SQL PROJECT

VENKADESH V

Project Name: Decode Gaming Behaviour

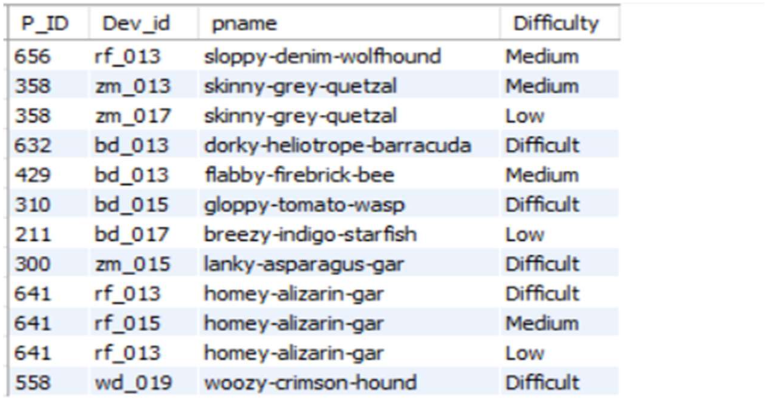
Create database: **create database game_analysis;**

Use database : **use game_analysis;**

Question 1

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

```
select l.P_ID, l.Dev_id,p.pname, l.Difficulty from level_details2 as l

join player_details as P on l.P_ID = P.P_id

where level = 0;
```

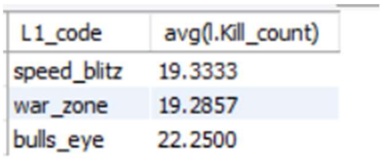


P_ID	Dev_id	pname	Difficulty
656	rf_013	sloppy-denim-wolfhound	Medium
358	zm_013	skinny-grey-quetzal	Medium
358	zm_017	skinny-grey-quetzal	Low
632	bd_013	dorky-heliotrope-barracuda	Difficult
429	bd_013	flabby-firebrick-bee	Medium
310	bd_015	gloppy-tomato-wasp	Difficult
211	bd_017	breezy-indigo-starfish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

Question 2

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

```
select p.L1_code, avg(l.Kill_count)  from player_details as p

join level_details2 as l on p.P_ID = l.P_id

where l.lives_earned = 2 and l.stages_crossed >=3

group by p.l1_code;
```

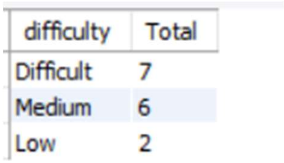


L1_code	avg(l.Kill_count)
speed_blitz	19.3333
war_zone	19.2857
bulls_eye	22.2500

Question 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.

```
select difficulty, count(stages_crossed) as Total from level_details2

where level = 2 and Dev_Id like 'zm%%'

group by difficulty

order by Total desc;
```

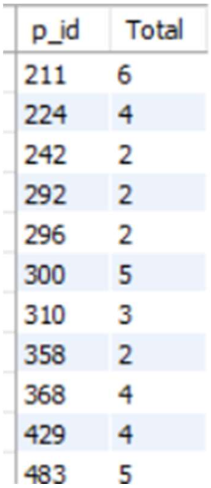


difficulty	Total
Difficult	7
Medium	6
Low	2

Question 4

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

```
select p.p_id,  count(distinct( l.TimeStamp)) as Total from player_details as p

join level_details2 as l on p.P_ID = l.P_id

group by P.P_id

having Total>1;
```

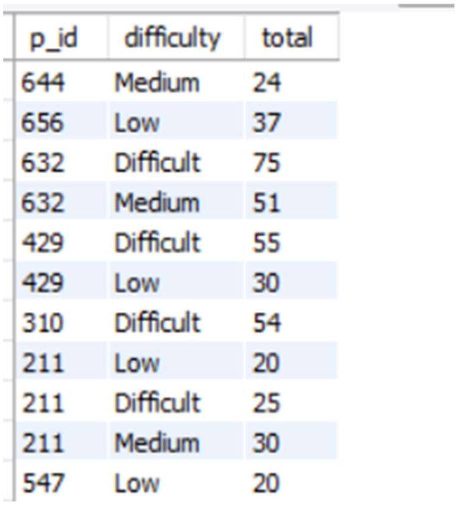


p_id	Total
211	6
224	4
242	2
292	2
296	2
300	5
310	3
358	2
368	4
429	4
483	5

Question 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, difficulty,sum(kill_count) as total from level_details2

where kill_count > (select avg(kill_count) as average from level_details2 where difficulty = "Medium")

group by P_id,difficulty;
```

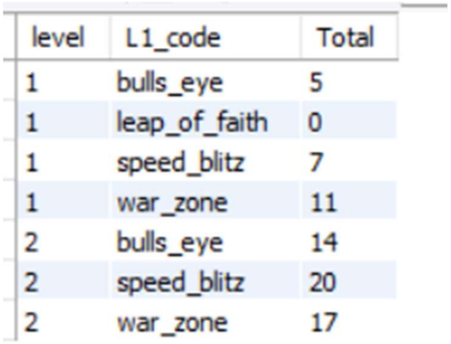


p_id	difficulty	total
644	Medium	24
656	Low	37
632	Difficult	75
632	Medium	51
429	Difficult	55
429	Low	30
310	Difficult	54
211	Low	20
211	Difficult	25
211	Medium	30
547	Low	20

Question 6

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

```
select l.level,p.L1_code, sum(l.lives_earned) as Total from level_details2 as l

join player_details as p on l.P_id = p.P_id

where l.level <>0

group by l.level,p.l1_code

order by level asc;
```

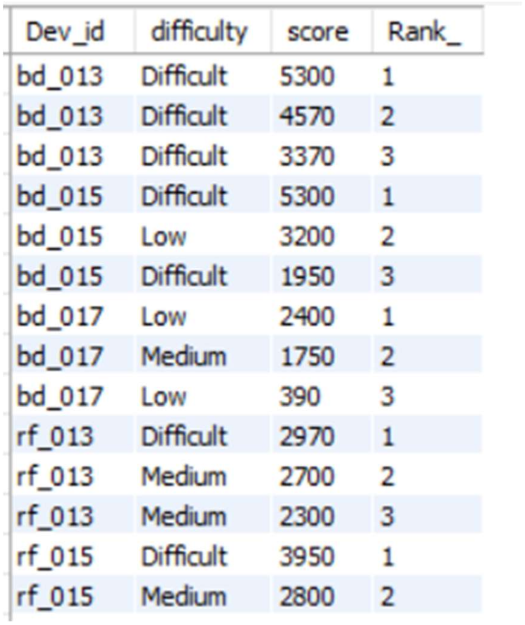


level	L1_code	Total
1	bulls_eye	5
1	leap_of_faith	0
1	speed_blitz	7
1	war_zone	11
2	bulls_eye	14
2	speed_blitz	20
2	war_zone	17

Question 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 demo as (select Dev_id,difficulty, score ,

row_number() over (partition by dev_id order by score desc) as Rank_

from level_details2)

select Dev_id,difficulty,score,Rank_ from demo

where rank_<=3;
```

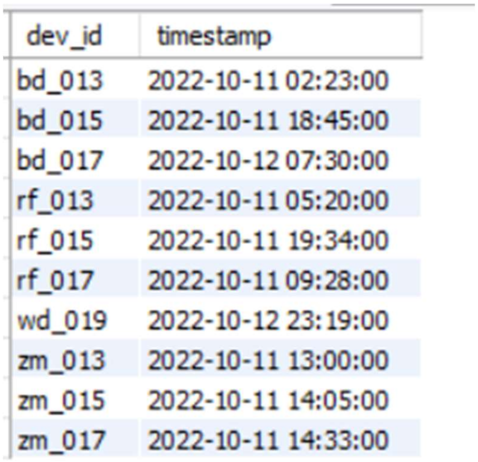


Dev_id	difficulty	score	Rank_
bd_013	Difficult	5300	1
bd_013	Difficult	4570	2
bd_013	Difficult	3370	3
bd_015	Difficult	5300	1
bd_015	Low	3200	2
bd_015	Difficult	1950	3
bd_017	Low	2400	1
bd_017	Medium	1750	2
bd_017	Low	390	3
rf_013	Difficult	2970	1
rf_013	Medium	2700	2
rf_013	Medium	2300	3
rf_015	Difficult	3950	1
rf_015	Medium	2800	2

Question 8

Find the `first_login` datetime for each device ID.

```
with first_login as(

select * , row_number() over (partition by dev_id order by timestamp) as Rank_ from level_details2)

select dev_id,timestamp from first_login

where Rank_ = 1;
```

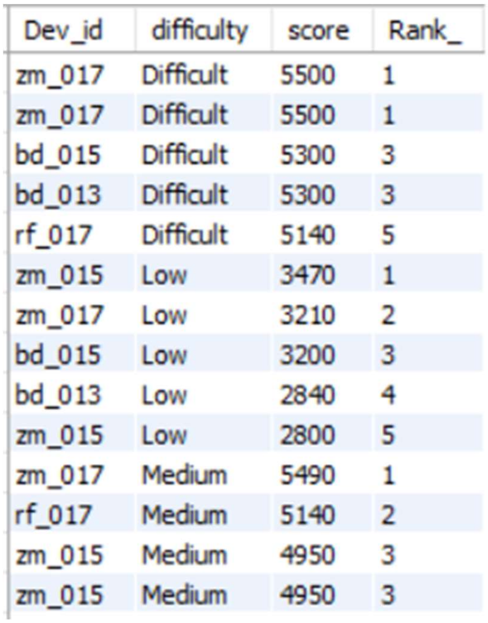


dev_id	timestamp
bd_013	2022-10-11 02:23:00
bd_015	2022-10-11 18:45:00
bd_017	2022-10-12 07:30:00
rf_013	2022-10-11 05:20:00
rf_015	2022-10-11 19:34:00
rf_017	2022-10-11 09:28:00
wd_019	2022-10-12 23:19:00
zm_013	2022-10-11 13:00:00
zm_015	2022-10-11 14:05:00
zm_017	2022-10-11 14:33:00

Question 9

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

```
with Top_5 as(select Dev_id,difficulty, score ,

rank() over (partition by difficulty order by score desc) as Rank_

from level_details2 )

select Dev_id,difficulty,score,Rank_ from Top_5

where rank_<=5;
```

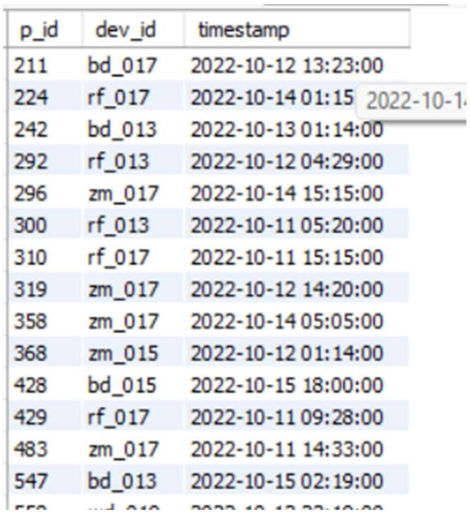


Dev_id	difficulty	score	Rank_
zm_017	Difficult	5500	1
zm_017	Difficult	5500	1
bd_015	Difficult	5300	3
bd_013	Difficult	5300	3
rf_017	Difficult	5140	5
zm_015	Low	3470	1
zm_017	Low	3210	2
bd_015	Low	3200	3
bd_013	Low	2840	4
zm_015	Low	2800	5
zm_017	Medium	5490	1
rf_017	Medium	5140	2
zm_015	Medium	4950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

Question 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.

```
with first_login as (select * , row_number() over (partition by p_id order by timestamp) as Rank_

 from level_details2)

select p_id,dev_id,timestamp from first_login

where Rank_ = 1;
```



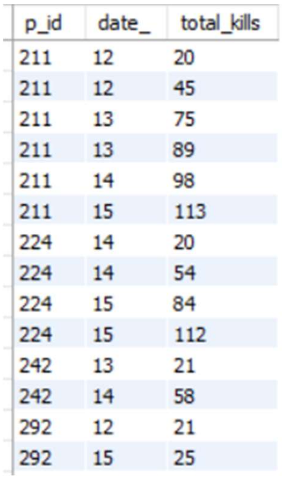
p_id	dev_id	timestamp
211	bd_017	2022-10-12 13:23:00
224	rf_017	2022-10-14 01:15
242	bd_013	2022-10-13 01:14:00
292	rf_013	2022-10-12 04:29:00
296	zm_017	2022-10-14 15:15:00
300	rf_013	2022-10-11 05:20:00
310	rf_017	2022-10-11 15:15:00
319	zm_017	2022-10-12 14:20:00
358	zm_017	2022-10-14 05:05:00
368	zm_015	2022-10-12 01:14:00
428	bd_015	2022-10-15 18:00:00
429	rf_017	2022-10-11 09:28:00
483	zm_017	2022-10-11 14:33:00
547	bd_013	2022-10-15 02:19:00

Question 11

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

a) Using window functions

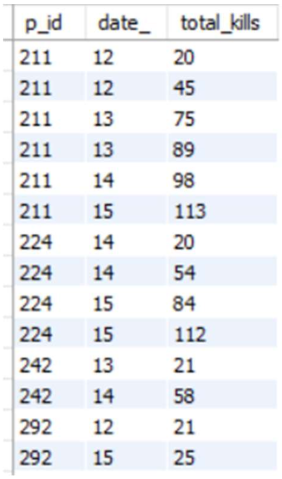
```
select p_id, day(timestamp)as date_ , sum(kill_count) over(partition by p_id  order by timestamp ) as
total_kills
from level_details2;
```



p_id	date_	total_kills
211	12	20
211	12	45
211	13	75
211	13	89
211	14	98
211	15	113
224	14	20
224	14	54
224	15	84
224	15	112
242	13	21
242	14	58
292	12	21
292	15	25

b) Without window functions

```
select a.p_id, day(a.timestamp) as date_, sum(b.kill_count) as total_kills

from level_details2 as a

 join level_details2 as b on a.p_id = b.p_id

where a.timestamp >= b.timestamp

group by a.p_id, a.timestamp

order by a.p_id;
```

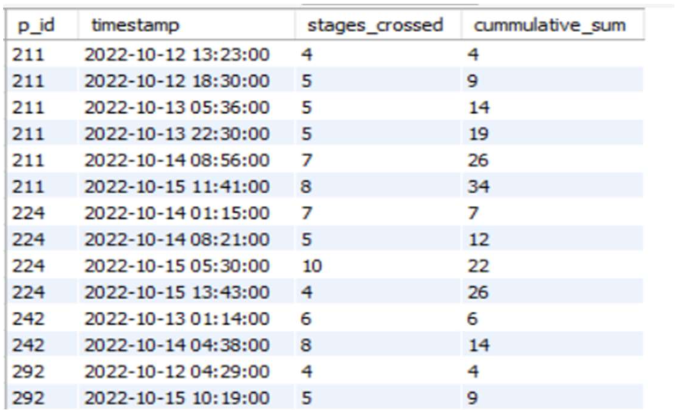


p_id	date_	total_kills
211	12	20
211	12	45
211	13	75
211	13	89
211	14	98
211	15	113
224	14	20
224	14	54
224	15	84
224	15	112
242	13	21
242	14	58
292	12	21
292	15	25

Question 12

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

```
select p_id,timestamp,stages_crossed,sum(stages_crossed)

over(partition by p_id order by timestamp rows between unbounded preceding and current row) as cummulative_sum

from level_details2;
```

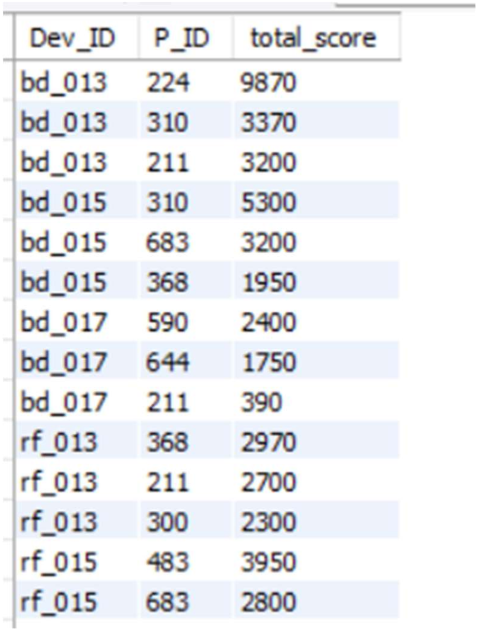


p_id	timestamp	stages_crossed	cummulative_sum
211	2022-10-12 13:23:00	4	4
211	2022-10-12 18:30:00	5	9
211	2022-10-13 05:36:00	5	14
211	2022-10-13 22:30:00	5	19
211	2022-10-14 08:56:00	7	26
211	2022-10-15 11:41:00	8	34
224	2022-10-14 01:15:00	7	7
224	2022-10-14 08:21:00	5	12
224	2022-10-15 05:30:00	10	22
224	2022-10-15 13:43:00	4	26
242	2022-10-13 01:14:00	6	6
242	2022-10-14 04:38:00	8	14
292	2022-10-12 04:29:00	4	4
292	2022-10-15 10:19:00	5	9

Question 13

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

```
WITH top_3 AS (SELECT Dev_ID, P_ID,SUM(score) AS total_score,

ROW_NUMBER() OVER (PARTITION BY Dev_ID ORDER BY SUM(score) DESC) AS ranked

FROM level_details2

GROUP BY Dev_ID,P_ID)

SELECT Dev_ID,P_ID,total_score FROM top_3

WHERE ranked <= 3;
```

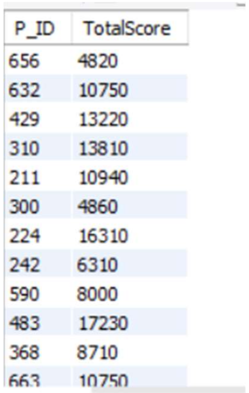


Dev_ID	P_ID	total_score
bd_013	224	9870
bd_013	310	3370
bd_013	211	3200
bd_015	310	5300
bd_015	683	3200
bd_015	368	1950
bd_017	590	2400
bd_017	644	1750
bd_017	211	390
rf_013	368	2970
rf_013	211	2700
rf_013	300	2300
rf_015	483	3950
rf_015	683	2800

Question 14

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

```
SELECT P_ID, TotalScore

FROM (SELECT P_ID, SUM(Score) AS TotalScore FROM level_details2

GROUP BY P_ID) AS PlayerScores

WHERE TotalScore > ( SELECT AVG(TotalScore) * 0.5 FROM ( SELECT SUM(Score) AS TotalScore

FROM level_details2

GROUP BY P_ID) AS AvgScore );
```

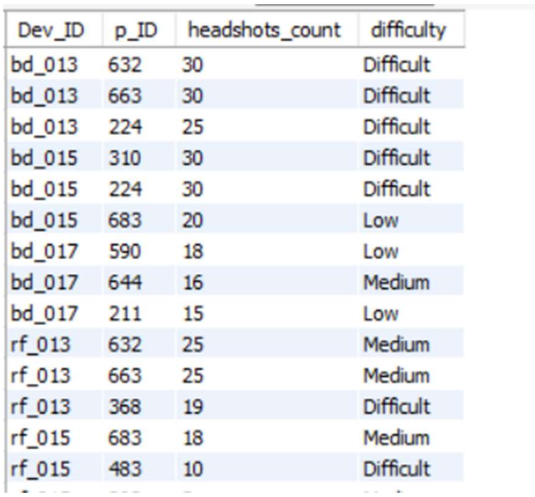


P_ID	TotalScore
656	4820
632	10750
429	13220
310	13810
211	10940
300	4860
224	16310
242	6310
590	8000
483	17230
368	8710
663	10750

Question 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.

```
DELIMITER //

CREATE PROCEDURE heatshotcount(IN n INT)

BEGIN

SELECT Dev_ID, p_ID, headshots_count, difficulty

FROM (SELECT Dev_ID, p_ID, headshots_count, difficulty,

ROW_NUMBER() OVER (PARTITION BY Dev_ID ORDER BY headshots_count DESC) AS Rank_

FROM level_details2) AS RankedHeadshots

WHERE Rank_ <= n;

END //

DELIMITER ;


call heatshotcount(3)   -- Replace 3 with the desired value of n
```



Dev_ID	p_ID	headshots_count	difficulty
bd_013	632	30	Difficult
bd_013	663	30	Difficult
bd_013	224	25	Difficult
bd_015	310	30	Difficult
bd_015	224	30	Difficult
bd_015	683	20	Low
bd_017	590	18	Low
bd_017	644	16	Medium
bd_017	211	15	Low
rf_013	632	25	Medium
rf_013	663	25	Medium
rf_013	368	19	Difficult
rf_015	683	18	Medium
rf_015	483	10	Difficult