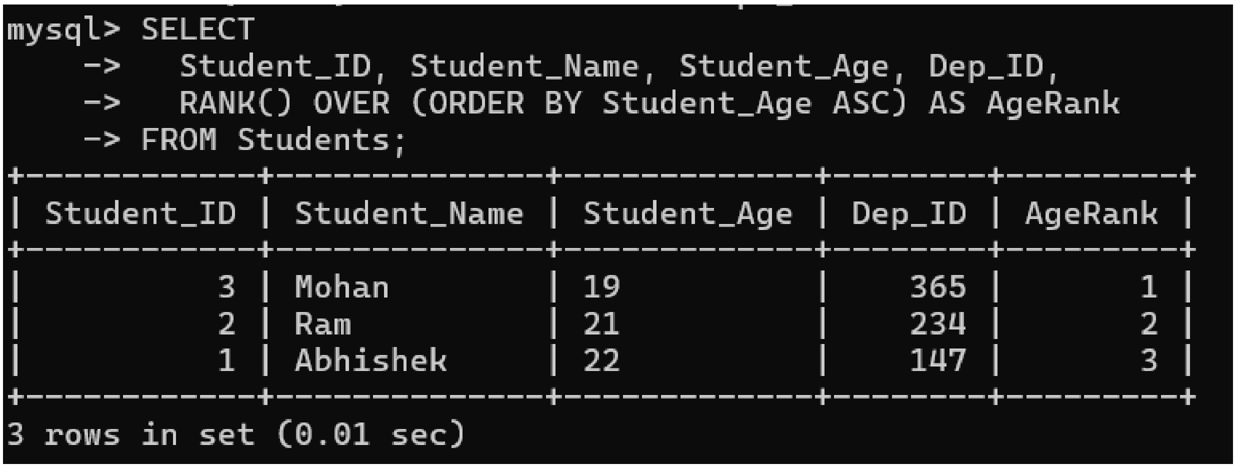
Date: 07-09-2025

Use Data Base



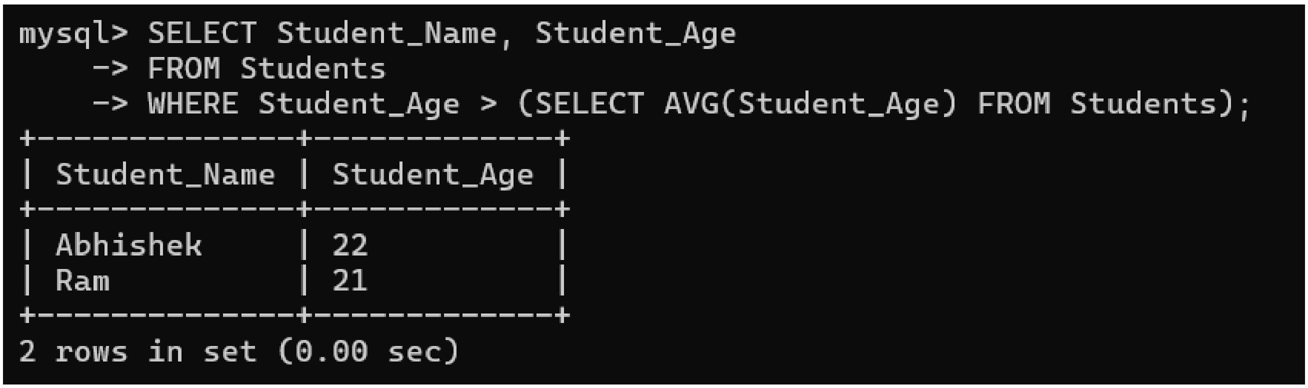
```
mysql> use codtech_Abhiintern;
Database changed
mysql> select * from students;
+------------+--------------+-------------+-------------+---------------+--------+
| Student_ID | Student_Name | Student_Age | Student_Add | Student_Phone | Dep_ID |
+------------+--------------+-------------+-------------+---------------+--------+
|          1 | Abhishek     | 22          | Delhi       | 8651274569    |    147 |
|          2 | Ram          | 21          | Bihar       | 8877241496    |    234 |
|          3 | Mohan        | 19          | Punjab      | 8651247436    |    365 |
+------------+--------------+-------------+-------------+---------------+--------+
3 rows in set (0.01 sec)
```

1. **Window Function – Age Ranking**



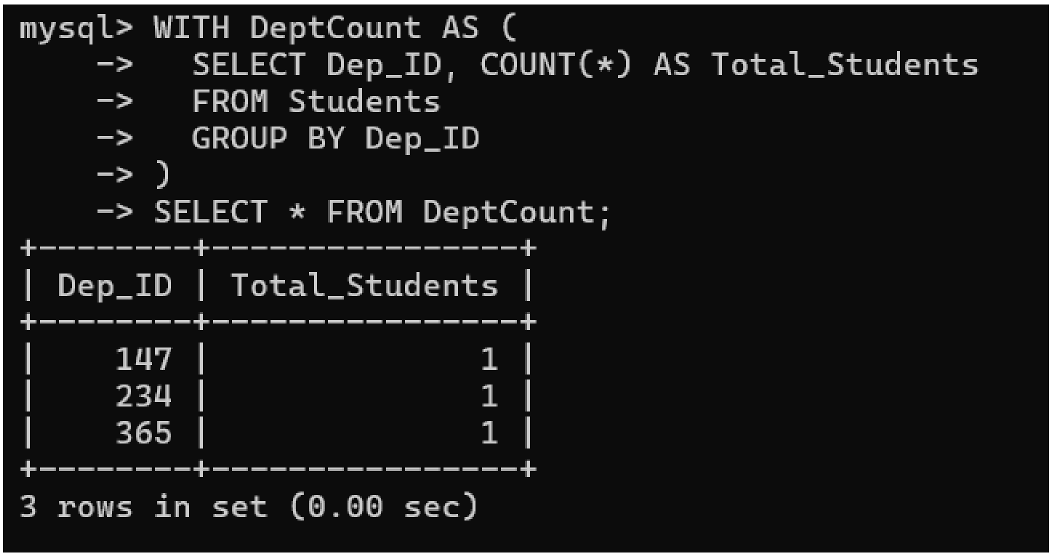
```
mysql> SELECT
    ->   Student_ID, Student_Name, Student_Age, Dep_ID,
    ->   RANK() OVER (ORDER BY Student_Age ASC) AS AgeRank
    -> FROM Students;
+------------+--------------+-------------+--------+---------+
| Student_ID | Student_Name | Student_Age | Dep_ID | AgeRank |
+------------+--------------+-------------+--------+---------+
|          3 | Mohan        | 19          |    365 |       1 |
|          2 | Ram          | 21          |    234 |       2 |
|          1 | Abhishek     | 22          |    147 |       3 |
+------------+--------------+-------------+--------+---------+
3 rows in set (0.01 sec)
```

2. **Subquery – Above Average Age**



```
mysql> SELECT Student_Name, Student_Age
    -> FROM Students
    -> WHERE Student_Age > (SELECT AVG(Student_Age) FROM Students);
+--------------+-------------+
| Student_Name | Student_Age |
+--------------+-------------+
| Abhishek     | 22          |
| Ram          | 21          |
+--------------+-------------+
2 rows in set (0.00 sec)
```

3. **CTE – Department-wise Count**



```
mysql> WITH DeptCount AS (
    ->   SELECT Dep_ID, COUNT(*) AS Total_Students
    ->   FROM Students
    ->   GROUP BY Dep_ID
    -> )
    -> SELECT * FROM DeptCount;
+--------+----------------+
| Dep_ID | Total_Students |
+--------+----------------+
|    147 |              1 |
|    234 |              1 |
|    365 |              1 |
+--------+----------------+
3 rows in set (0.00 sec)
```

Conclusion

- **Window functions make ranking clear (based on Age).**
- **Subqueries easily extract students who are above the average.**
- **CTEs provide a summary of students department-wise.**