

Assignment - 1

Question

1) Create any 2 tables of your choice.

2) Apply:-

1. TIME functions
2. ROW_NUMBER
3. RANK
4. DENSE_RANK
5. PARTITION_BY
6. FIRST_VALUE
7. PERCENT_RANK
8. All join operations

Queries:

Here I am going to create 2 tables named “Teams” And “Reviews”

1.Table 1 - “Teams”

Query:

```
CREATE TABLE teams (
    team_id INT PRIMARY KEY,
    team_name VARCHAR(50),
    guide_name VARCHAR(50),
    department VARCHAR(5),
    projectstart_date DATE
);
```

Table 2 - “Reviews”

Query:

```
CREATE TABLE reviews (
    review_id INT PRIMARY KEY,
    team_id INT,
    project_name VARCHAR(100),
    score INT,
    review_round VARCHAR(20),
    review_time DATETIME
);
```

INSERTING DATA Into table 1

INSERT INTO teams VALUES

```
(101, 'Batch 1', 'Mr. Rao', 'CSM', '2026-01-03'),  
(102, 'Batch 2', 'Ms. Lakshmi', 'CSM', '2026-01-05'),  
(103, 'Batch 3', 'Mr. Suresh', 'CSM', '2026-01-07'),  
(104, 'Batch 4', 'Ms. Priya', 'CSM', '2026-01-09');
```

INSERTING DATA Into table 2

INSERT INTO reviews VALUES

```
(1, 101, 'College Grievance System', 82, 'Round 1', '2024-09-01 10:00:00'),  
(2, 102, 'Face liveness Detection', 91, 'Round 1', '2024-09-01 10:30:00'),  
(3, 103, 'Library Management System', 82, 'Round 1', '2024-09-01 11:00:00'),  
(4, 104, 'ChatBot Application', 76, 'Round 1', '2024-09-01 11:30:00');
```

Inorder to display all the above info use the query's

```
SELECT * FROM teams;  
SELECT * FROM reviews;
```

2.(1)Time Functions

-CURDATE()-

```
SELECT team_name,  
DATEDIFF(CURDATE(), projectstart_date) AS days_active  
FROM teams;
```

-CURETIME()-

```
SELECT CURTIME();
```

2.(2) ROW_NUMBER

```
SELECT project_name, score,  
ROW_NUMBER() OVER (ORDER BY score DESC) AS row_no  
FROM reviews;
```

2.(3) RANK

```
SELECT project_name, score,  
RANK() OVER (ORDER BY score DESC) AS rank_no  
FROM reviews;
```

2.(4) DENSE RANK

```
SELECT project_name, score,  
DENSE_RANK() OVER (ORDER BY score DESC) AS dense_rank_no  
FROM reviews;
```

2.(5) PARTITION_BY

```
SELECT t.team_name, t.guide_name, r.score,  
RANK() OVER (  
    PARTITION BY t.guide_name  
    ORDER BY r.score DESC  
) AS guide_rank  
FROM teams t  
JOIN reviews r  
ON t.team_id = r.team_id;
```

2.(6) FIRST_VALUE

```
SELECT review_round, project_name, score,  
FIRST_VALUE(score)  
OVER (PARTITION BY review_round ORDER BY score DESC) AS  
highest_score  
FROM reviews;
```

2.(7). PERCENT_RANK

```
SELECT project_name, score,  
PERCENT_RANK() OVER (ORDER BY score) AS percent_rank  
FROM reviews;
```

2.(8) ALL JOIN OPERATIONS

-INNER JOIN()-

```
SELECT t.team_name, r.project_name, r.score FROM teams t  
INNER JOIN reviews r ON t.team_id = r.team_id;
```

-LEFT JOIN()-

```
SELECT t.team_name, r.project_name, r.score FROM teams t  
LEFT JOIN reviews r ON t.team_id = r.team_id;
```

-RIGHT JOIN()-

```
SELECT t.team_name, r.project_name, r.score FROM teams t  
RIGHT JOIN reviews r ON t.team_id = r.team_id;
```

-Full outer Join() -(UNION)-

```
SELECT t.team_name, r.project_name, r.score FROM teams t  
LEFT JOIN reviews r ON t.team_id = r.team_id  
UNION SELECT t.team_name, r.project_name, r.score FROM teams t  
RIGHT JOIN reviews r ON t.team_id = r.team_id;
```
