SQL Problem Statement

Mentorness Internship Program

Project Name: Decode Gaming Behavior

In this internship, you will be working with a dataset related to a game. The dataset includes two tables: `Player Details` and `Level Details`. Below is a brief description of the dataset and the tasks you need to perform:

Dataset Description:

Player Details Table:

- `P_ID`: Player ID
- `PName`: Player Name
- `L1_status`: Level 1 Status
- `L2_status`: Level 2 Status
- `L1_code`: Systemgenerated Level 1 Code
- `L2_code`: Systemgenerated Level 2 Code

Level Details Table:

- `P_ID`: Player ID
- `Dev_ID`: Device ID
- `start_time`: Start Time
- `stages_crossed`: Stages Crossed
- `level`: Game Level
- `difficulty`: Difficulty Level
- `kill_count`: Kill Count
- `headshots_count`: Headshots Count
- `score`: Player Score
- `lives_earned`: Extra Lives Earned

What you have to do?

Use the “Game Analysis.sql” file. Below are 15 questions for which you have to find the answers by writing SQL queries. Each question carries 2 marks.

1. Extract `P_ID`, `Dev_ID`, `PName`, and `Difficulty_level` of all players at Level 0.
2. Find `Level1_code`wise average `Kill_Count` where `lives_earned` is 2, and at least 3 stages are crossed.

3. Find the total number of stages crossed at each difficulty level for Level 2 with players using `zm_series` devices. Arrange the result in decreasing order of the total number of stages crossed.
4. Extract `P_ID` and the total number of unique dates for those players who have played games on multiple days.
5. Find `P_ID` and levelwise sum of `kill_counts` where `kill_count` is greater than the average kill count for Medium difficulty.
6. Find `Level` and its corresponding `Level_code`wise sum of lives earned, excluding Level 0. Arrange in ascending order of level.
7. Find the top 3 scores based on each `Dev_ID` and rank them in increasing order using `Row_Number`. Display the difficulty as well.
8. Find the `first_login` datetime for each device ID.
9. Find the top 5 scores based on each difficulty level and rank them in increasing order using `Rank`. Display `Dev_ID` as well.
10. Find the device ID that is first logged in (based on `start_datetime`) for each player (`P_ID`). Output should contain player ID, device ID, and first login datetime.
11. For each player and date, determine how many `kill_counts` were played by the player so far.
    a) Using window functions
    b) Without window functions
12. Find the cumulative sum of stages crossed over `start_datetime` for each `P_ID`, excluding the most recent `start_datetime`.
13. Extract the top 3 highest sums of scores for each `Dev_ID` and the corresponding `P_ID`.
14. Find players who scored more than 50% of the average score, scored by the sum of scores for each `P_ID`.
15. Create a stored procedure to find the top `n` `headshots_count` based on each `Dev_ID` and rank them in increasing order using `Row_Number`. Display the difficulty as well.