

# SQL Implementation Task

---

## General instructions

1. Implement all the problems using SQL.
2. Every configuration and code written should be pushed on GitHub/Bitbucket (Private Repo).
3. You are not permitted to share the doc with anyone, even with your colleagues.
4. SQL Code Commenting has to be added.

## Code Level Requirement -

1. You need to write the sql query for creating the tables first
2. Then you need to write the solution for fetching the output as per the sample output given

## Acceptance Criterion -

1. Push the code to a repo on **Git** in separate files.
  2. Make sure to have **Visual Representation** drawn on draw.io/miro for the data flows for the Problem statement and the solution.
  - 3.
- 

## Problem Statement 1 -

You are given two tables: Player and points.

1. Intern contains three columns: ID, Name and Runs.

| Column | Type    |
|--------|---------|
| ID     | Integer |
| Name   | String  |
| runs   | Integer |

2. Expertise table contains the following data:

| Expertise | Min_Runs | Max_Runs |
|-----------|----------|----------|
| 1         | 0        | 9        |
| 2         | 10       | 19       |
| 3         | 20       | 29       |
| 4         | 30       | 39       |
| 5         | 40       | 49       |
| 6         | 50       | 59       |
| 7         | 60       | 69       |
| 8         | 70       | 79       |
| 9         | 80       | 89       |
| 10        | 90       | 100      |

HR has to organize a cricket match in the organization and wants to generate a report for all the top scoring players for finding out what is the expertise of each player in the game and would create team captains based on this report. To find expertise she organizes a test match and records the top runs scoring players.

The report contains three columns: Name, Expertise and Runs.

She has the following plan on her mind for generating the report -

1. She doesn't want the NAMES of those players who have expertise lower than 8.
2. The report must be in descending order by Expertise -- i.e. higher Expertise are entered first.
3. If there is more than one player with the same Expertise (8-10) assigned to them, order those particular players by their name alphabetically.
4. Finally, if the Expertise is lower than 8, use "NULL" as their name and list them by their Expertise in descending order.
5. If there is more than one player with the same Expertise (1-7) assigned to them, order those particular players by their runs in ascending order.

Write a query to help HR generate such a report .

## Recorded Runs -

| ID | Name    | Runs |
|----|---------|------|
| 1  | Rahul   | 88   |
| 2  | Nitin   | 99   |
| 3  | Ashish  | 78   |
| 4  | Pradeep | 63   |
| 5  | Veeru   | 81   |
| 6  | Saloni  | 68   |

Consider the following table with the Expertise assigned to the Players:

| ID | Name    | Runs | EXPERTISE |
|----|---------|------|-----------|
| 1  | Rahul   | 88   | 9         |
| 2  | Nitin   | 99   | 10        |
| 3  | Ashish  | 78   | 8         |
| 4  | Pradeep | 63   | 7         |
| 5  | Veeru   | 81   | 9         |
| 6  | Saloni  | 68   | 7         |

Sample Output -

| NAME   | RUNS | EXPERTISE |
|--------|------|-----------|
| Nitin  | 99   | 10        |
| Rahul  | 88   | 9         |
| Veeru  | 81   | 9         |
| Ashish | 78   | 8         |
| Null   | 63   | 7         |
| Null   | 68   | 7         |

## Problem 2 -

HR is organizing interviews with many candidates from different colleges using coding challenges and contests.

1. Write a query to print the contest\_id, interviewer\_id, name, and the sums of total\_submissions, total\_accepted\_submissions, total\_views, and total\_unique\_views for each contest sorted by contest\_id. Exclude the contest from the result if all four sums are 0.

Note: A specific contest can be used to screen candidates at more than one college, but each college only holds one screening contest.

---

### Input Format

The following tables hold interview data:

- Contests: The contest\_id is the id of the contest, interviewer\_id is the id of the interviewer who created the contest, and name is the name of the candidate.

| Column         | Type    |
|----------------|---------|
| contest_id     | Integer |
| interviewer_id | Integer |
| name           | String  |

- Colleges: The college\_id is the id of the college, and contest\_id is the id of the contest that the HR used to screen the candidates.

| Column     | Type    |
|------------|---------|
| college_id | Integer |
| contest_id | Integer |

- Challenges: The challenge\_id is the id of the challenge that belongs to one of the contests whose contest\_id HR forgot, and college\_id is the id of the college where the challenge was given to candidates.

| Column        | Type    |
|---------------|---------|
| chanllenge_id | Integer |
| college_id    | Integer |

- View\_Stats: The challenge\_id is the id of the challenge, total\_views is the number of times the challenge was viewed by candidates, and total\_unique\_views is the number of times the challenge was viewed by unique candidates.

| Column             | Type    |
|--------------------|---------|
| chanllenge_id      | Integer |
| total_views        | Integer |
| total_unique_views | Integer |

- **Submission\_Stats:** The `challenge_id` is the id of the challenge, `total_submissions` is the number of submissions for the challenge, and `total_accepted_submission` is the number of submissions that achieved full scores.

| Column                                  | Type    |
|---|---------|
| <code>chanllenge_id</code>              | Integer |
| <code>total_submissions</code>          | Integer |
| <code>total_accepted_submissions</code> | Integer |

---

## Sample Input

Contests Table:

| contest_id | interviewer_id | name    |
|------------|----------------|---------|
| contest-1  | xs-1           | Nitin   |
| contest-2  | xs-2           | Riya    |
| contest-3  | xs-3           | Chandan |

Colleges Table:

| college_id | contest_id |
|------------|------------|
| col-1      | contest-1  |
| col-2      | contest-2  |
| col-3      | contest-3  |

Challenges Table:

| challenge_id | contest_id |
|--------------|------------|
| challenge-1  | contest-1  |
| challenge-2  | contest-1  |
| challenge-3  | contest-2  |
| challenge-4  | contest-3  |

View\_Stats Table:

| challenge_id | total_views | total_unique_views |
|--------------|-------------|--------------------|
| challenge-2  | 26          | 19                 |
| challenge-2  | 15          | 14                 |
| challenge-1  | 43          | 10                 |
| challenge-1  | 72          | 13                 |
| challenge-4  | 35          | 17                 |
| challenge-3  | 11          | 10                 |
| challenge-4  | 41          | 15                 |
| challenge-4  | 75          | 11                 |



Submission\_Stats Table:

| challenge_id | total_submissions | total_unique_views |
|--------------|-------------------|--------------------|
| challenge-4  | 34                | 12                 |
| challenge-2  | 27                | 10                 |
| challenge-2  | 56                | 18                 |
| challenge-4  | 74                | 12                 |
| challenge-4  | 83                | 8                  |
| challenge-4  | 68                | 24                 |
| challenge-4  | 82                | 14                 |
| challenge-2  | 28                | 11                 |

## Sample Output

| contest-id | interviewer_id | name    | total_submissions | total_accepted_submissions | total_views | total_unique_views |
|------------|----------------|---------|-------------------|----------------------------|-------------|--------------------|
| contest-1  | xs-1           | Nitin   | 111               | 39                         | 156         | 56                 |
| contest-2  | xs-2           | Riya    | 0                 | 0                          | 11          | 10                 |
| contest-3  | xs-3           | Chandan | 150               | 38                         | 41          | 15                 |