

SQL Functions

MySQL Workbench

Schemas:

- mitesh
- orders
- product
 - Tables: products, sales
 - Views: vw_category_summary, vw_productinfo
 - Stored Procedures: GetProductsByCategory
 - Functions
- pw_skills
- pwskills
- sakila
- students
 - Tables: student_performance
 - Views

Table: student_performance

Columns:

student_id	name	course	score	attendance	mentor	join_date	city
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	Mr.Nair	2023-06-20	Mumbai
104	Tanvi patel	SQL	84	89	Mr.Iyyer	2023-05-30	Bengaluru
105	Ayesha khan	python	67	81	Mr.Nair	2023-07-10	Hyderabad
106	Dev sharma	SQL	73	78	Mr.Iyyer	2023-06-18	Pune
107	Arjun varma	Tableau	95	98	Mr. Kapoor	2023-06-18	Delhi
108	Meera pillai	Tableau	82	87	Mr. 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. Iyyer	2023-05-27	Bengaluru
111	Siddharth Jain	python	85	90	Mr.Nair	2023-07-02	Mumbai
112	Sneha Kulkarni	Tableau	74	83	Mr.kapoor	2023-06-10	pune
113	Rohan Gupta	SQL	89	91	Mr.Iyyer	2023-05-25	Delhi
114	Ishita Joshi	Data Science	93	97	Dr.Sharma	2023-06-25	Bengaluru
115	Yuvraj Rao	Python	71	84	Mr. Nair	2023-07-12	Hyderabad

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
13	13:27:49	insert into student_performance values (102,'Riya Singh','Data Science',76,85,'Dr.Sharma','2023-07-01','Delhi')	Error Code: 1136. Column count doesn't match value count at row 9	0.000 sec
14	13:30:15	insert into student_performance values (102,'Riya Singh','Data Science',76,85,'Dr.Sharma','2023-07-01','Delhi')	14 row(s) affected Records: 14 Duplicates: 0 Warnings: 0	0.016 sec
15	13:30:21	select * from student_performance	15 row(s) returned	0.000 sec / 0.000 sec

SQL Functions

QUESTION 1

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

ANSWER

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

- Schemas:** The current schema is "pw_skills_DA_Batch".
- Tables:** The "students" table is selected.
- Code Editor:** The code is as follows:

```
33 (114,'Ishita Joshi','Data Science',93,97,'Dr.Sharma','2023-06-25','Bengaluru'),  
34 (115,'Yuvraj Rao','Python',71,84,'Mr. Nain','2023-07-12','Hydrabad');  
35  
36 • select * from student_performance;  
37  
38 ## QUESTION 1 (Create a ranking of students based on score (highest first)).  
39  
40 • SELECT  
    student_id,  
    name,  
    score,  
    RANK() OVER (ORDER BY score DESC) AS rank_no  
FROM student_performance;  
41  
42  
43  
44  
45  
46  
47  
48
```

Result Grid: The results show the student_id, name, score, and rank_no for five students:

student_id	name	score	rank_no
107	Arjun varma	95	1
114	Ishita Joshi	93	2
110	Priya Desai	92	3
103	Kabir khanna	91	4
113	Rohan Gupta	89	5

Action Output: The log shows the execution of the query and the creation of the ranking function.

QUESTION 2

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

ANSWER

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

- Schemas:** The current schema is "pw_skills_DA_Batch".
- Tables:** The "students" table is selected, showing its columns: student_id, name, score, and previous_score.
- Query Editor:** The query being run is:

```
student_id,
student_name,
score,
ROW_NUMBER() OVER (ORDER BY score DESC) AS rank_no
FROM students_performance;
```

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

```
SELECT
name,
score,
```
- Result Grid:** The results show the student names and scores from highest to lowest, with the previous score for each row.

name	score	previous_score
Arjun varma	95	NULL
Ishita Joshi	93	95
Priya Desai	92	93
Kabir khanna	91	92
Rohan Gupta	89	91
Aarav Mehta	88	89
Siddharth Jain	85	88
Tanvi Patel	84	85
Meera Pillai	82	84
Nikhil Rao	79	82
Riya Singh	76	79
Sneha Kulkarni	74	76
Dev Sharma	73	74
Muskan	71	73
- Action Output:** The log shows the execution of three statements, with the third one being successful.

#	Time	Action	Message	Duration / Fetch
20	15:11:42	SELECT student_id, student_name, score, ROW_NUMBER() OVER (ORDER BY score DESC) AS rank_no FROM students_performance;	Error Code: 1146. Table 'students.students_performance' doesn't exist.	0.000 sec
21	15:11:56	SELECT name, score, LAG(score) OVER (ORDER BY score DESC) AS previous_score FROM students.students_performance;	Error Code: 1146. Table 'students.students_performance' doesn't exist.	0.000 sec
22	15:12:15	SELECT name, score, LAG(score) OVER (ORDER BY score DESC) AS previous_score FROM students.students_performance;	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.

ANSWER

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

- Schemas:** The current schema is "pw_skills_DA_Batch".
- Tables:** The "students" table is selected, showing the "student_performance" table structure.
- Query Editor:** The query is:53 ## QUESTION 3
54
55 • SELECT
56 UPPER(name) AS student_name_upper,
57 MONTHNAME(join_date) AS join_month
58 FROM student_performance;
59
60
61
62
63
- Result Grid:** The results show student names converted to uppercase and their joining months.

student_name_upper	join_month
AARAV MEHTA	June
RIYA SINGH	July
KABIR KHANNA	June
TANVI PATEL	May
AYESHA KHAN	July
DEV SHARMA	May
ARJUN VARMA	June
MEERA ILLAI	June
NIDHI RAO	July
PRIYA DESAI	May
SIDDHARTH JAIN	July
SNEHA KULKARNI	June

- Action Output:** The log shows the execution of three statements. Statement 23 fails because it references the column "student_name" which does not exist in the current context.

QUESTION 4

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

ANSWER

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator joins* ADVANCE SQL ASINGMENT FUNCTUONS ASSINGMENT*

SCHEMAS Filter objects

- mitesh
- orders
- product
 - Tables
 - products
 - sales
 - Views
 - vw_category_summary
 - vw_productinfo
 - Stored Procedures
 - GetProductsByCategory
 - Functions
 - pw_skills
 - pwskills
 - sakila
 - students
 - Tables
 - student_performance
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views

Administration Schemas Information

Table: student_performance

Columns:

student_id	int PK
name	varchar(50)
course	varchar(30)
score	int
attendance	int
mentor	varchar(50)
join_date	date
city	varchar(50)

Object Info Session

```
58     FROM student_performance;
59
60     ## QUESTION 4
61 •  SELECT
62     name,
63     attendance,
64     LEAD(attendance) OVER (ORDER BY attendance) AS next_student_attendance
65     FROM student_PERFORMANCE;
```

Result Grid Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types Query Stats Read Only

name	attendance	next_student_attendance
Dev sharma	78	81
Ayesha khan	81	82
Nikhil Rao	82	83
Sneha Kulkarni	83	84
Yuvraj Rao	84	85
Riya Singh	85	87
Meera pillai	87	89
Tanvi patel	89	90
Siddharth Jain	90	91
Rohan Gupta	91	92
Aarav Mehta	92	94
Priya Desai	94	96
Kabir khanna	96	97
Ishita Joshi	97	98
Arjun varma	98	NULL

Result 9 x Output Action Output

#	Time	Action	Message	Duration / Fetch
27	15:51:57	SELECT student_name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_s... Error Code: 1146. Table 'students.students' doesn't exist		0.000 sec
28	15:52:11	SELECT student_name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_s... Error Code: 1054. Unknown column 'student_name' in field list'		0.000 sec
29	15:52:21	SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_student_a... 15 row(s) returned		0.000 sec / 0.000 sec

QUESTION 5**Assign students into 4 performance groups using NTILE().****ANSWER**

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator ADVANCE SQL ASINGMENT FUNCTUNGS ASSINGMENT* joins*
Schemas Filter objects
mitesh
orders
product
Tables products sales
Views vw_category_summary vw_productinfo
Stored Procedures GetProductsByCategory
Functions pw_skills
pwskills
sakila
students
Tables student_performance
Columns name score attendance
Indexes
Foreign Keys
Triggers
Views
Administration Schemas
Information Table: student_performance
Columns:
student_id int PK
name varchar(50)
course varchar(30)
score int
attendance int
mentor varchar(50)
join_date date
city varchar(50)
Object Info Session
Result Grid Filter Rows Export Wrap Cell Content Result Grid
name score performance_group
Arjun varma 95 1
Ishita Joshi 93 1
Priya Desai 92 1
Kabir khanna 91 1
Rohan Gupta 89 2
Aarav Mehta 88 2
Siddharth Jain 85 2
Tanvi patel 84 2
Meera pillai 82 3
Nikhil Rao 79 3
Riya Singh 76 3
Sneha Kulkarni 74 3
Dev sharma 73 4
Yuvraj Rao 71 4
Ayesha khan 67 4
Result 10 x Read Only
Output Action Output
Time Action Message Duration / Fetch
28 15:52:11 SELECT student_name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_s... Error Code: 1054. Unknown column 'student_name' in field list 0.000 sec
29 15:52:21 SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_student_a... 15 row(s) returned 0.000 sec / 0.000 sec
30 17:13:55 SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM stu... 15 row(s) returned 0.000 sec / 0.000 sec
Field Types Query Stats

QUESTION 6

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

ANSWER

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

Query Editor:

```
name,
course,
attendance,
ROW_NUMBER() OVER (
    PARTITION BY course
    ORDER BY attendance DESC
) AS row_no
FROM student_performance;
```

Result Grid:

	name	course	attendance	row_no
1	Ishita Joshi	Data Science	97	1
2	Aarav Mehta	data science	92	2
3	Riya Singh	Data Science	85	3
4	Nikki Rao	Data science	82	4
5	Kabir khanna	Python	96	1
6	Siddharth Jain	python	90	2
7	Yuvraj Rao	Python	84	3
8	Ayesha khan	python	81	4
9	Priya Desai	SQL	94	1
10	Rohan Gupta	SQL	91	2
11	Tanvi Patel	SQL	89	3
12	Dev Sharma	SQL	78	4
13	Arjun Varma	Tableau	98	1
14	Meera Pillai	Tableau	87	2
15	Sneha Kulkarni	Tableau	83	3

Action Output:

#	Time	Action	Message	Duration / Fetch
29	15:52:21	SELECT name, attendance, LEAD(attendance) OVER (ORDER BY attendance) AS next_student_a...	15 row(s) returned	0.000 sec / 0.000 sec
30	17:13:55	SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM stu...	15 row(s) returned	0.000 sec / 0.000 sec
31	17:21:47	SELECT name, course, attendance, ROW_NUMBER() OVER (PARTITION BY course ...	15 row(s) returned	0.000 sec / 0.000 sec

QUESTION 7

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

ANSWER

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

- File:** ADVANCE SQL ASINGMENT
- Table:** student_performance
- Query:**

```
85      FROM student_performance;
86
87  ##  QUESTION 7
88 •  SELECT
89      name,
90      join_date,
91      DATEDIFF('2025-01-01', join_date) AS days_enrolled
92  FROM student_PERFORMANCE;
```

- Result Grid:** Displays the results of the query.

name	join_date	days_enrolled
Aarav Mehta	2023-06-12	569
Riya Singh	2023-07-01	550
Kabir khanna	2023-06-20	561
Tanvi patel	2023-05-30	582
Ayesha khan	2023-07-10	541
Dev sharma	2023-05-28	584
Arjun varma	2023-06-18	563
Meera pillai	2023-06-18	563
Nikhil Rao	2023-07-05	546
Priya Desai	2023-05-27	585
Siddharth Jain	2023-07-02	549
Sneha Kulkarni	2023-06-10	571
Rohan Gupta	2023-05-25	587
Ishita Joshi	2023-06-25	556
Yuvraj Rao	2023-07-12	539

- Output:** Shows the execution log.

#	Time	Action	Message	Duration / Fetch
30	17:13:55	SELECT name, score, NTILE(4) OVER (ORDER BY score DESC) AS performance_group FROM stu...	15 row(s) returned	0.000 sec / 0.000 sec
31	17:21:47	SELECT name, course, attendance, ROW_NUMBER() OVER (PARTITION BY course ...	15 row(s) returned	0.000 sec / 0.000 sec
32	17:25:46	SELECT name, join_date, DATEDIFF('2025-01-01',join_date) AS days_enrolled FROM student_PE...	15 row(s) returned	0.015 sec / 0.000 sec

QUESTION 8

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

ANSWER

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

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbars:** Standard toolbar with icons for file operations, database navigation, and search.
- Navigator:** Shows the database schema with the following structure:
 - SCHEMAS:** mitesh, orders, product (Tables: products, sales), vw_category_summary, vw_productinfo, GetProductsByCategory, Functions, pw_skills, pwskills, sakila, students (Tables: student_performance).
 - Tables:** student_performance (Columns: student_id, name, course, score, attendance, mentor, join_date, city).
- Query Editor:** Contains the following SQL code:

```
91    DATEDIFF('2025-01-01', join_date) AS days_enrolled
92    FROM student_PERFORMANCE;
93
94    ## question 8
95 • SELECT
96        name,
97        DATE_FORMAT(join_date, '%M %Y') AS join_month_year
98    FROM student_performance;
```
- Result Grid:** Displays the results of the query:

name	join_month_year
Aarav Mehta	June 2023
Riya Singh	July 2023
Kabir khanna	June 2023
Tanvi patel	May 2023
Ayesha khan	July 2023
Dev sharma	May 2023
Arjun varma	June 2023
Meera pilai	June 2023
Nikhil Rao	July 2023
Priya Desai	May 2023
Siddharth Jain	July 2023
Sneha Kulkarni	June 2023
Rohan Gupta	May 2023
Ishita Joshi	June 2023
Yuvraj Rao	July 2023
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
35	17:35:30	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS join_month_year FROM student_performance	15 row(s) returned	0.000 sec / 0.000 sec
36	17:35:37	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS join_month_year FROM student_performance	15 row(s) returned	0.015 sec / 0.000 sec
37	17:35:48	SELECT name, DATE_FORMAT(join_date, '%M %Y') AS join_month_year FROM student_performance	15 row(s) returned	0.000 sec / 0.000 sec

QUESTION 9

Replace the city 'Mumbai' with 'MUM' for display purposes.

ANSWER

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

- Schemas:** The current schema is "product".
- Tables:** The "student_performance" table is selected.
- Columns:** The columns are listed as: student_id (int PK), name (varchar(50)), course (varchar(30)), score (int), attendance (int), mentor (varchar(50)), join_date (date), and city (varchar(50)).
- SQL Editor:** The code entered is:

```
97    DATE_FORMAT(join_date, '%M %Y') AS join_month_year
98  FROM student_performance;
99
100 ## QUESTION 9 Replace the city 'Mumbai' with 'MUM' for display purposes.
101 • SELECT
102   name,
103   REPLACE(city, 'Mumbai', 'MUM') AS city_display
104  FROM student_performance;
105
106
107
108
109
110
111
```
- Result Grid:** The results show the names and city displays for five students. The city 'Mumbai' has been replaced with 'MUM'.

name	city_display
Kabir khanna	MUM
Tanvi patel	Bengaluru
Ayesha khan	Hydrabad
Dev sharma	Pune
Arjun varma	Delhi
- Action Output:** The log shows three actions: 36, 37, and 38, all returning 15 row(s) returned with 0.015 sec / 0.000 sec duration.

QUESTION 10

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

ANSWER

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

- Title Bar:** MySQL Workbench - pw_skills_DA_Batch
- Menu Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Includes icons for Home, New, Open, Save, Run, Stop, Refresh, Undo, Redo, Find, Replace, Copy, Paste, and Delete.
- Schemas Navigator:** Shows the database structure with Schemas (mitesh, orders, product, views, stored procedures, functions, pw_skills, pwskills, sakila, students), Tables (product, vw_category_summary, vw_productinfo, GetProductsByCategory, student_performance), Columns, Indexes, Foreign Keys, and Triggers.
- Current Schema:** students
- Query Editor:** ADVANCE SQL ASINGMENT - FUNCTIONS ASSIGMENT*

```
101 • SELECT
102     name,
103     REPLACE(city, 'Mumbai', 'MUM') AS city_display
104 FROM student_performance;
105 ## QUESTION 10
106 • SELECT
107     name,
108     course,
109     score,
110     FIRST_VALUE(score) OVER (
111         PARTITION BY course
112         ORDER BY score DESC
113 )
```

- Result Grid:** Displays the results of the query, showing student names, courses, scores, and their highest score in each course.

name	course	score	highest_score_in_course
Ishita Joshi	Data Science	93	93
Aarav Mehta	data science	88	93
Nikhil Rao	Data science	79	93
Riya Singh	Data Science	76	93
Kabir khanna	Python	91	91
Siddharth Jain	python	85	91
Yuvraj Rao	Python	71	91
Ayesha Khan	python	67	91
Priya Desai	SQL	92	92

- Action Output:** Shows the history of actions taken in the session, including failed attempts to select from the student_name column and a successful execution of the query.