

GAME ANALYSIS WITH SQL

K.SANDHYA



INTRODUCTION TO SQL

SQL (Structured Query language) is a programming Language used for managing and manipulating relational databases. It allows you to perform tasks such as querying data, updating records, and creating or modifying database structures.



Game analysis is the process of examining various aspects of a game, such as its mechanics, dynamics, aesthetics, narrative, and player experience, to gain insights into how the game works, what makes it enjoyable or engaging, and how it can be improved. It often involves dissecting gameplay elements, studying player behavior, and evaluating the overall design and effectiveness of the game.

DATASET

PLAYER DETAILS

- **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

- **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

Some Queries are Here

```
# 4.Extract 'P_ID' and the total number of unique dates for those players who have played games on multiple days.---
SELECT P_ID, COUNT(DISTINCT date (start_datetime)) AS unique_dates
FROM ld
GROUP BY P_ID
HAVING COUNT(DISTINCT date (start_datetime)) > 1;
```

```
# 5. Find 'P_ID' and levelwise sum of 'kill_counts' where 'kill_count' is greater than the average kill count for Medium difficulty.---
SELECT P_ID, level, SUM(kill_count) AS total_kills
FROM ld
WHERE difficulty = 'Medium'
GROUP BY P_ID, level
HAVING SUM(kill_count) > (SELECT AVG(kill_count) FROM ld WHERE difficulty = 'Medium');
```

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...--

```
WITH Top3 AS (
    SELECT *,
        ROW_NUMBER() OVER (PARTITION BY Dev_ID ORDER BY score DESC) AS ranks
    FROM ld
)
SELECT Dev_ID, difficulty, score, ranks
FROM Top3
WHERE ranks <= 3
ORDER BY Dev_ID, ranks;
```

CONCLUSION

GAME ANALYSIS IS LIKE TAKING A MAGNIFYING
GLASS TO A GAME TO SEE WHAT MAKES IT
AWESOME AND HOW IT COULD BE EVEN
BETTER

THANK YOU