



# **“PLAYER BEHAVIOR AND GAME PERFORMANCE ANALYSIS”**

## **USING SQL**

**AUTHOR : VIVEK KUMAR VERMA**

## AIM OF THE PROJECT

"The project aims to analyze game data using SQL to gain insights into player behavior, game performance, and user engagement. The analysis focuses on player activity patterns, performance metrics, and trends to provide actionable insights for improving game design and strategy."

# CONTENT

- Project overview
- Description of Data set
- Data Exploration & Analysis
- Insights & Conclusion

## PROJECT OVERVIEW

The dataset includes two tables: **`Player Details`** and **`Level Details`**. This project leverages SQL to analyze game data, aiming to uncover insights into player detail, game performance, and user engagement.

# DESCRIPTION OF EACH COLUMN IN DATASET

## Player Details Table:

- `P\_ID`: Player ID
- `P\_Name`: Player Name
- `L1\_status`: Level 1 Status
- `L2\_status`: Level 2 Status
- `L1\_code`: System generated Level 1 Code
- `L2\_code`: System generated Level 2 Code

# SECOND TABLE :

## 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

# DATA SET DESCRIPTION

PLAYER DETAIL TABLE

P_ID	PName	L1_Status	L2_status	L1_Code	L2_Code
310	gloppy-tomato-wasp	1	1	war_zone	splippery_slope
319	chummy-flax-crab	1	0	speed_blitz	
320	chewy-harlequin-gharial	0	0		
358	skinny-grey-quetzal	0	0		
368	homely-vermillion-toad	1	1	war_zone	resurgence
376	pretty-champagne-spaniel	0	0		
422	pasty-silver-raccoon	1	1	bulls_eye	splippery_slope
428	leaky-magnolia-iguana	1	0	leap_of_f...	
429	flabby-firebrick-bee	1	1	speed_blitz	cosmic_vision
430	messy-wisteria-termite	1	1	leap_of_f...	resurgence
441	woozy-magenta-birman	1	0	speed_blitz	
463	messy-magnolia-woodpecker	1	0	war_zone	
483	tasty-peach-fly	1	1	bulls_eye	cosmic_vision
547	scanty-beige-ray	1	0	bulls_eye	
558	woozy-crimson-hound	0	0		
590	stealthy-xanthic-cattle	1	1	war_zone	splippery_slope
603	smelly-linen-leopard	1	1	war_zone	splippery_slope
632	dorky-heliotrope-barracuda	1	1	speed_blitz	splippery_slope

## LEVEL DETAIL TABLE

P_ID	Dev_Id	start_datetime	Stages_crossed	Level	Difficulty	Kill_Count	Headshots_Count	Score	Lives_Earned
211	bd_013	2022-10-12 18:30:30	5	1	Difficult	25	15	3200	2
211	bd_017	2022-10-12 13:23:45	4	0	Low	20	15	390	2
211	rf_013	2022-10-13 05:36:15	5	1	Medium	30	11	2700	1
211	rf_017	2022-10-15 11:41:19	8	2	Difficult	15	11	1100	1
211	zm_015	2022-10-13 22:30:18	5	2	Low	14	8	2800	0
211	zm_017	2022-10-14 08:56:24	7	2	Medium	9	3	750	2
224	bd_013	2022-10-15 05:30:28	10	2	Difficult	30	22	5300	4
224	bd_013	2022-10-15 13:43:50	4	2	Difficult	28	25	4570	2
224	bd_015	2022-10-14 08:21:49	5	1	Difficult	34	30	1300	0
224	rf_017	2022-10-14 01:15:56	7	1	Medium	20	18	5140	0
242	bd_013	2022-10-13 01:14:29	6	1	Low	21	17	2840	0
242	zm_015	2022-10-14 04:38:50	8	1	Low	37	26	3470	2
292	rf_013	2022-10-12 04:29:45	4	1	Low	21	17	1890	0
292	rf_015	2022-10-15 10:19:30	5	1	Medium	4	3	670	0
296	zm_015	2022-10-14 19:35:49	4	1	Medium	4	0	100	0
296	zm_017	2022-10-14 15:15:15	2	1	Difficult	7	3	1040	0
300	bd_013	2022-10-11 19:19:19	5	1	Low	25	11	540	0
300	bd_013	2022-10-13 23:15:42	3	2	Medium	8	4	590	1
300	rf_013	2022-10-11 05:20:40	7	1	Medium	23	17	2300	0



# PROBLEM STATEMENT

## Game Analysis dataset

- Players play a game divided into 3-levels (L0,L1 and L2)
- Each level has 3 difficulty levels ( Low, Medium, High )
- At each level, players have to kill the opponents using guns/physical fight.
- Each level has multiple stages at each difficulty level.
- A player can only play L1 using its system generated L1\_code.
- Only players who have played Level1 can possibly play Level2 using its system generated L2\_code.
- By default a player can play L0.
- Each player can login to the game using a Dev\_ID.
- Players can earn extra lives at each stage in a level.

## DATA EXPLORATION & ANALYSIS

```
alter table level_details.level_details2 drop myunknowncolumn;  
alter table level_details.level_details2 change timestamp start_datetime datetime;  
alter table level_details.level_details2 modify Dev_Id varchar(10);  
alter table level_details.level_details2 modify Difficulty varchar(15);  
alter table level_details.level_details2 add primary key(P_ID,Dev_id,start_datetime);
```

# Q1) EXTRACT `P\_ID`, `DEV\_ID`, `PNAME` AND `DIFFICULTY\_LEVEL` OF ALL PLAYERS.

```
select pd.P_ID,pd.PNAME,ld.Dev_ID,ld.Difficulty from player_details.player_details as pd
inner join level_details.level_details2 as ld
on pd.P_ID=ld.P_ID;
```

P_ID	PNAME	Dev_ID	Difficulty
211	breezy-indigo-starfish	bd_013	Difficult
211	breezy-indigo-starfish	bd_017	Low
211	breezy-indigo-starfish	rf_013	Medium
211	breezy-indigo-starfish	rf_017	Difficult
211	breezy-indigo-starfish	zm_015	Low
211	breezy-indigo-starfish	zm_017	Medium
224	nippy-peach-neanderthal	bd_013	Difficult
224	nippy-peach-neanderthal	bd_013	Difficult
224	nippy-peach-neanderthal	bd_015	Difficult
224	nippy-peach-neanderthal	rf_017	Medium
242	slaphappy-cinnamon-sq...	bd_013	Low
242	slaphappy-cinnamon-sq...	zm_015	Low
292	ugly-goldenrod-numbat	rf_013	Low
292	ugly-goldenrod-numbat	rf_015	Medium
296	silly-taupe-ray	zm_015	Medium
296	silly-taupe-ray	zm_017	Difficult
300	lanky-asparagus-gar	bd_013	Low
300	lanky-asparagus-gar	bd_013	Medium
300	lanky-asparaqus-qar	rf_013	Medium

❏ **Insight:** This query provides a foundational understanding of player distribution and the difficulty levels they have played at. It could reveal if most players tend to play at a particular difficulty level, which might suggest balancing issues or player preference.

## Q2) TOTAL NUMBER OF STAGES CROSSED AT EACH DIFFICULTY LEVEL FOR LEVEL 2 WITH PLAYERS.

```
select count(Stages_crossed) AS total_stages_crossed, Difficulty from level_details.level_details2
where Level=2
group by Difficulty;
```

	total_stages_crossed	Difficulty
▶	16	Difficult
	3	Low
	10	Medium

❑ **Insight:** This result could help understand player progression challenges. If one difficulty level has significantly fewer stages crossed, it could imply that the level is too difficult or too easy, impacting player retention.

### Q3) `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_count
from level_details.level_details2
group by P_ID
having unique_dates_count>1;
```

	P_ID	unique_dates_count
▶	211	4
	224	2
	242	2
	292	2
	300	3
	310	3
	368	2
	483	3
	590	3
	632	3
	641	2

- ❑ **Insight:** This query identifies the most engaged players. The more unique dates a player logs in, the higher the engagement, which is crucial for understanding player retention and loyalty.

## Q4) `P\_ID` AND LEVEL WISE 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_kill_count from level_details.level_details2
where Kill_Count > (select avg(Kill_Count) from level_details.level_details2
                    where Difficulty='Medium')
group by Level,P_ID;
```

	P_ID	Level	total_kill_count
▶	211	1	55
	211	0	20
	224	2	58
	224	1	54
	242	1	58
	292	1	21
	300	1	48
	310	0	34
	310	1	20
	368	2	24
	368	1	20
	429	1	30
	429	2	55
	483	1	40

❑ **Insight:** Identifies high-performing players at the medium difficulty level. This can help tailor rewards or challenges for these players and balance the difficulty for future game updates.

## Q5) `LEVEL` AND ITS CORRESPONDING `LEVEL\_CODE` WISE SUM OF LIVES EARNED, EXCLUDING LEVEL 0. ARRANGING IN ASCENDING ORDER OF LEVEL.

```
select ld.Level,pd.L1_Code,pd.L2_Code,sum(ld.Lives_Earned) as total_lives_earned
from player_details.player_details pd
join level_details.level_details2 ld
on ld.P_ID=pd.P_ID
where ld.Level<>0
group by pd.L1_Code,pd.L2_Code,Level
order by Level asc;
```

	Level	L1_Code	L2_Code	total_lives_earned
▶	1	war_zone	slippery_slope	7
	1	bullseye		3
	1	war_zone		4
	1	speed_blitz	cosmic_vision	4
	1	speed_blitz		0
	1	war_zone	resurgence	0
	1	leap_of_faith		0
	1	bullseye	cosmic_vision	1
	1	speed_blitz	slippery_slope	3
	1	bullseye	resurgence	1
	2	war_zone	slippery_slope	14
	2	speed_blitz	cosmic_vision	6
	2	war_zone	resurgence	3

❑ **Insight:** Understanding where players earn extra lives provides insights into level difficulty. Levels where players earn significantly more lives might indicate areas where the game is too challenging or too easy.

## Q6) TOP 3 SCORES BASED ON EACH `DEV\_ID` AND RANK THEM IN INCREASING ORDER USING `ROW\_NUMBER`. SHOWING THE DIFFICULTY AS WELL.

```
with my_cte as (  
    select  
        Score,Dev_Id,Difficulty,ROW_NUMBER() over (partition by Dev_Id order by Score DESC) AS rank_number  
    from  
        level_details.level_details2  
    )  
select Score,Dev_Id,Difficulty,rank_number from my_cte  
where rank_number <=3  
order by Dev_Id,rank_number;
```

	Score	Dev_Id	Difficulty	rank_number
▶	5300	bd_013	Difficult	1
	4570	bd_013	Difficult	2
	3370	bd_013	Difficult	3
	5300	bd_015	Difficult	1
	3200	bd_015	Low	2
	1950	bd_015	Difficult	3
	2400	bd_017	Low	1
	1750	bd_017	Medium	2
	390	bd_017	Low	3
	2970	rf_013	Difficult	1
	2700	rf_013	Medium	2
	2300	rf_013	Medium	3
	3950	rf_015	Difficult	1

- ❑ **Insight:** This ranking provides insight into the best-performing players per device. If one device consistently has top scores, there could be device-specific optimizations or advantages that need balancing.



## Q7) 'FIRST\_LOGIN' DATETIME FOR EACH DEVICE ID.

```
SELECT
    Dev_ID,
    MIN(start_datetime) AS first_login_datetime
FROM
    level_details.level_details2
GROUP BY
    Dev_ID;
```

	Dev_ID	first_login_datetime
▶	bd_013	2022-10-11 02:23:45
	bd_017	2022-10-12 07:30:18
	rf_013	2022-10-11 05:20:40
	rf_017	2022-10-11 09:28:56
	zm_015	2022-10-11 14:05:08
	zm_017	2022-10-11 14:33:27
	bd_015	2022-10-11 18:45:55
	rf_015	2022-10-11 19:34:25
	zm_013	2022-10-11 13:00:22
	wd_019	2022-10-12 23:19:17

- ❑ **Insight:** This can help analyze the initial player engagement and onboarding experience. Early patterns can indicate how effective the game's tutorial or first-time user experience is.

## Q8) TOP 5 SCORES BASED ON EACH DIFFICULTY LEVEL AND RANK THEM IN INCREASING ORDER USING `RANK`. SHOWING `DEV\_ID` AS WELL.

```
with my_cte as (  
    select Score,Dev_Id,Difficulty,  
    RANK() over (partition by Difficulty order by Score DESC) as rank_number  
    from level_details.level_details2  
    )  
select Dev_Id,Difficulty,Score,rank_number from my_cte  
where rank_number <=5  
order by rank_number,Difficulty;
```

	Dev_Id	Difficulty	Score	rank_number
▶	zm_017	Difficult	5500	1
	zm_017	Difficult	5500	1
	zm_015	Low	3470	1
	zm_017	Medium	5490	1
	zm_017	Low	3210	2
	rf_017	Medium	5140	2
	bd_013	Difficult	5300	3
	bd_015	Difficult	5300	3
	bd_015	Low	3200	3
	zm_015	Medium	4950	3
	zm_015	Medium	4950	3
	bd_013	Low	2840	4
	rf_017	Difficult	5140	5
	zm_015	Low	2800	5

❏ **Insight:** This helps compare player performance across different difficulty levels, revealing which levels might need balancing.

**Q9) DEVICE ID THAT IS FIRST LOGGED IN (BASED ON `START\_DATETIME`) FOR EACH PLAYER (`P\_ID`). OUTPUT CONTAIN PLAYER ID, DEVICE ID, AND FIRST LOGIN DATETIME.**

```
select Dev_Id,P_ID,min(start_datetime) as first_logged from level_details.level_details2
group by P_ID,Dev_Id,start_datetime
order by start_datetime asc;
```

	Dev_Id	P_ID	first_logged
▶	bd_013	683	2022-10-11 02:23:45
	rf_013	300	2022-10-11 05:20:40
	rf_017	429	2022-10-11 09:28:56
	zm_013	429	2022-10-11 13:00:22
	zm_015	644	2022-10-11 14:05:08
	zm_017	483	2022-10-11 14:33:27
	rf_017	310	2022-10-11 15:15:15
	bd_013	656	2022-10-11 17:47:09
	bd_015	683	2022-10-11 18:45:55
	bd_013	300	2022-10-11 19:19:19
	bd_013	429	2022-10-11 19:28:43
	rf_015	644	2022-10-11 19:34:25
	zm_017	429	2022-10-11 21:39:00
	bd_015	483	2022-10-11 22:20:10
	zm_015	368	2022-10-12 01:14:34
	zm_015	300	2022-10-12 01:45:17
	rf_015	483	2022-10-12 02:40:20

❑ **Insight:** Identifies the primary device used by each player, which could be useful for targeting device-specific updates or features.

## Q10) CUMULATIVE SUM OF STAGES CROSSED OVER `START\_DATETIME` FOR EACH `P\_ID`, EXCLUDING THE MOST RECENT `START\_DATETIME`.

```
with cte_ranked as (  
    select P_ID, start_datetime, Stages_crossed,  
           Rank() over (partition by P_ID order by start_datetime Desc) as rank_number  
    from level_details.level_details2  
)  
select P_ID, start_datetime,  
       sum(Stages_crossed) over (partition by P_ID order by start_datetime ) as cumulative_sum_of_stages_crossed  
from cte_ranked  
where rank_number > 1  
order by  
P_ID, start_datetime;
```

	P_ID	start_datetime	cummulative_sum_of_stages_crossed
▶	211	2022-10-12 13:23:45	4
	211	2022-10-12 18:30:30	9
	211	2022-10-13 05:36:15	14
	211	2022-10-13 22:30:18	19
	211	2022-10-14 08:56:24	26
	224	2022-10-14 01:15:56	7

❑ **Insight:** This shows player progress over time. If most players show a plateau in progression, it might indicate a need for more engaging content

# Q11)EXTRACT THE TOP 3 HIGHEST SUMS OF SCORES FOR EACH `DEV\_ID` AND THE CORRESPONDING `P\_ID`

```
with cte_summed as (  
    select P_ID,Dev_Id,sum(Score) as total_score from level_details.level_details2  
    group by P_ID,Dev_Id  
)  
  
ranked_score as (  
    select P_ID,Dev_Id,total_score,  
    Rank() over (partition by Dev_Id order by total_score desc) as rank_number  
    from cte_summed  
)  
  
select Dev_Id,P_ID,total_score  
from ranked_score  
where rank_number <=3  
order by  
    Dev_Id,rank_number;
```

	Dev_Id	P_ID	total_score
▶	bd_013	224	9870
	bd_013	310	3370
	bd_013	211	3200
	bd_015	310	5300
	bd_015	683	3200
	bd_015	368	1950
	bd_017	590	2400
	bd_017	644	1750
	bd_017	211	390
	rf_013	368	2970
	rf_013	211	2700
	rf_013	300	2300
	rf_015	483	3950
	rf_015	683	2800
	rf_015	590	900
	rf_017	310	5140
	rf_017	224	5140
	rf_017	429	3500
	wd_019	483	4390

- ❑ **Insight:** This identifies the most skilled or persistent players on each device, useful for targeting high-value players with specific in-game rewards or events.

# INSIGHTS AND CONCLUSION :

- **Understanding Player Behavior:** Identified active players, preferred difficulty levels, and popular game stages, offering insights into player preferences and areas needing balance.
- **Evaluating Game Performance:** Analyzed key metrics such as scores, kill counts, and stages crossed to detect performance trends and highlight top-performing players.
- **Optimizing Game Engagement:** Discovered engagement patterns, such as frequent login times and device preferences, enabling targeted improvements in game content and performance.
- **Actionable Recommendations:** Provided data-driven suggestions for enhancing game design, balancing difficulty levels, and developing strategies to boost player retention and satisfaction.
- **Conclusion:** This analysis equips game developers with essential insights to refine gameplay, optimize engagement strategies, and enhance the overall player experience.