SQL Functions

Assignment Questions

Note : Create a table in MySQL Workbench using the following code :

- **SQL Code to Create the Table**

```sql
CREATE TABLE Student_Performance (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    course VARCHAR(30),
    score INT,
    attendance INT,
    mentor VARCHAR(50),
    join_date DATE,
    city VARCHAR(50)
);
```

- **Insert the Data**

```sql
INSERT INTO Student_Performance
(student_id, name, course, score, attendance, mentor,
join_date, city)
VALUES
(101, 'Aarav Mehta', 'Data Science', 88, 92, 'Dr. Sharma',
'2023-06-12', 'Mumbai'),
(102, 'Riya Singh', 'Data Science', 76, 85, 'Dr. Sharma',
'2023-07-01', 'Delhi'),
(103, 'Kabir Khanna', 'Python', 91, 96, 'Ms. Nair',
'2023-06-20', 'Mumbai'),
(104, 'Tanvi Patel', 'SQL', 84, 89, 'Mr.
Iyer','2023-05-30', 'Bengaluru'),
(105, 'Ayesha Khan', 'Python', 67, 81, 'Ms. Nair',

'2023-07-10', 'Hyderabad'),
(106, 'Dev Sharma', 'SQL', 73, 78, 'Mr. Iyer',
'2023-05-28', 'Pune'),
(107, 'Arjun Verma', 'Tableau', 95, 98, 'Ms. Kapoor',
'2023-06-15', 'Delhi'),
(108, 'Meera Pillai', 'Tableau', 82, 87, 'Ms. Kapoor',
'2023-06-18', 'Kochi'),
(109, 'Nikhil Rao', 'Data Science', 79, 82, 'Dr. Sharma',
'2023-07-05', 'Chennai'),
(110, 'Priya Desai', 'SQL', 92, 94, 'Mr. Iyer',
'2023-05-27', 'Bengaluru'),
(111, 'Siddharth Jain', 'Python', 85, 90, 'Ms. Nair',
'2023-07-02', 'Mumbai'),
(112, 'Sneha Kulkarni', 'Tableau', 74, 83, 'Ms. Kapoor',
'2023-06-10', 'Pune'),
(113, 'Rohan Gupta', 'SQL', 89, 91, 'Mr. Iyer',
'2023-05-25', 'Delhi'),
(114, 'Ishita Joshi', 'Data Science', 93, 97, 'Dr. Sharma',
'2023-06-25', 'Bengaluru'),
(115, 'Yuvraj Rao', 'Python', 71, 84, 'Ms. Nair',
'2023-07-12', 'Hyderabad');
```

Question 1 : Create a ranking of students based on score (highest first).

Question 2 : Show each student's score and the previous student's score (based on score order).

Question 3 : Convert all student names to uppercase and extract the month name from join_date.

Question 4 : Show each student's name and the next student's attendance (ordered by attendance).

Question 5 : Assign students into 4 performance groups using NTILE().

Question 6 : For each course, assign a row number based on attendance (highest first).

Question 7 : Calculate the number of days each student has been enrolled (from join_date to today).
(Assume current date = '2025-01-01')

Question 8 : Format join_date as "Month Year" (e.g., "June 2023").

Question 9 : Replace the city 'Mumbai' with 'MUM' for display purposes.

Question 10 : For each course, find the highest score using FIRST_VALUE().