

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

- SQL Code to Create the Table

```
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

```
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).

Ans:-

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `student_db` selected.
- Query Editor:** Contains the following SQL code:

```
40
41  ## Create a ranking of students based on score (highest first).
42
43 • SELECT name, score,
44   RANK() OVER (ORDER BY score DESC) AS rank_position
45   FROM Student_Performance;
46
47
48
49
```
- Result Grid:** Displays the results of the query:

name	score	rank_position
Arjun Verma	95	1
Ishita Joshi	93	2
Priya Desai	92	3
Kabir Khanna	91	4
Rohan Gupta	89	5
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
4	01:18:55	INSERT INTO Student_Performance (student_id, name, course, score, attendance, mentor, join_date, city)	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.000 sec
5	01:19:18	select * from Student_Performance LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec
6	01:27:17	SELECT name, score, RANK() OVER (ORDER BY score DESC) AS rank_position FROM Student_Performa...	15 row(s) returned	0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `student_db` selected.
- Query Editor:** Contains the following SQL code:

```
47
48  ## Show each student's score and the previous student's score (based on score order).
49
50 • SELECT name, score,
51   LAG(score) OVER (ORDER BY score DESC) AS previous_score
52   FROM Student_Performance;
53
54
55
56
```
- Result Grid:** Displays the results of the query:

name	score	previous_score
Arjun Verma	95	NULL
Ishita Joshi	93	95
Priya Desai	92	93
Kabir Khanna	91	92
Rohan Gupta	89	91
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
5	01:19:18	select * from Student_Performance LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec
6	01:27:17	SELECT name, score, RANK() OVER (ORDER BY score DESC) AS rank_position FROM Student_Performa...	15 row(s) returned	0.000 sec / 0.000 sec
7	01:28:44	SELECT name, score, LAG(score) OVER (ORDER BY score DESC) AS previous_score FROM Student_Perf...	15 row(s) returned	0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query converts student names to uppercase and extracts the month from their join date:

```
53
54
55
56 ## Convert all student names to uppercase and extract the month name from join_date.
57
58 • SELECT UPPER(name) AS name_upper,
59     MONTHNAME(join_date) AS join_month
60 FROM Student_Performance;
61
62
```

The results grid displays the converted data:

	name_upper	join_month
1	AARAV MEHTA	June
2	RITYA SINGH	July
3	KABIR KHANNA	June
4	TANVI PATEL	May
5	AYESHA KHAN	July

The output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
6	01:27:17	SELECT name, score, RANK() OVER (ORDER BY score DESC) AS rank_position FROM Student_Performa...	15 row(s) returned	0.000 sec / 0.000 sec
7	01:28:44	SELECT name, score, LAG(score) OVER (ORDER BY score DESC) AS previous_score FROM Student_Perf...	15 row(s) returned	0.000 sec / 0.000 sec
8	01:29:29	SELECT UPPER(name) AS name_upper, MONTHNAME(join_date) AS join_month FROM Student_Performa...	15 row(s) returned	0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query retrieves student names and the next student's attendance, ordered by attendance:

```
61
62
63
64 ## Show each student's name and the next student's attendance (ordered by attendance).
65
66 • SELECT name, attendance,
67     LEAD(attendance) OVER (ORDER BY attendance DESC) AS next_attendance
68 FROM Student_Performance;
69
70
```

The results grid displays the data:

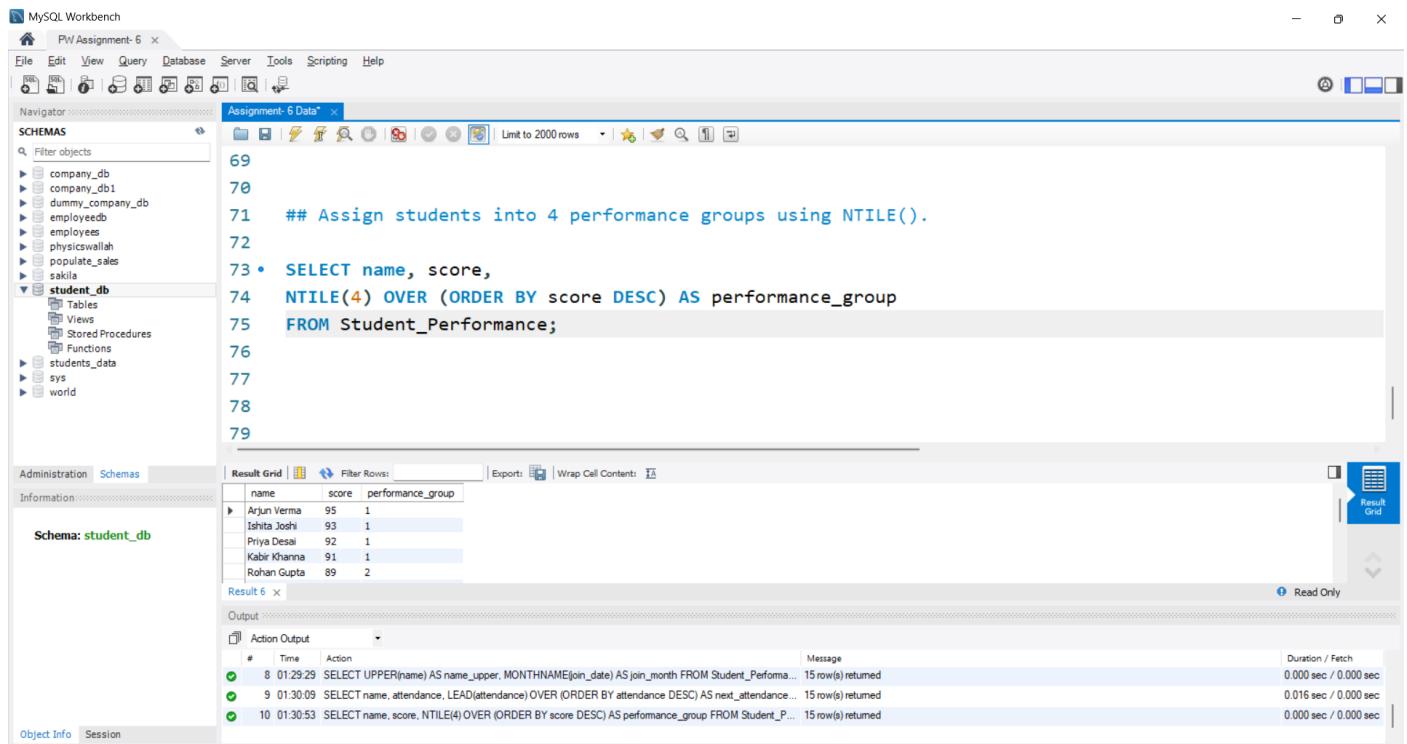
	name	attendance	next_attendance
1	Arjun Verma	98	97
2	Ishita Joshi	97	96
3	Kabir Khanna	96	94
4	Priya Desai	94	92
5	Aarav Mehta	92	91

The output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
7	01:28:44	SELECT name, score, LAG(score) OVER (ORDER BY score DESC) AS previous_score FROM Student_Perf...	15 row(s) returned	0.000 sec / 0.000 sec
8	01:29:29	SELECT UPPER(name) AS name_upper, MONTHNAME(join_date) AS join_month FROM Student_Perfoma...	15 row(s) returned	0.000 sec / 0.000 sec
9	01:30:09	SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance DESC) AS next_attendance...	15 row(s) returned	0.016 sec / 0.000 sec

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

Ans:-



MySQL Workbench - PW Assignment- 6

File Edit View Query Database Server Tools Scripting Help

Navigator SCHEMAS Filter objects company_db company_db1 dummy_company_db employee_db employees physicswallah populate_sales sakila student_db Tables Views Stored Procedures Functions students_data sys world

Assignment- 6 Data* Limit to 2000 rows

```
69
70
71 ## Assign students into 4 performance groups using NTILE().
72
73 • SELECT name, score,
74 NTILE(4) OVER (ORDER BY score DESC) AS performance_group
75 FROM Student_Performance;
76
77
78
79
```

Result Grid Filter Rows Export Wrap Cell Content Result Grid Read Only

name	score	performance_group
Arjun Verma	95	1
Ishita Joshi	93	1
Priya Desai	92	1
Kabir Khanna	91	1
Rohan Gupta	89	2

Result 6 x

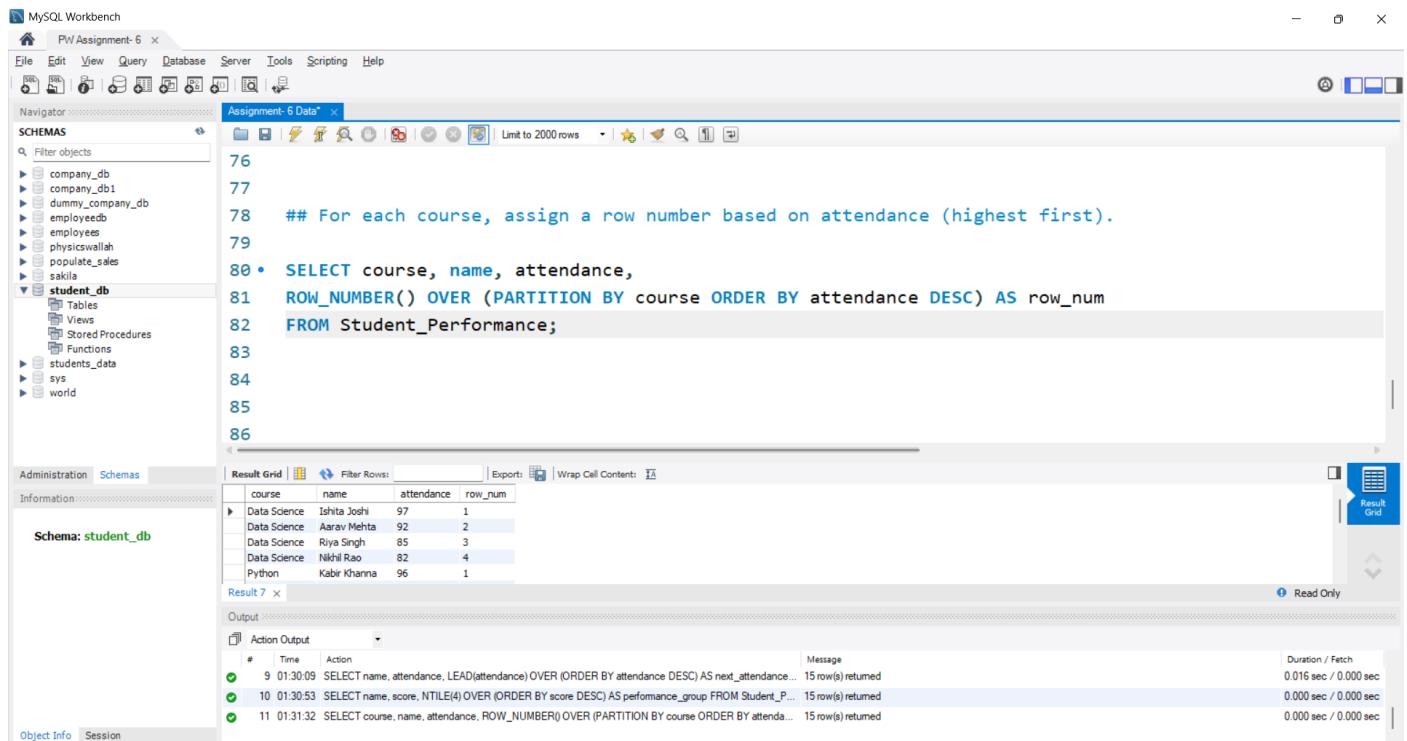
Output Action Output

#	Time	Action	Message	Duration / Fetch
8	01:29:29	SELECT UPPER(name) AS name_upper, MONTHNAME(join_date) AS join_month FROM Student_Performance;	15 row(s) returned	0.000 sec / 0.000 sec
9	01:30:09	SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance DESC) AS next_attendance...	15 row(s) returned	0.016 sec / 0.000 sec
10	01:30:53	SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM Student_Performance;	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

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

Ans:-



MySQL Workbench - PW Assignment- 6

File Edit View Query Database Server Tools Scripting Help

Navigator SCHEMAS Filter objects company_db company_db1 dummy_company_db employee_db employees physicswallah populate_sales sakila student_db Tables Views Stored Procedures Functions students_data sys world

Assignment- 6 Data* Limit to 2000 rows

```
76
77
78 ## For each course, assign a row number based on attendance (highest first).
79
80 • SELECT course, name, attendance,
81 ROW_NUMBER() OVER (PARTITION BY course ORDER BY attendance DESC) AS row_num
82 FROM Student_Performance;
83
84
85
86
```

Result Grid Filter Rows Export Wrap Cell Content Result Grid Read Only

course	name	attendance	row_num
Data Science	Ishita Joshi	97	1
Data Science	Aarav Mehta	92	2
Data Science	Riya Singh	85	3
Data Science	Nikhil Rao	82	4
Python	Kabir Khanna	96	1

Result 7 x

Output Action Output

#	Time	Action	Message	Duration / Fetch
9	01:30:09	SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance DESC) AS next_attendance...	15 row(s) returned	0.016 sec / 0.000 sec
10	01:30:53	SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM Student_Performance;	15 row(s) returned	0.000 sec / 0.000 sec
11	01:31:32	SELECT course, name, attendance, ROW_NUMBER() OVER (PARTITION BY course ORDER BY attendance DESC) AS row_num FROM Student_Performance;	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Question 7 : Calculate the number of days each student has been enrolled (from join_date to today).

(Assume current date = '2025-01-01')

Ans:-

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** student_db
- Query Editor:** Assignment-6 Data*

```
83
84    ## each student has been enrolled (from join_date to today). (Assume current date = '2025-01-01')
85
86
87 • SELECT name,
88     DATEDIFF('2025-01-01', join_date) AS days_enrolled
89     FROM Student_Performance;
90
91
92
93
```

- Result Grid:** Shows the results of the query:

name	days_enrolled
Aarav Mehta	569
Riya Singh	550
Kabir Khanna	561
Tanvi Patel	582
Ayesha Khan	541

- Action Output:** Displays the execution log:

#	Time	Action	Message	Duration / Fetch
10	01:30:53	SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM Student_P...	15 row(s) returned	0.000 sec / 0.000 sec
11	01:31:32	SELECT course, name, attendance, ROW_NUMBER() OVER (PARTITION BY course ORDER BY attenda...	15 row(s) returned	0.000 sec / 0.000 sec
12	01:33:14	SELECT name, DATEDIFF('2025-01-01',join_date) AS days_enrolled FROM Student_Performance LIMIT 0...	15 row(s) returned	0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** student_db
- Query Editor:** Assignment-6 Data*

```
90
91
92    ## Format join_date as “Month Year” (e.g., “June 2023”).
93
94 • SELECT name,
95     DATE_FORMAT(join_date, '%M %Y') AS formatted_join_date
96     FROM Student_Performance;
97
98
99
100
```

- Result Grid:** Shows the results of the query:

name	formatted_join_date
Aarav Mehta	June 2023
Riya Singh	July 2023
Kabir Khanna	June 2023
Tanvi Patel	May 2023
Ayesha Khan	July 2023

- Action Output:** Displays the execution log:

#	Time	Action	Message	Duration / Fetch
11	01:31:32	SELECT course, name, attendance, ROW_NUMBER() OVER (PARTITION BY course ORDER BY attenda...	15 row(s) returned	0.000 sec / 0.000 sec
12	01:33:14	SELECT name, DATEDIFF('2025-01-01',join_date) AS days_enrolled FROM Student_Performance LIMIT 0...	15 row(s) returned	0.000 sec / 0.000 sec
13	01:33:57	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS formatted_join_date FROM Student_Performance ...	15 row(s) returned	0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with a query editor titled "Assignment- 6 Data". The code is as follows:

```
97
98
99 ## Replace the city 'Mumbai' with 'MUM' for display purposes.
100
101 • SELECT name,
102 REPLACE(city, 'Mumbai', 'MUM') AS city_display
103 FROM Student_Performance;
104
105
106
107
```

The result grid shows the output of the query:

name	city_display
Aarav Mehta	MUM
Riya Singh	Delhi
Kabir Khanna	MUM
Tanvi Patel	Bengaluru
Ayesha Khan	Hyderabad

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
12	01:33:14	SELECT name, DATEDIFF(2025-01-01',join_date) AS days_enrolled FROM Student_Performance LIMIT 0... 15 row(s) returned		0.000 sec / 0.000 sec
13	01:33:57	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS formatted_join_date FROM Student_Performance ... 15 row(s) returned		0.000 sec / 0.000 sec
14	01:35:23	SELECT name, REPLACE(city, 'Mumbai', 'MUM') AS city_display FROM Student_Performance LIMIT 0, 2000 15 row(s) returned		0.000 sec / 0.000 sec

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

Ans:-

The screenshot shows the MySQL Workbench interface with a query editor titled "Assignment- 6 Data". The code is as follows:

```
105
106 ## For each course, find the highest score using FIRST_VALUE().
107
108 • SELECT course, name, score,
109 FIRST_VALUE(score) OVER (PARTITION BY course ORDER BY score DESC) AS highest_score_in_course
110 FROM Student_Performance;
111
```

The result grid shows the output of the query:

course	name	score	highest_score_in_course
Data Science	Ishita Joshi	93	93
Data Science	Aarav Mehta	88	93
Data Science	Nikhil Rao	79	93
Data Science	Riya Singh	76	93
Python	Kabir Khanna	91	91
Python	Siddharth Jain	85	91
Python	Yuvraaj Rao	71	91
Python	Ayesha Khan	67	91
SQL	Priya Desai	92	92
SQL	Rohan Gupta	89	92
SQL	Tanvi Patel	84	92
SQL	Dev Sharma	73	92

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	01:33:57	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS formatted_join_date FROM Student_Performance ... 15 row(s) returned		0.000 sec / 0.000 sec
14	01:35:23	SELECT name, REPLACE(city, 'Mumbai', 'MUM') AS city_display FROM Student_Performance LIMIT 0, 2000 15 row(s) returned		0.000 sec / 0.000 sec
15	01:36:01	SELECT course, name, score, FIRST_VALUE(score)OVER (PARTITION BY course ORDER BY score DE... 15 row(s) returned		0.000 sec / 0.000 sec