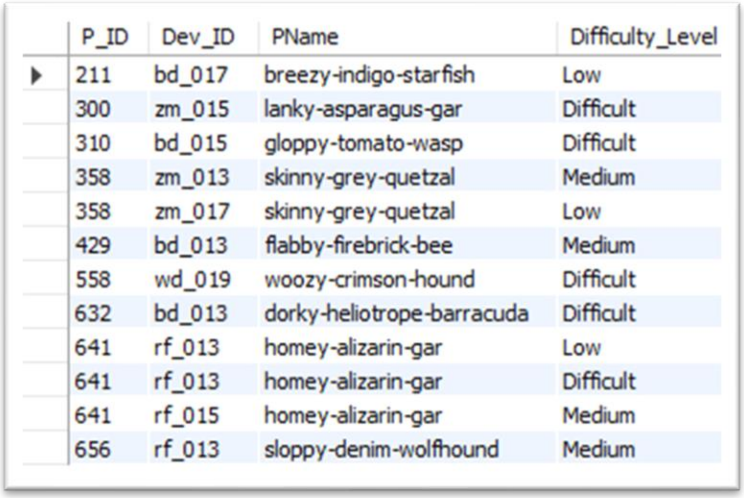
Decode Gaming Behavior

About Dataset : working with a dataset related to a game. The dataset includes two tables: `Player Details` and `Level Details`.

Problem Statement : 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.

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

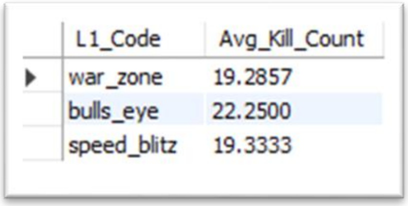
```
Select T1.P_ID,
        T2.Dev_ID,
        T1.PName,
        T2.Difficulty as Difficulty_Level
From player_details as T1 Inner Join level_details as T2 Using(P_ID)
Where T2.Level=0;
```



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

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

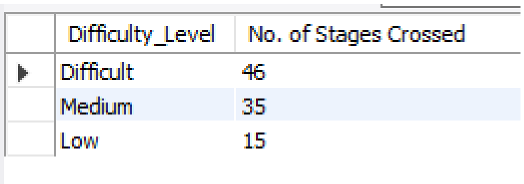
```
Select T1.L1_Code,
        avg(T2.Kill_Count) as Avg_Kill_Count
From player_details as T1 Inner Join level_details as T2 Using(P_ID)
Where T2.Stages_crossed>=3 and T2.Lives_Earned=2
Group by T1.L1_Code;
```



L1_Code	Avg_Kill_Count
war_zone	19.2857
bulls_eye	22.2500
speed_blitz	19.3333

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.

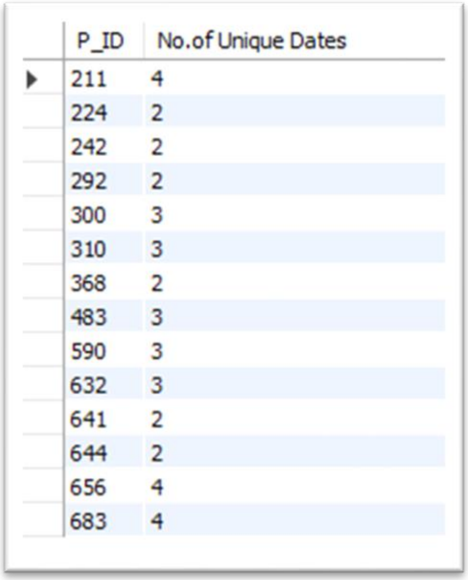
```
Select Difficulty as Difficulty_Level,Sum(stages_crossed) as "No. of Stages
Crossed"
From level_details
where level=2 and Dev_ID like "zm%"
Group by Difficulty_Level
order by Sum(stages_crossed) desc;
```



Difficulty_Level	No. of 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.

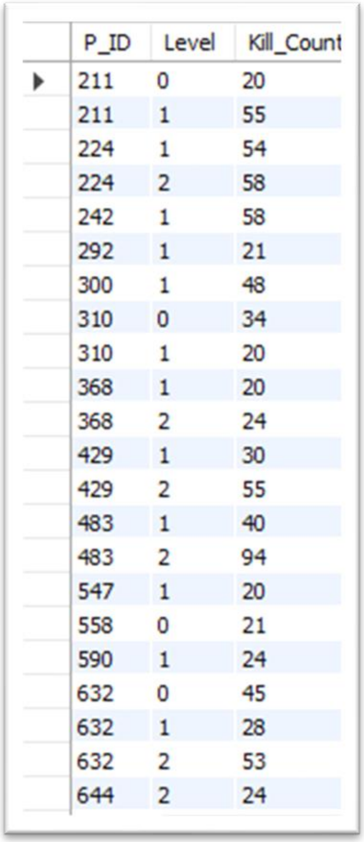
```
Select T1.P_ID, Count(Distinct(Date(T2.start_datetime))) as "No.of Unique Dates"
From player_details as T1 Inner Join level_details as T2 Using(P_ID)
Group by T1.P_ID
Having Count(Distinct(Date(T2.start_datetime)))>1;
```



P_ID	No.of Unique Dates
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.

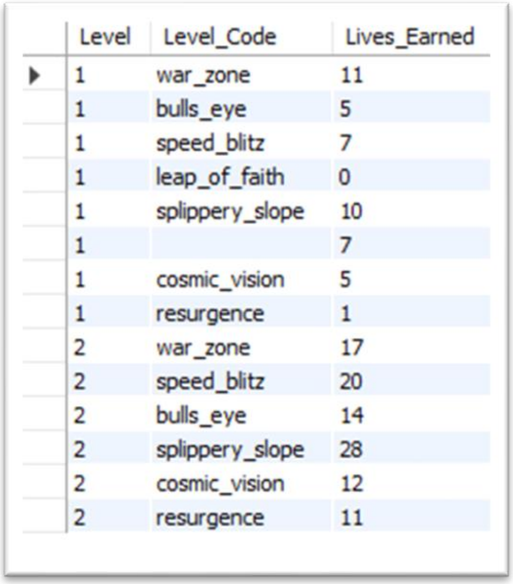
```
Select P_ID,Level,Sum(kill_count) as Kill_Count

From level_details

Where kill_count >(Select avg(kill_count) From level_details where Difficulty="Medium")

Group by P_ID,Level

order by P_ID,Level;
```



P_ID	Level	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

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

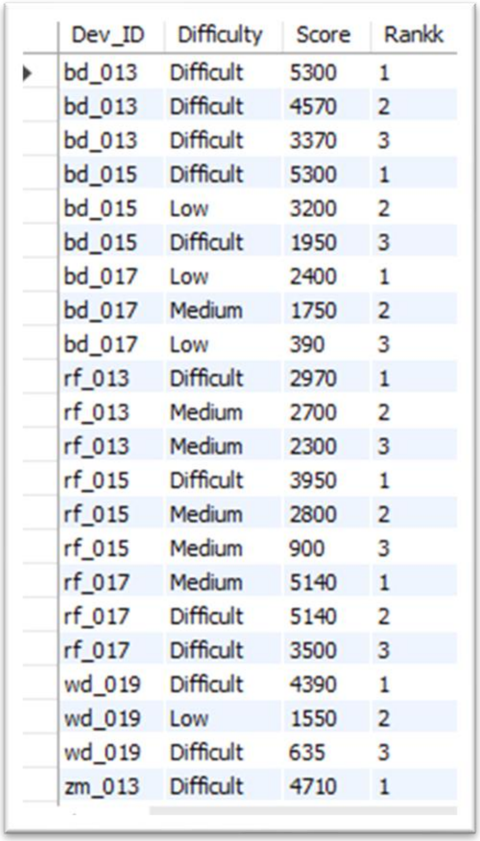
```
Select T2.Level,T1.L1_Code as Level_Code,Sum(T2.Lives_Earned) as Lives_Earned

From player_details as T1 Inner Join level_details as T2 Using(P_ID)

Where T2.Level<>0

Group by T2.Level,T1.L1_Code


Union All


Select T2.Level,T1.L2_Code as Level_Code,Sum(T2.Lives_Earned) as Lives_Earned

From player_details as T1 Inner Join level_details as T2 Using(P_ID)

Where T2.Level<>0

Group by T2.Level,T1.L2_Code

Order by Level;
```



Level	Level_Code	Lives_Earned
1	war_zone	11
1	bulls_eye	5
1	speed_blitz	7
1	leap_of_faith	0
1	splippery_slope	10
1		7
1	cosmic_vision	5
1	resurgence	1
2	war_zone	17
2	speed_blitz	20
2	bulls_eye	14
2	splippery_slope	28
2	cosmic_vision	12
2	resurgence	11

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.

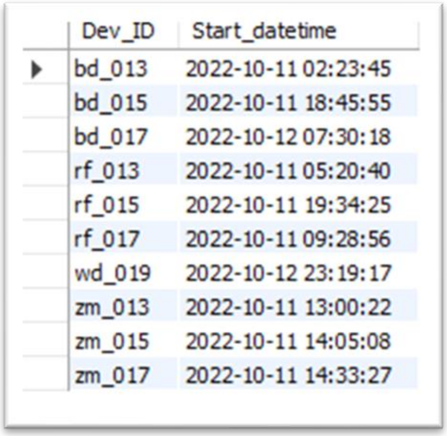
```
Select *

From (Select Dev_ID,Difficulty,Score,Row_Number() Over(Partition by Dev_ID
Order by Score desc) as Rankk From Level_Details) as Project

Where Rankk<=3;
```



Dev_ID	Difficulty	Score	Rankk
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
rf_015	Medium	900	3
rf_017	Medium	5140	1
rf_017	Difficult	5140	2
rf_017	Difficult	3500	3
wd_019	Difficult	4390	1
wd_019	Low	1550	2
wd_019	Difficult	635	3
zm_013	Difficult	4710	1

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

```
Select Dev_ID,Start_datetime

From (Select Dev_ID,start_datetime,Row_number() Over(Partition by Dev_ID
Order by start_datetime) as Rankk

From level_details) as Project

Where Rankk=1;
```

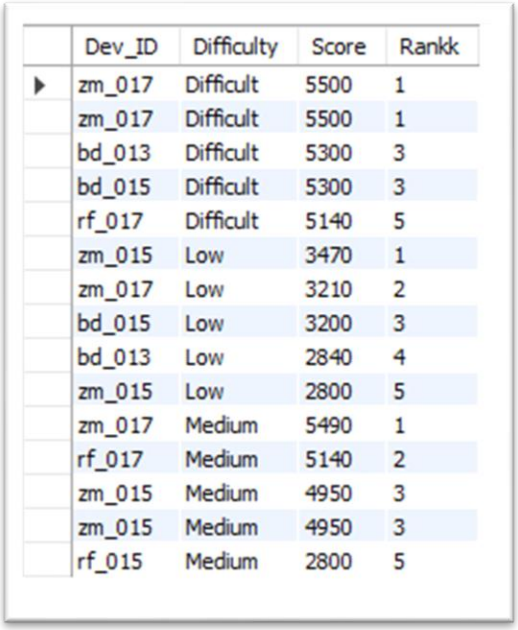


	Dev_ID	Start_datetime
▶	bd_013	2022-10-11 02:23:45
	bd_015	2022-10-11 18:45:55
	bd_017	2022-10-12 07:30:18
	rf_013	2022-10-11 05:20:40
	rf_015	2022-10-11 19:34:25
	rf_017	2022-10-11 09:28:56
	wd_019	2022-10-12 23:19:17
	zm_013	2022-10-11 13:00:22
	zm_015	2022-10-11 14:05:08
	zm_017	2022-10-11 14:33:27

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

```
Select *

From (Select Dev_ID,

                    Difficulty,

                    Score,Rank() Over(Partition by Difficulty Order by Score desc) as

                     Rankk From Level_details) as Project

Where Rankk<=5;
```

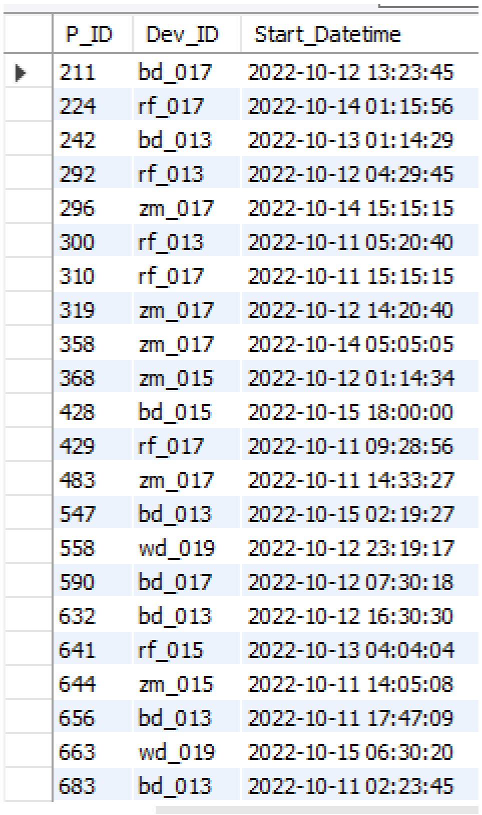
.



Dev_ID	Difficulty	Score	Rankk
zm_017	Difficult	5500	1
zm_017	Difficult	5500	1
bd_013	Difficult	5300	3
bd_015	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 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.

```
Select P_ID,Dev_ID,Start_Datetime

From (Select P_ID,Dev_ID,Start_Datetime,Rank() Over(Partition by P_ID Order by start_Datetime) as Rankk From Level_details) as Project

Where Rankk=1;
```



P_ID	Dev_ID	Start_Datetime
211	bd_017	2022-10-12 13:23:45
224	rf_017	2022-10-14 01:15:56
242	bd_013	2022-10-13 01:14:29
292	rf_013	2022-10-12 04:29:45
296	zm_017	2022-10-14 15:15:15
300	rf_013	2022-10-11 05:20:40
310	rf_017	2022-10-11 15:15:15
319	zm_017	2022-10-12 14:20:40
358	zm_017	2022-10-14 05:05:05
368	zm_015	2022-10-12 01:14:34
428	bd_015	2022-10-15 18:00:00
429	rf_017	2022-10-11 09:28:56
483	zm_017	2022-10-11 14:33:27
547	bd_013	2022-10-15 02:19:27
558	wd_019	2022-10-12 23:19:17
590	bd_017	2022-10-12 07:30:18
632	bd_013	2022-10-12 16:30:30
641	rf_015	2022-10-13 04:04:04
644	zm_015	2022-10-11 14:05:08
656	bd_013	2022-10-11 17:47:09
663	wd_019	2022-10-15 06:30:20
683	bd_013	2022-10-11 02:23:45

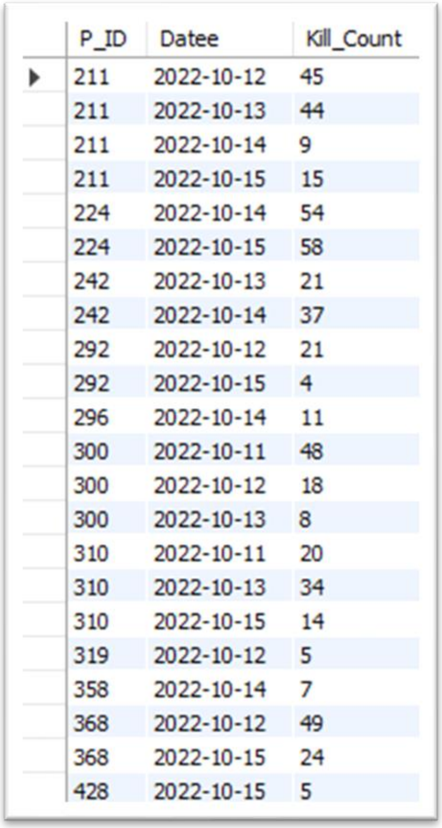
11) For each player and date, determine how many `kill_counts` were played by the player so far.

a) Using window functions

```
Select P_ID,Datee,Kill_Count

From (Select P_ID,Date(Start_datetime) as Datee,SUM(kill_Count) Over(Partition by P_ID,Date(Start_datetime)) as Kill_Count,

        Row_Number() Over(Partition by P_ID,Date(Start_datetime)) as Row_No

        From level_details) as Project

Where Row_No=1;
```

b) Without window functions

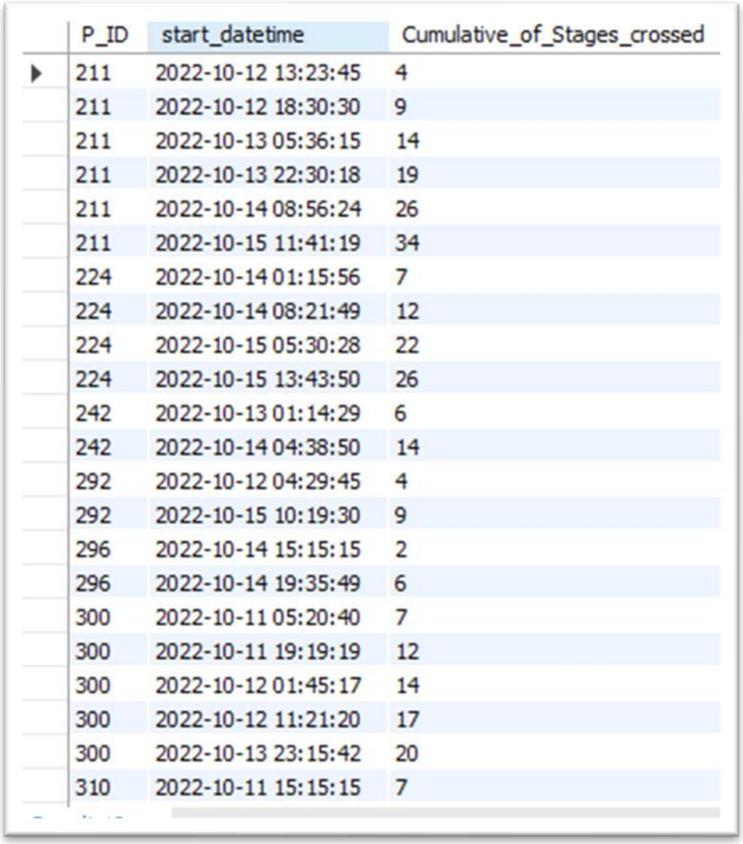
```
Select P_ID,Date(Start_datetime) as Datee,Sum(kill_count) as Kill_Count
From level_details
Group by P_ID,Date(Start_datetime);
```



P_ID	Datee	Kill_Count
211	2022-10-12	45
211	2022-10-13	44
211	2022-10-14	9
211	2022-10-15	15
224	2022-10-14	54
224	2022-10-15	58
242	2022-10-13	21
242	2022-10-14	37
292	2022-10-12	21
292	2022-10-15	4
296	2022-10-14	11
300	2022-10-11	48
300	2022-10-12	18
300	2022-10-13	8
310	2022-10-11	20
310	2022-10-13	34
310	2022-10-15	14
319	2022-10-12	5
358	2022-10-14	7
368	2022-10-12	49
368	2022-10-15	24
428	2022-10-15	5

12) Find the cumulative sum of stages crossed over a start_datetime for each `P_ID`

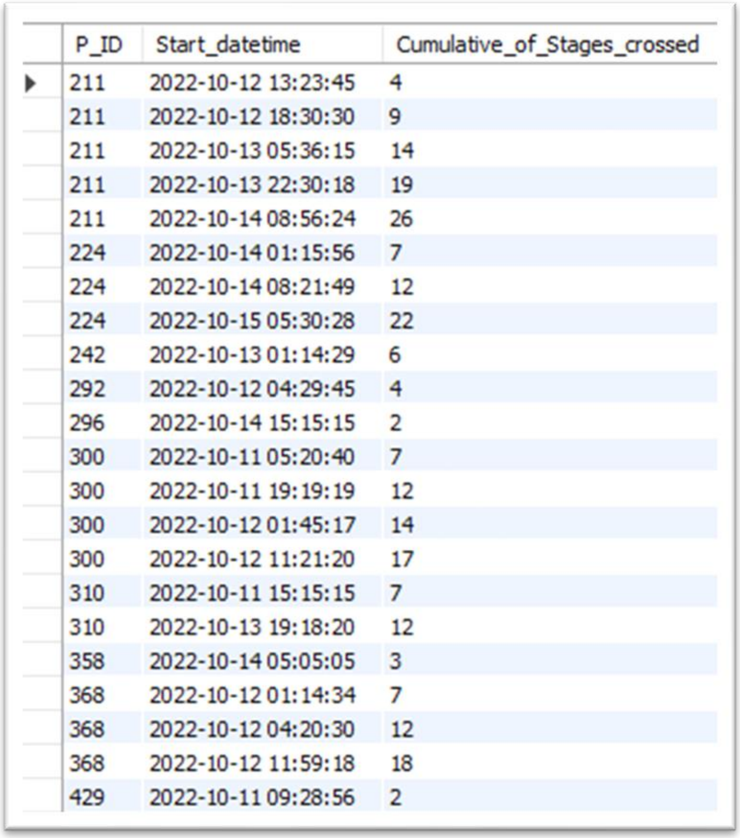
```
Select P_ID,start_datetime, sum(stages_crossed) Over(partition by P_ID Order by Start_datetime) as Cumulative_of_Stages_crossed

From level_details;
```



P_ID	start_datetime	Cumulative_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
211	2022-10-15 11:41:19	34
224	2022-10-14 01:15:56	7
224	2022-10-14 08:21:49	12
224	2022-10-15 05:30:28	22
224	2022-10-15 13:43:50	26
242	2022-10-13 01:14:29	6
242	2022-10-14 04:38:50	14
292	2022-10-12 04:29:45	4
292	2022-10-15 10:19:30	9
296	2022-10-14 15:15:15	2
296	2022-10-14 19:35:49	6
300	2022-10-11 05:20:40	7
300	2022-10-11 19:19:19	12
300	2022-10-12 01:45:17	14
300	2022-10-12 11:21:20	17
300	2022-10-13 23:15:42	20
310	2022-10-11 15:15:15	7

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

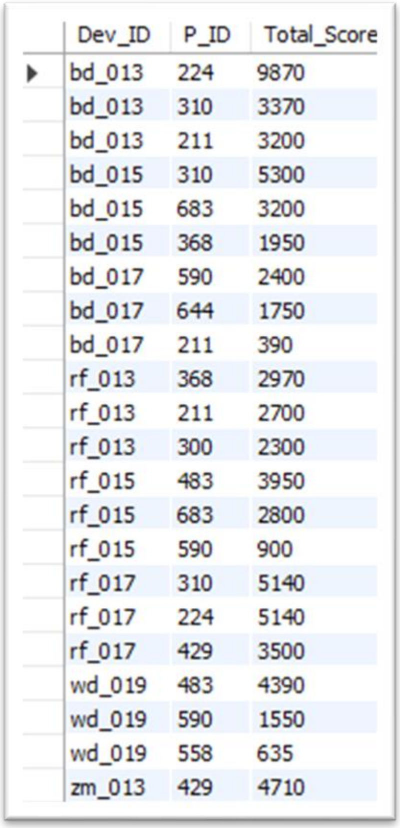
```
Select P_ID,Start_datetime,Cumulative_of_Stages_crossed

From ( Select P_ID,start_datetime,

          sum(stages_crossed) Over(partition by P_ID Order by

          Start_datetime) as Cumulative_of_Stages_crossed,
          Row_Number() Over(Partition by P_ID) as Row_NO
         From level_details
         ) as Mentorness
Where (P_ID,Row_NO)
           NOT IN
          (Select P_ID,MAX(Row_No) as Row_No
           From (Select P_ID,start_datetime,
                    sum(stages_crossed) Over(partition by P_ID Order
                    by Start_datetime) as Cumulative_of_Stages_crossed,
                      Row_Number() Over(Partition by P_ID) as Row_NO
                      From level_details) as Project
                      Group by P_ID);
```



P_ID	Start_datetime	Cumulative_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
224	2022-10-14 08:21:49	12
224	2022-10-15 05:30:28	22
242	2022-10-13 01:14:29	6
292	2022-10-12 04:29:45	4
296	2022-10-14 15:15:15	2
300	2022-10-11 05:20:40	7
300	2022-10-11 19:19:19	12
300	2022-10-12 01:45:17	14
300	2022-10-12 11:21:20	17
310	2022-10-11 15:15:15	7
310	2022-10-13 19:18:20	12
358	2022-10-14 05:05:05	3
368	2022-10-12 01:14:34	7
368	2022-10-12 04:20:30	12
368	2022-10-12 11:59:18	18
429	2022-10-11 09:28:56	2

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

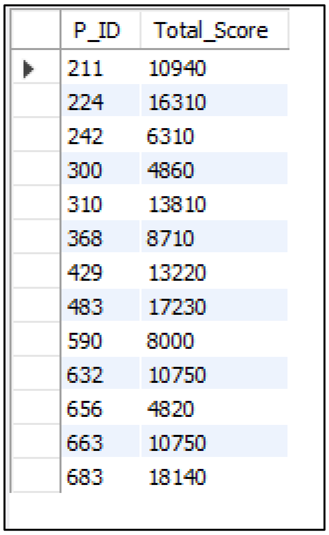
```
Select Dev_ID,P_ID,Total_Score
From  (Select Dev_ID,P_ID,Total_Score,Row_Number() Over(Partition by
Dev_ID order by Total_Score desc) as Rankk
            From(
                            Select Dev_ID,P_ID,Sum(Score) as Total_Score
           From level_details
           Group by Dev_ID,P_ID
           Order by Dev_ID asc,Total_Score desc
           ) as Project) as Mentorness
Where Rankk<=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
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

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

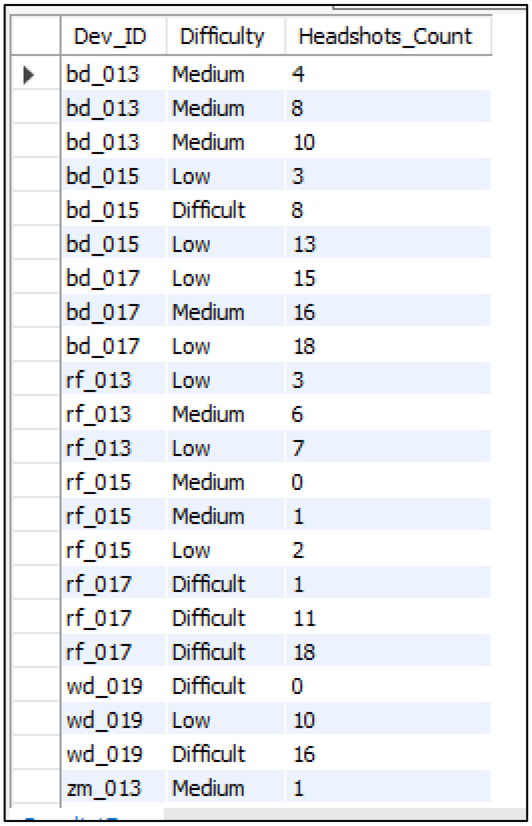
```
Select *

From (Select P_ID, Sum(Score) as Total_Score From level_details Group by P_ID) as Mentorness

Where Total_Score>(Select 0.5*Avg(Total_Score) As Avg_Score From ( Select P_ID, Sum(Score) as Total_Score From level_details Group by P_ID) as Project);
```



P_ID	Total_Score
211	10940
224	16310
242	6310
300	4860
310	13810
368	8710
429	13220
483	17230
590	8000
632	10750
656	4820
663	10750
683	18140

16) 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.

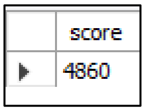
```
Delimiter //
Create Procedure TopN(IN P_TopN Int)
Begin

Select Dev_ID,Difficulty,Headshots_Count
From
        (
        Select Dev_ID,Difficulty,Headshots_Count,Row_Number() Over(Partition by Dev_ID Order by Headshots_Count) as Rankk
        From level_details
   ) as Project
Where Rankk<=P_TopN;

End //

Call TopN(3);
```



Dev_ID	Difficulty	Headshots_Count
bd_013	Medium	4
bd_013	Medium	8
bd_013	Medium	10
bd_015	Low	3
bd_015	Difficult	8
bd_015	Low	13
bd_017	Low	15
bd_017	Medium	16
bd_017	Low	18
rf_013	Low	3
rf_013	Medium	6
rf_013	Low	7
rf_015	Medium	0
rf_015	Medium	1
rf_015	Low	2
rf_017	Difficult	1
rf_017	Difficult	11
rf_017	Difficult	18
wd_019	Difficult	0
wd_019	Low	10
wd_019	Difficult	16
zm_013	Medium	1

17) Create a function to return sum of Score for a given player_id.

```
Delimiter //

Create Function Get_Score(F_ID Int)

Returns Int

Deterministic

Begin

Return

        (Select Sum(Score) From level_details Where P_ID=F_ID);

End //

Select Get_Score(300) as score;
```



score
4860