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

In [1]:
```
import pandas as pd
import sqlite3
```

In [2]:
```
df_1=pd.read_csv("player_details.csv")
df_1
```

Out[2]:

	Unnamed: 0	P_ID	PName	L1_Status	L2_Status	L1_Code	L2_Code
0	0	656	sloppy-denim-wolfhound	1	0	war_zone	NaN
1	1	358	skinny-grey-quetzal	0	0	NaN	NaN
2	2	296	silly-taupe-ray	1	0	war_zone	NaN
3	3	644	randy-turquoise-scorpion	1	1	speed_blitz	cosmic_vision
4	4	320	chewy-harlequin-gharial	0	0	NaN	NaN
5	5	632	dorky-heliotrope-barracuda	1	1	speed_blitz	splippery_slope
6	6	428	leaky-magnolia-iguana	1	0	leap_of_faith	NaN
7	7	429	flabby-firebrick-bee	1	1	speed_blitz	cosmic_vision
8	8	310	gloppy-tomato-wasp	1	1	war_zone	splippery_slope
9	9	211	breezy-indigo-starfish	1	1	war_zone	splippery_slope
10	10	319	chummy-flax-crab	1	0	speed_blitz	NaN
11	11	547	scanty-beige-ray	1	0	bulls_eye	NaN
12	12	376	pretty-champagne-spaniel	0	0	NaN	NaN
13	13	300	lanky-asparagus-gar	1	1	speed_blitz	cosmic_vision
14	14	224	nippy-peach-neanderthal	1	1	war_zone	splippery_slope
15	15	641	homey-alizarin-gar	0	0	NaN	NaN
16	16	430	messy-wisteria-termite	1	1	leap_of_faith	resurgence
17	17	558	woozy-crimson-hound	0	0	NaN	NaN
18	18	463	messy-magnolia-woodpecker	1	0	war_zone	NaN
19	19	603	smelly-linen-leopard	1	1	war_zone	splippery_slope
20	20	242	slaphappy-cinnamon-squirrel	1	0	bulls_eye	NaN
21	21	292	ugly-goldenrod-numbat	1	0	bulls_eye	NaN
22	22	590	stealthy-xanthic-cattle	1	1	war_zone	splippery_slope
23	23	483	tasty-peach-fly	1	1	bulls_eye	cosmic_vision
24	24	368	homely-vermilion-toad	1	1	war_zone	resurgence
25	25	653	breezy-buff-tarantula	1	0	leap_of_faith	NaN
26	26	441	woozy-magenta-birman	1	0	speed_blitz	NaN
27	27	663	fuzzy-cornflower-whippet	1	1	bulls_eye	resurgence
28	28	422	pasty-silver-raccoon	1	1	bulls_eye	splippery_slope

	Unnamed: 0	P_ID	PName	L1_Status	L2_Status	L1_Code	L2_Code
29	29	683	craggy-ivory-dragonfly	1	1	speed_blitz	splippery_slope

In [3]:
```
df_2=pd.read_csv("level_details2.csv")
df_2.head(10)
```

Out[3]:

	Unnamed: 0	P_ID	Dev_ID	TimeStamp	Stages_crossed	Level	Difficulty	Kill_Count	Headshots_(
0	0	644	zm_015	2022-10-11 14:05:08	3	1	Medium	11	
1	1	644	rf_015	2022-10-11 19:34:25	1	1	Low	7	
2	2	644	bd_017	2022-10-12 23:52:18	6	2	Medium	24	
3	3	656	rf_013	2022-10-15 18:12:50	7	0	Medium	15	
4	4	656	bd_015	2022-10-13 22:19:45	4	1	Low	19	
5	5	656	rf_017	2022-10-14 07:32:00	2	1	Difficult	3	
6	6	656	bd_013	2022-10-11 17:47:09	10	1	Low	18	
7	7	296	zm_017	2022-10-14 15:15:15	2	1	Difficult	7	
8	8	296	zm_015	2022-10-14 19:35:49	4	1	Medium	4	
9	9	632	bd_013	2022-10-12 16:30:30	5	0	Difficult	45	

In [4]:
```
conn = sqlite3.connect('Game_Analysis.db')
```

In [5]:
```
df_1.to_sql('Player_Details',conn,if_exists='replace',index=False)
df_2.to_sql('Level_Details',conn,if_exists='replace',index=False)
```

Out[5]:
```
77
```

In [6]:
```
%load_ext sql
```

In [7]:
```
%sql sqlite:///Game_Analysis.db
```

In [8]:
```
%%sql
SELECT * FROM Player_Details;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[8]:

Unnamed: 0	P_ID	PName	L1_Status	L2_Status	L1_Code	L2_Code
0	656	sloppy-denim-wolfhound	1	0	war_zone	None
1	358	skinny-grey-quetzal	0	0	None	None
2	296	silly-taupe-ray	1	0	war_zone	None
3	644	randy-turquoise-scorpion	1	1	speed_blitz	cosmic_vision
4	320	chewy-harlequin-gharial	0	0	None	None
5	632	dorky-heliotrope-barracuda	1	1	speed_blitz	splippery_slope
6	428	leaky-magnolia-iguana	1	0	leap_of_faith	None
7	429	flabby-firebrick-bee	1	1	speed_blitz	cosmic_vision
8	310	gloppy-tomato-wasp	1	1	war_zone	splippery_slope
9	211	breezy-indigo-starfish	1	1	war_zone	splippery_slope
10	319	chummy-flax-crab	1	0	speed_blitz	None
11	547	scanty-beige-ray	1	0	bulls_eye	None
12	376	pretty-champagne-spaniel	0	0	None	None
13	300	lanky-asparagus-gar	1	1	speed_blitz	cosmic_vision
14	224	nippy-peach-neanderthal	1	1	war_zone	splippery_slope
15	641	homey-alizarin-gar	0	0	None	None
16	430	messy-wisteria-termite	1	1	leap_of_faith	resurgence
17	558	woozy-crimson-hound	0	0	None	None
18	463	messy-magnolia-woodpecker	1	0	war_zone	None
19	603	smelly-linen-leopard	1	1	war_zone	splippery_slope
20	242	slaphappy-cinnamon-squirrel	1	0	bulls_eye	None
21	292	ugly-goldenrod-numbat	1	0	bulls_eye	None
22	590	stealthy-xanthic-cattle	1	1	war_zone	splippery_slope
23	483	tasty-peach-fly	1	1	bulls_eye	cosmic_vision
24	368	homely-vermilion-toad	1	1	war_zone	resurgence
25	653	breezy-buff-tarantula	1	0	leap_of_faith	None
26	441	woozy-magenta-birman	1	0	speed_blitz	None
27	663	fuzzy-cornflower-whippet	1	1	bulls_eye	resurgence
28	422	pasty-silver-raccoon	1	1	bulls_eye	splippery_slope
29	683	craggy-ivory-dragonfly	1	1	speed_blitz	splippery_slope

In [9]:
```
%%sql
SELECT * FROM Level_Details LIMIT 10;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[9]:

Unnamed: 0	P_ID	Dev_ID	TimeStamp	Stages_crossed	Level	Difficulty	Kill_Count	Headshots_Cou
0	644	zm_015	2022-10-11 14:05:08	3	1	Medium	11	
1	644	rf_015	2022-10-11 19:34:25	1	1	Low	7	
2	644	bd_017	2022-10-12 23:52:18	6	2	Medium	24	
3	656	rf_013	2022-10-15 18:12:50	7	0	Medium	15	
4	656	bd_015	2022-10-13 22:19:45	4	1	Low	19	
5	656	rf_017	2022-10-14 07:32:00	2	1	Difficult	3	
6	656	bd_013	2022-10-11 17:47:09	10	1	Low	18	
7	296	zm_017	2022-10-14 15:15:15	2	1	Difficult	7	
8	296	zm_015	2022-10-14 19:35:49	4	1	Medium	4	
9	632	bd_013	2022-10-12	5	0	Difficult	45	

In [10]:
```
%%sql
SELECT COUNT(*) FROM Player_Details;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[10]:

COUNT(*)
30

In [11]:
```
%%sql
SELECT COUNT(*) FROM Level_Details;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[11]:

COUNT(*)
77

1. Extract `P_ID`, `Dev_ID`, `PName`, and `Difficulty_level` of all players at Level 0.

In [12]:
```
%%sql
SELECT pd.P_ID, ld.Dev_ID, pd.PName, ld.difficulty AS Difficulty_level
FROM Player_Details pd
INNER JOIN Level_Details ld ON pd.P_ID = ld.P_ID
WHERE ld.level = 0;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[12]:

P_ID	Dev_ID	PName	Difficulty_level
656	rf_013	sloppy-denim-wolfhound	Medium
632	bd_013	dorky-heliotrope-barracuda	Difficult
429	bd_013	flabby-firebrick-bee	Medium
310	bd_015	gloppy-tomato-wasp	Difficult
211	bd_017	breezy-indigo-starfish	Low
300	zm_015	lanky-asparagus-gar	Difficult
358	zm_017	skinny-grey-quetzal	Low
358	zm_013	skinny-grey-quetzal	Medium
641	rf_013	homey-alizarin-gar	Low
641	rf_015	homey-alizarin-gar	Medium
641	rf_013	homey-alizarin-gar	Difficult
558	wd_019	woozy-crimson-hound	Difficult

2. Find `Level1_code` wise average `Kill_Count` where `lives_earned` is 2, and at least 3 stages are crossed.

In [13]:

```
%%sql
SELECT L1_code, AVG(kill_count) AS avg_kill_count
FROM Player_Details pd
INNER JOIN Level_Details ld ON pd.P_ID = ld.P_ID
WHERE lives_earned = 2
  AND stages_crossed >= 3
GROUP BY L1_code;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[13]:

L1_Code	avg_kill_count
bulls_eye	22.25
speed_blitz	19.333333333333332
war_zone	19.285714285714285

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.

In [14]:
```
%%sql
SELECT ld.difficulty, SUM(ld.stages_crossed) AS total_stages_crossed
FROM Level_Details ld
JOIN Player_Details pd ON ld.P_ID = pd.P_ID
WHERE ld.level = 2
AND ld.Dev_ID LIKE 'zm_%'
GROUP BY ld.difficulty
ORDER BY total_stages_crossed DESC;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[14]:

Difficulty	total_stages_crossed
Difficult	46
Medium	35
Low	15

4. Extract `P_ID` and the total number of unique dates for those players who have played games on multiple days.

In [15]:
```
%%sql
SELECT ld.P_ID, COUNT(DISTINCT DATE(ld.timestamp)) AS unique_dates_count
FROM Level_Details ld
GROUP BY ld.P_ID
HAVING COUNT(DISTINCT DATE(ld.timestamp)) > 1;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[15]:

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
644	2
656	4
683	4

5. Find `P_ID` and levelwise sum of `kill_counts` where `kill_count` is greater than the average kill count for Medium difficulty.

In [16]:

```
%%sql
SELECT ld.P_ID, ld.level, SUM(ld.kill_count) AS sum_kill_count
FROM Level_Details ld
JOIN (
    SELECT AVG(kill_count) AS avg_kill_count
    FROM Level_Details
    WHERE difficulty = 'Medium'
) AS avg_kill
ON ld.kill_count > avg_kill.avg_kill_count
GROUP BY ld.P_ID, ld.level;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[16]:

P_ID	Level	sum_kill_count
211	0	20
211	1	55
224	1	54
224	2	58
242	1	58
292	1	21
300	1	48
310	0	34
310	1	20
368	1	20
368	2	24
429	1	30
429	2	55
483	1	40
483	2	94
547	1	20
558	0	21
590	1	24
632	0	45
632	1	28
632	2	53
644	2	24
656	1	37
663	1	73
663	2	53
683	1	21
683	2	64

6. Find `Level` and its corresponding `Level_code` wise sum of lives earned, excluding Level 0. Arrange in ascending order of level.

In [17]:
```
%%sql
SELECT ld.level AS Level, MAX(pd.l1_code) AS Level_code, SUM(ld.lives_earned) AS To
FROM Level_Details ld
JOIN Player_Details pd ON ld.P_ID = pd.P_ID
```

```
WHERE ld.level > 0
GROUP BY ld.level
ORDER BY ld.level ASC;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[17]:

Level	Level_code	Total_Lives_Earned
1	war_zone	23
2	war_zone	51

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.

In [18]:

```
%%sql
WITH TopScores AS (
    SELECT ld.Dev_ID, ld.score, ld.difficulty,
           ROW_NUMBER() OVER (PARTITION BY ld.Dev_ID ORDER BY ld.score DESC) AS rar
    FROM Level_Details ld
)
SELECT Dev_ID, score, difficulty
FROM TopScores
WHERE rank <= 3
ORDER BY Dev_ID, rank;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[18]:

Dev_ID	score	difficulty
bd_013	5300	Difficult
bd_013	4570	Difficult
bd_013	3370	Difficult
bd_015	5300	Difficult
bd_015	3200	Low
bd_015	1950	Difficult
bd_017	2400	Low
bd_017	1750	Medium
bd_017	390	Low
rf_013	2970	Difficult
rf_013	2700	Medium
rf_013	2300	Medium
rf_015	3950	Difficult
rf_015	2800	Medium
rf_015	900	Medium
rf_017	5140	Difficult
rf_017	5140	Medium
rf_017	3500	Difficult
wd_019	4390	Difficult
wd_019	1550	Low
wd_019	635	Difficult
zm_013	4710	Difficult
zm_013	2350	Medium
zm_013	120	Medium
zm_015	4950	Medium
zm_015	4950	Medium
zm_015	3470	Low
zm_017	5500	Difficult
zm_017	5500	Difficult
zm_017	5490	Medium

8. Find the `first_login` datetime for each device ID.

In [19]:
```
%%sql
SELECT Dev_ID, MIN(timestamp) AS first_login
```

```
FROM Level_Details
GROUP BY Dev_ID;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[19]:

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

9. Find the top 5 scores based on each difficulty level and rank them in increasing order using `Rank`. Display `Dev_ID` as well.

In [20]:

```
%%sql
WITH RankedScores AS (
    SELECT ld.Dev_ID,
           ld.difficulty,
           ld.score,
           RANK() OVER (PARTITION BY ld.difficulty ORDER BY ld.score DESC) AS Rank
    FROM Level_Details ld
)
SELECT Dev_ID, difficulty, score, Rank
FROM RankedScores
WHERE Rank <= 5
ORDER BY difficulty, Rank;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[20]:

Dev_ID	difficulty	score	Rank
zm_017	Difficult	5500	1
zm_017	Difficult	5500	1
bd_015	Difficult	5300	3
bd_013	Difficult	5300	3
rf_017	Difficult	5140	5
zm_015	Low	3470	1
zm_017	Low	3210	2
bd_015	Low	3200	3
bd_013	Low	2840	4
zm_015	Low	2800	5
zm_017	Medium	5490	1
rf_017	Medium	5140	2
zm_015	Medium	4950	3
zm_015	Medium	4950	3
rf_015	Medium	2800	5

10. Find the top 5 scores based on each difficulty level and rank them in increasing order using `Rank`. Display `Dev_ID` as well.

In [21]:
```
%%sql
WITH RankedScores AS (
    SELECT
        ld.Dev_ID,
        ld.difficulty,
        ld.score,
        RANK() OVER (PARTITION BY ld.difficulty ORDER BY ld.score DESC) AS score_ra
    FROM Level_Details ld
)
SELECT
    rs.Dev_ID,
    rs.difficulty,
    rs.score,
    rs.score_rank
FROM RankedScores rs
WHERE rs.score_rank <= 5
ORDER BY rs.difficulty ASC, rs.score_rank ASC;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[21]:

Dev_ID	difficulty	score	score_rank
zm_017	Difficult	5500	1
zm_017	Difficult	5500	1
bd_015	Difficult	5300	3
bd_013	Difficult	5300	3
rf_017	Difficult	5140	5
zm_015	Low	3470	1
zm_017	Low	3210	2
bd_015	Low	3200	3
bd_013	Low	2840	4
zm_015	Low	2800	5
zm_017	Medium	5490	1
rf_017	Medium	5140	2
zm_015	Medium	4950	3
zm_015	Medium	4950	3
rf_015	Medium	2800	5

11. For each player and date, determine how many `kill_counts` were played by the player so far.

- Using window functions
- Without window functions

In [22]:
```
%%sql
SELECT ld.P_ID, ld.timestamp, SUM(ld.kill_count) OVER (PARTITION BY ld.P_ID ORDER B
FROM Level_Details ld; --Using window functions
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[22]:

P_ID	TimeStamp	total_kill_count
211	2022-10-12 13:23:45	20
211	2022-10-12 18:30:30	45
211	2022-10-13 05:36:15	75
211	2022-10-13 22:30:18	89
211	2022-10-14 08:56:24	98
211	2022-10-15 11:41:19	113
224	2022-10-14 01:15:56	20
224	2022-10-14 08:21:49	54
224	2022-10-15 05:30:28	84
224	2022-10-15 13:43:50	112
242	2022-10-13 01:14:29	21
242	2022-10-14 04:38:50	58
292	2022-10-12 04:29:45	21
292	2022-10-15 10:19:30	25
296	2022-10-14 15:15:15	7
296	2022-10-14 19:35:49	11
300	2022-10-11 05:20:40	23
300	2022-10-11 19:19:19	48
300	2022-10-12 01:45:17	52
300	2022-10-12 11:21:20	66
300	2022-10-13 23:15:42	74
310	2022-10-11 15:15:15	20
310	2022-10-13 19:18:20	54
310	2022-10-15 23:30:50	68
319	2022-10-12 14:20:40	5
358	2022-10-14 05:05:05	4
358	2022-10-14 18:23:29	7
368	2022-10-12 01:14:34	20
368	2022-10-12 04:20:30	34
368	2022-10-12 11:59:18	49
368	2022-10-15 14:47:53	73
428	2022-10-15 18:00:00	5
429	2022-10-11 09:28:56	30
429	2022-10-11 13:00:22	55
429	2022-10-11 19:28:43	69
429	2022-10-11 21:39:00	99

P_ID	TimeStamp	total_kill_count
483	2022-10-11 14:33:27	50
483	2022-10-11 22:20:10	70
483	2022-10-12 02:40:20	89
483	2022-10-12 19:30:11	109
483	2022-10-13 06:20:40	134
547	2022-10-15 02:19:27	15
547	2022-10-15 07:15:15	32
547	2022-10-15 20:16:49	52
558	2022-10-12 23:19:17	21
590	2022-10-12 07:30:18	24
590	2022-10-12 19:23:15	34
590	2022-10-13 04:20:27	51
590	2022-10-13 13:45:40	60
590	2022-10-14 06:31:24	75
632	2022-10-12 16:30:30	45
632	2022-10-12 19:36:40	73
632	2022-10-13 06:30:20	77
632	2022-10-13 10:56:17	100
632	2022-10-14 23:41:25	130
641	2022-10-13 04:04:04	2
641	2022-10-14 01:25:30	6
641	2022-10-14 23:19:17	14
644	2022-10-11 14:05:08	11
644	2022-10-11 19:34:25	18
644	2022-10-12 23:52:18	42
656	2022-10-11 17:47:09	18
656	2022-10-13 22:19:45	37
656	2022-10-14 07:32:00	40
656	2022-10-15 18:12:50	55
663	2022-10-15 06:30:20	4
663	2022-10-15 09:56:17	27
663	2022-10-15 17:30:30	72
663	2022-10-15 19:36:40	100
663	2022-10-15 23:41:25	130
683	2022-10-11 02:23:45	16
683	2022-10-11 18:45:55	37

P_ID	TimeStamp	total_kill_count
683	2022-10-12 14:36:15	53
683	2022-10-13 08:16:29	72
683	2022-10-13 22:30:17	97
683	2022-10-15 16:10:30	117
683	2022-10-15 22:20:16	127

In [23]:
```
%%sql
SELECT ld.P_ID, ld.timestamp, SUM(ld2.kill_count) AS total_kill_count
FROM Level_Details ld
INNER JOIN Level_Details ld2 ON ld.P_ID = ld2.P_ID AND ld.timestamp >= ld2.timestam
GROUP BY ld.P_ID, ld.timestamp; --Without window function
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[23]:

P_ID	TimeStamp	total_kill_count
211	2022-10-12 13:23:45	20
211	2022-10-12 18:30:30	45
211	2022-10-13 05:36:15	75
211	2022-10-13 22:30:18	89
211	2022-10-14 08:56:24	98
211	2022-10-15 11:41:19	113
224	2022-10-14 01:15:56	20
224	2022-10-14 08:21:49	54
224	2022-10-15 05:30:28	84
224	2022-10-15 13:43:50	112
242	2022-10-13 01:14:29	21
242	2022-10-14 04:38:50	58
292	2022-10-12 04:29:45	21
292	2022-10-15 10:19:30	25
296	2022-10-14 15:15:15	7
296	2022-10-14 19:35:49	11
300	2022-10-11 05:20:40	23
300	2022-10-11 19:19:19	48
300	2022-10-12 01:45:17	52
300	2022-10-12 11:21:20	66
300	2022-10-13 23:15:42	74
310	2022-10-11 15:15:15	20
310	2022-10-13 19:18:20	54
310	2022-10-15 23:30:50	68
319	2022-10-12 14:20:40	5
358	2022-10-14 05:05:05	4
358	2022-10-14 18:23:29	7
368	2022-10-12 01:14:34	20
368	2022-10-12 04:20:30	34
368	2022-10-12 11:59:18	49
368	2022-10-15 14:47:53	73
428	2022-10-15 18:00:00	5
429	2022-10-11 09:28:56	30
429	2022-10-11 13:00:22	55
429	2022-10-11 19:28:43	69
429	2022-10-11 21:39:00	99

P_ID	TimeStamp	total_kill_count
483	2022-10-11 14:33:27	50
483	2022-10-11 22:20:10	70
483	2022-10-12 02:40:20	89
483	2022-10-12 19:30:11	109
483	2022-10-13 06:20:40	134
547	2022-10-15 02:19:27	15
547	2022-10-15 07:15:15	32
547	2022-10-15 20:16:49	52
558	2022-10-12 23:19:17	21
590	2022-10-12 07:30:18	24
590	2022-10-12 19:23:15	34
590	2022-10-13 04:20:27	51
590	2022-10-13 13:45:40	60
590	2022-10-14 06:31:24	75
632	2022-10-12 16:30:30	45
632	2022-10-12 19:36:40	73
632	2022-10-13 06:30:20	77
632	2022-10-13 10:56:17	100
632	2022-10-14 23:41:25	130
641	2022-10-13 04:04:04	2
641	2022-10-14 01:25:30	6
641	2022-10-14 23:19:17	14
644	2022-10-11 14:05:08	11
644	2022-10-11 19:34:25	18
644	2022-10-12 23:52:18	42
656	2022-10-11 17:47:09	18
656	2022-10-13 22:19:45	37
656	2022-10-14 07:32:00	40
656	2022-10-15 18:12:50	55
663	2022-10-15 06:30:20	4
663	2022-10-15 09:56:17	27
663	2022-10-15 17:30:30	72
663	2022-10-15 19:36:40	100
663	2022-10-15 23:41:25	130
683	2022-10-11 02:23:45	16
683	2022-10-11 18:45:55	37

P_ID	TimeStamp	total_kill_count
683	2022-10-12 14:36:15	53
683	2022-10-13 08:16:29	72
683	2022-10-13 22:30:17	97
683	2022-10-15 16:10:30	117
683	2022-10-15 22:20:16	127

12. Find the cumulative sum of stages crossed over `start_datetime` for each `P_ID`,excluding the most recent `start_datetime`.

In [24]:
```
%%sql
SELECT ld.P_ID,
       ld.timestamp,
       SUM(ld.stages_crossed) OVER (PARTITION BY ld.P_ID ORDER BY ld.timestamp ROWS
FROM Level_Details ld
WHERE ld.timestamp < (SELECT MAX(timestamp) FROM Level_Details ld2 WHERE ld.P_ID =
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[24]:

P_ID	TimeStamp	cumulative_stages_crossed
211	2022-10-12 13:23:45	None
211	2022-10-12 18:30:30	4
211	2022-10-13 05:36:15	9
211	2022-10-13 22:30:18	14
211	2022-10-14 08:56:24	19
224	2022-10-14 01:15:56	None
224	2022-10-14 08:21:49	7
224	2022-10-15 05:30:28	12
242	2022-10-13 01:14:29	None
292	2022-10-12 04:29:45	None
296	2022-10-14 15:15:15	None
300	2022-10-11 05:20:40	None
300	2022-10-11 19:19:19	7
300	2022-10-12 01:45:17	12
300	2022-10-12 11:21:20	14
310	2022-10-11 15:15:15	None
310	2022-10-13 19:18:20	7
358	2022-10-14 05:05:05	None
368	2022-10-12 01:14:34	None
368	2022-10-12 04:20:30	7
368	2022-10-12 11:59:18	12
429	2022-10-11 09:28:56	None
429	2022-10-11 13:00:22	2
429	2022-10-11 19:28:43	9
483	2022-10-11 14:33:27	None
483	2022-10-11 22:20:10	10
483	2022-10-12 02:40:20	15
483	2022-10-12 19:30:11	22
547	2022-10-15 02:19:27	None
547	2022-10-15 07:15:15	8
590	2022-10-12 07:30:18	None
590	2022-10-12 19:23:15	3
590	2022-10-13 04:20:27	5
590	2022-10-13 13:45:40	10
632	2022-10-12 16:30:30	None
632	2022-10-12 19:36:40	5

P_ID	TimeStamp	cumulative_stages_crossed
632	2022-10-13 06:30:20	10
632	2022-10-13 10:56:17	15
641	2022-10-13 04:04:04	None
641	2022-10-14 01:25:30	2
644	2022-10-11 14:05:08	None
644	2022-10-11 19:34:25	3
656	2022-10-11 17:47:09	None
656	2022-10-13 22:19:45	10
656	2022-10-14 07:32:00	14
663	2022-10-15 06:30:20	None
663	2022-10-15 09:56:17	5
663	2022-10-15 17:30:30	10
663	2022-10-15 19:36:40	15
683	2022-10-11 02:23:45	None
683	2022-10-11 18:45:55	4
683	2022-10-12 14:36:15	7
683	2022-10-13 08:16:29	14
683	2022-10-13 22:30:17	21
683	2022-10-15 16:10:30	26

13. Extract the top 3 highest sums of scores for each `Dev_ID` and the corresponding `P_ID`.

In [25]:
```
%%sql
SELECT Dev_ID, P_ID, total_score
FROM (
    SELECT Dev_ID, P_ID, SUM(score) AS total_score,
           ROW_NUMBER() OVER (PARTITION BY Dev_ID ORDER BY SUM(score) DESC) AS rn
    FROM Level_Details
    GROUP BY Dev_ID, P_ID
) t
WHERE rn <= 3;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[25]:

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
wd_019	590	1550
wd_019	558	635
zm_013	429	4710
zm_013	483	2350
zm_013	358	120
zm_015	663	4950
zm_015	632	4950
zm_015	242	3470
zm_017	683	8900
zm_017	632	5600
zm_017	663	5500

14. Find players who scored more than 50% of the average score, scored by the sum of scores for each `P_ID`.

In [26]:
```sql
%%sql
SELECT pd.P_ID, pd.PName
FROM Player_Details pd
JOIN (
    SELECT P_ID, SUM(score) AS total_score
    FROM Level_Details
    GROUP BY P_ID
) AS total_scores ON pd.P_ID = total_scores.P_ID
WHERE pd.P_ID IN (
    SELECT P_ID
    FROM Level_Details
    GROUP BY P_ID
    HAVING SUM(score) > 0 -- Ensuring players have non-zero total score
)
AND pd.P_ID IN (
    SELECT P_ID
    FROM Level_Details
    GROUP BY P_ID
    HAVING AVG(score) > 0 -- Ensuring players have non-zero average score
)
AND pd.P_ID IN (
    SELECT P_ID
    FROM Level_Details
    GROUP BY P_ID
    HAVING AVG(score) > total_scores.total_score * 0.5
);
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[26]:

P_ID	PName
428	leaky-magnolia-iguana
319	chummy-flax-crab
558	woozy-crimson-hound

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.

In [27]:
```sql
%%sql
DROP VIEW IF EXISTS TopHeadshotsByDeviceID;

CREATE VIEW TopHeadshotsByDeviceID AS
WITH RankedHeadshots AS (
    SELECT ld.Dev_ID,
           ld.difficulty,
           ld.headshots_count,
           ROW_NUMBER() OVER (PARTITION BY ld.Dev_ID ORDER BY ld.headshots_count AS
      FROM Level_Details ld
)
SELECT Dev_ID,
       difficulty,
```

```
        headshots_count,
        HeadshotRank
    FROM RankedHeadshots;
```

```
 * sqlite:///Game_Analysis.db
Done.
Done.
```

Out[27]:
```
[]
```

In [28]:
```
%%sql
SELECT * FROM TopHeadshotsByDeviceID WHERE HeadshotRank <= 5;
```

```
 * sqlite:///Game_Analysis.db
Done.
```

Out[28]:

Dev_ID	difficulty	headshots_count	HeadshotRank
bd_013	Medium	4	1
bd_013	Medium	8	2
bd_013	Medium	10	3
bd_013	Difficult	11	4
bd_013	Low	11	5
bd_015	Low	3	1
bd_015	Difficult	8	2
bd_015	Low	13	3
bd_015	Medium	17	4
bd_015	Low	20	5
bd_017	Low	15	1
bd_017	Medium	16	2
bd_017	Low	18	3
rf_013	Low	3	1
rf_013	Medium	6	2
rf_013	Low	7	3
rf_013	Difficult	7	4
rf_013	Medium	8	5
rf_015	Medium	0	1
rf_015	Medium	1	2
rf_015	Low	2	3
rf_015	Medium	3	4
rf_015	Difficult	10	5
rf_017	Difficult	1	1
rf_017	Difficult	11	2
rf_017	Difficult	18	3
rf_017	Medium	18	4
rf_017	Difficult	27	5
wd_019	Difficult	0	1
wd_019	Low	10	2
wd_019	Difficult	16	3
wd_019	Difficult	19	4
zm_013	Medium	1	1
zm_013	Medium	10	2
zm_013	Difficult	20	3
zm_015	Medium	0	1

Dev_ID	difficulty	headshots_count	HeadshotRank
zm_015	Difficult	0	2
zm_015	Medium	3	3
zm_015	Medium	5	4
zm_015	Low	8	5
zm_017	Difficult	0	1
zm_017	Difficult	3	2
zm_017	Medium	3	3
zm_017	Low	3	4
zm_017	Low	3	5

In [ ]: