**NAME: SAMSON MUREU**

**PROJECT NAME**

**Game-Analysis-using-SQL**

**Table of contents**

* Project Name
* Dataset Description
* What to Do

**Project Name: Decode Gaming Behavior**

I will be working with a dataset related to a game. The dataset includes two tables:

1. Player Details
2. Level Details.

**Below is a brief description of the dataset and the tasks to perform:**

**Dataset Description:**

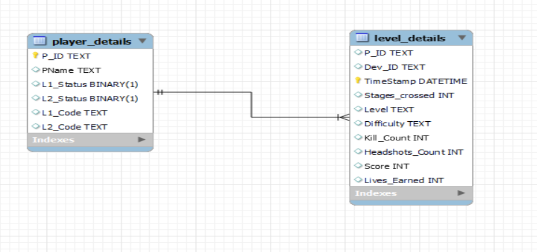
**Player Details Table:**

* 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:**

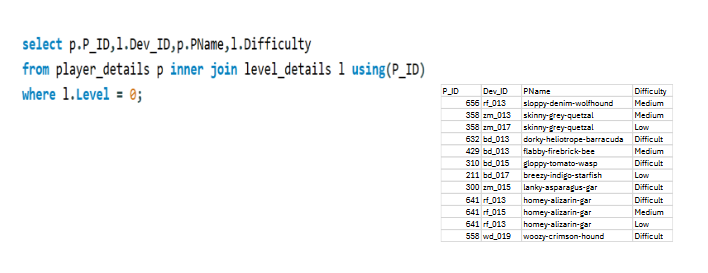
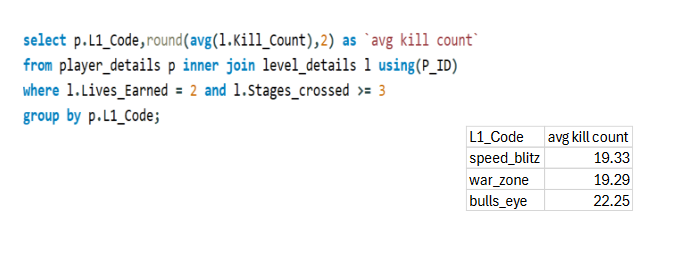
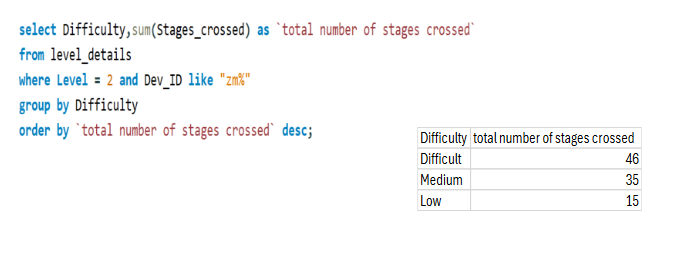
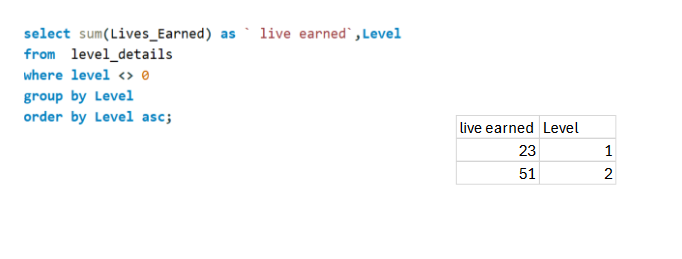
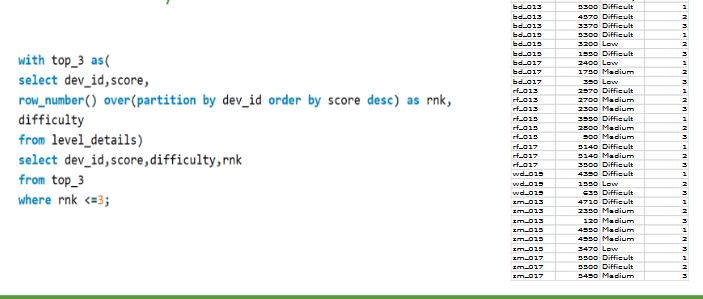
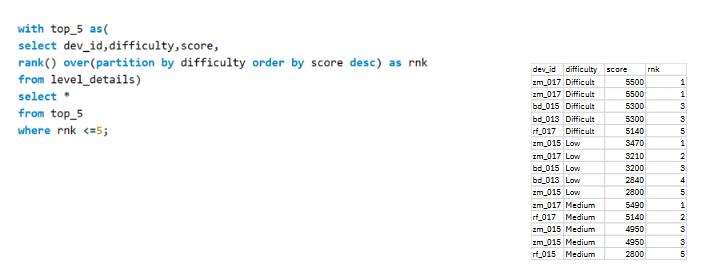
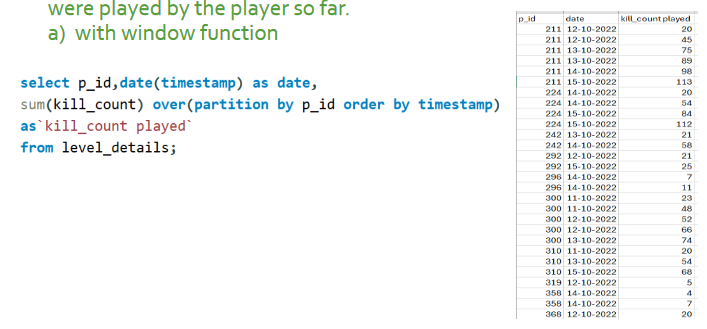
* P\_ID: Player ID
* Dev\_ID: Device ID
* start\_time: Start Time
* stages\_crossed: Stages Crossed
* level: Game Level
* difficulty: Difficulty Level
* kill\_count: Kill Count
* headshots\_count: Headshots Count
* score: Player Score
* lives\_earned: Extra Lives Earned

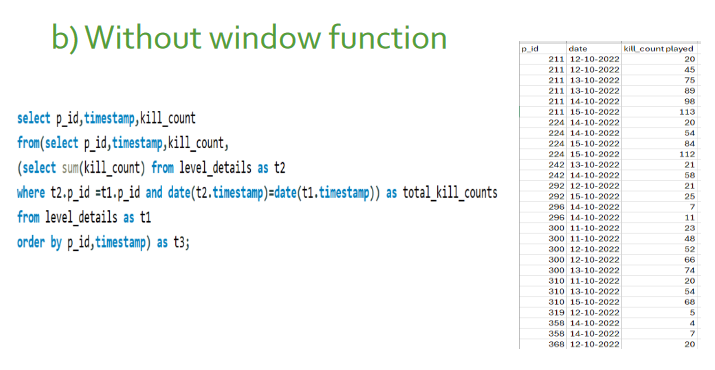
**Entity relationship**

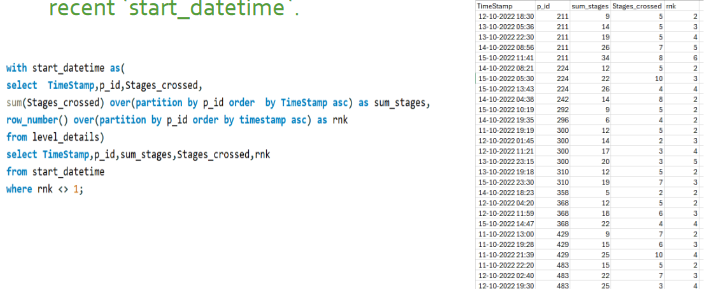
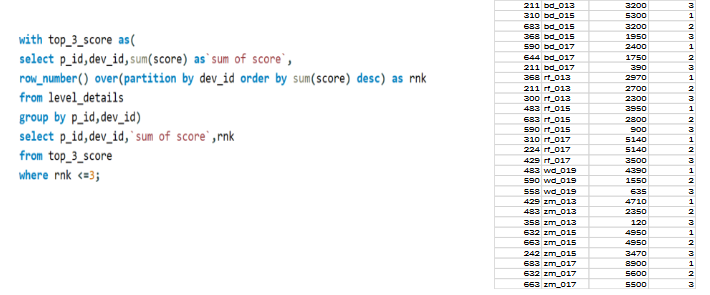
[](https://private-user-images.githubusercontent.com/84933401/322310310-8a04d24d-db2f-4afa-977f-ebfd5e616a2e.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..LUUY_r2iLunWJujzFSA4SEttRFUUcJCExXG2XyE0yOU)

**What to do?**

Use the “Game Analysis.sql” file. to answer 15 questions by writing SQL queries.

1. Extract P\_ID, Dev\_ID, PName, and Difficulty\_level of all players at Level 0.[](https://private-user-images.githubusercontent.com/84933401/322310444-972cbf7e-fb5b-4351-af76-db4631d6ff1e.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..FgZ9gn-_DqYLMaVfdH-skG8z7CynAkRkSNZcf9GZ9jI)
2. Find Level1\_codewise average Kill\_Count where lives\_earned is 2, and at least 3 stages are crossed.[](https://private-user-images.githubusercontent.com/84933401/322310461-8e746c02-f05b-4656-994b-ccf55a201744.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..OuhPoB935Hlyi05BnxGs9N7hmgOSNl3SiKqTRuFt3Qc)
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.[](https://private-user-images.githubusercontent.com/84933401/322310952-c0c84663-c465-4bfb-bf70-0e2bbaff06ff.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..JIGbFWU33ALs6j1Y_c9U0V8eY21keGciZzguBGBgb6c)
4. Extract P\_ID and the total number of unique dates for those players who have played games on multiple days.[](https://private-user-images.githubusercontent.com/84933401/322310978-4c878388-2eb1-4fff-9a7d-6ba97c57841a.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..Gq9Fg0UhwDeZVn4JZiJbgH-PmQnIW8A7cKST7ICcUHM)
5. Find P\_ID and levelwise sum of kill\_counts where kill\_count is greater than the average kill count for Medium difficulty.[](https://private-user-images.githubusercontent.com/84933401/322310991-e69809c8-395e-42fd-9f28-aeead331e8fd.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..w2N_PtjSB9HLACz4yNLMUHoehGZz9Tdm37X2UbXYykg)
6. Find Level and its corresponding Level\_codewise sum of lives earned, excluding Level '0' Arrange in ascending order of level.[](https://private-user-images.githubusercontent.com/84933401/322311577-1f767fc5-41ae-4eaa-b789-f490e0702105.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..mzIdx_gImqK3rcwPmASAyiOtC3HwqTc89GgXRaO3oHY)
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.[](https://private-user-images.githubusercontent.com/84933401/322311582-85d75483-66f2-415a-9280-bf0732200f5f.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..lZysfDcGv-gRtbzNzLjO-ozhuz_8x5tlPdoySu4lRCI)
8. Find the first\_login datetime for each device ID.[](https://private-user-images.githubusercontent.com/84933401/322311587-3e4570e7-ab17-4491-a313-fdb8cfd03d0d.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..SqFivqsfLninxa7b_ZsNtXxRyV2YsseTV5mFiX46CRw)
9. Find the top 5 scores based on each difficulty level and rank them in increasing order using Rank. Display Dev\_ID as well.[](https://private-user-images.githubusercontent.com/84933401/322311592-826b67da-5bf2-4f48-a0f9-ad7db4cf6750.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..zLgviL1oHcHnNou4nNLY9m1FaNmwWmUbnTyeHBxeEJY)
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.[](https://private-user-images.githubusercontent.com/84933401/322311602-eeb28e70-12d1-433c-b6b3-cafe11b7b402.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..YjA07YgW2fUu-jN9WMjZQ-TUSTbpJ6hwod74-RutErc)
11. For each player and date, determine how many kill\_counts were played by the player so far. a). Using window functions[](https://private-user-images.githubusercontent.com/84933401/322311606-b122025a-bd47-4c6c-b906-f20dd43907af.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..akIn_WQgNNwpQSt_r1e2WBpIE58U7J1fjg2uG_mufaM)

b). Without window functions[](https://private-user-images.githubusercontent.com/84933401/322311615-fb4e5568-6209-4298-bc0a-57003452917b.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..W5GZhhL0tRbPzaM2Da88hbowpoi9pj1agk_0em6Psvc)

1. Find the cumulative sum of stages crossed over start\_datetime for each P\_ID, excluding the most recent start\_datetime.[](https://private-user-images.githubusercontent.com/84933401/322311620-518f1acb-11cc-47ba-8c2b-dee43020a526.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..llhK70oRhsRSVoLzq4AGQfdskI8hbRuFI5jj5roJhgE)
2. Extract the top 3 highest sums of scores for each Dev\_ID and the corresponding P\_ID.[](https://private-user-images.githubusercontent.com/84933401/322311623-d98152f3-d052-4521-87b3-b9d836dd5796.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..kh5M_8dYC5Avnl45RPrbpbTeTbxcnbAE4x_boj1Lobk)
3. Find players who scored more than 50% of the average score, scored by the sum of scores for each P\_ID.[](https://private-user-images.githubusercontent.com/84933401/322311629-75a48e82-6c92-4076-9ae3-32078850ae0f.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..8u6zdNrUeTBt9RUUaDanCmE0Tf3hwDOpRcbyc_q03AM)
4. 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. [](https://private-user-images.githubusercontent.com/84933401/322311648-c8379766-9932-43a8-8efc-810697e2c7b1.PNG?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..1LxYlGBGQODOG5SVgqph3_sahNI_bOL0ei3tNra34Kk)