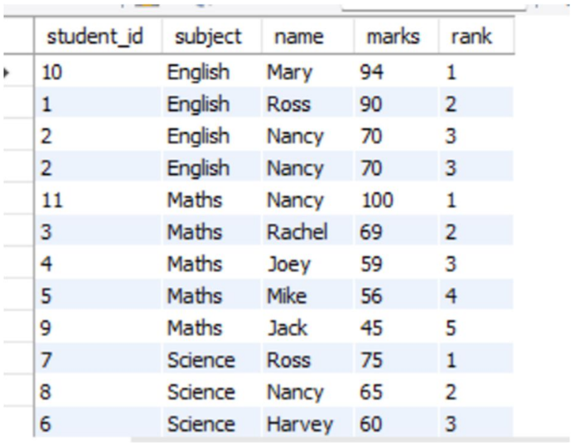
WINDOWS FUNCTION

Question 1:

Rank the students on the basis of their marks subject-wise.

Output



student_id	subject	name	marks	rank
10	English	Mary	94	1
1	English	Ross	90	2
2	English	Nancy	70	3
2	English	Nancy	70	3
11	Maths	Nancy	100	1
3	Maths	Rachel	69	2
4	Maths	Joey	59	3
5	Maths	Mike	56	4
9	Maths	Jack	45	5
7	Science	Ross	75	1
8	Science	Nancy	65	2
6	Science	Harvey	60	3

Query

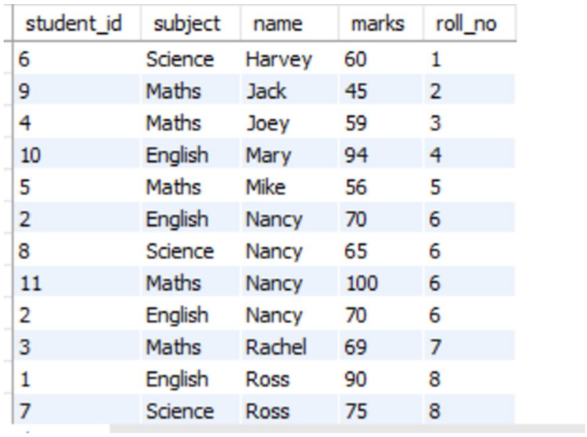
select *,dense_rank() over(partition by subject order by marks DESC) as "rank" from student

---

Question 2

Provide the new roll numbers to the students on the basis of their names alphabetically.

Output



student_id	subject	name	marks	roll_no
6	Science	Harvey	60	1
9	Maths	Jack	45	2
4	Maths	Joey	59	3
10	English	Mary	94	4
5	Maths	Mike	56	5
2	English	Nancy	70	6
8	Science	Nancy	65	6
11	Maths	Nancy	100	6
2	English	Nancy	70	6
3	Maths	Rachel	69	7
1	English	Ross	90	8
7	Science	Ross	75	8

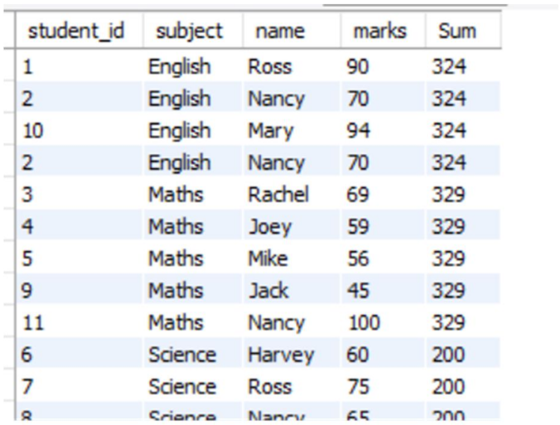
Query:

select *,dense_rank() over(order by name ASC) as "roll_no" from student

Question 3

Use the aggregate window functions to display the sum of marks in each row within its partition (Subject).

Output



student_id	subject	name	marks	Sum
1	English	Ross	90	324
2	English	Nancy	70	324
10	English	Mary	94	324
2	English	Nancy	70	324
3	Maths	Rachel	69	329
4	Maths	Joey	59	329
5	Maths	Mike	56	329
9	Maths	Jack	45	329
11	Maths	Nancy	100	329
6	Science	Harvey	60	200
7	Science	Ross	75	200
8	Science	Nancy	65	200

Query

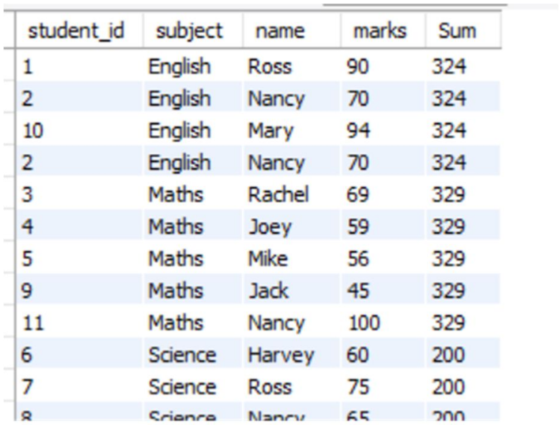
select *,sum(marks) over(partition by subject) as "Sum" from student

select distinct subject , sum(marks) over(partition by subject) as "Sum" from student

Question 4

Display the records from the students table where partition should be done on subjects and use sum as a window function on marks

Output



student_id	subject	name	marks	Sum
1	English	Ross	90	324
2	English	Nancy	70	324
10	English	Mary	94	324
2	English	Nancy	70	324
3	Maths	Rachel	69	329
4	Maths	Joey	59	329
5	Maths	Mike	56	329
9	Maths	Jack	45	329
11	Maths	Nancy	100	329
6	Science	Harvey	60	200
7	Science	Ross	75	200
8	Science	Nancy	65	200

Query

```
select *,sum(marks) over(partition by subject ) as "Sum" from student
```

```
select distinct subject , sum(marks) over(partition by subject ) as "Sum" from student
```

Question 5

Find the dense rank of the students on the basis of their marks subjectwise.
Store the result in a new view with name as 'Students_Ranked'

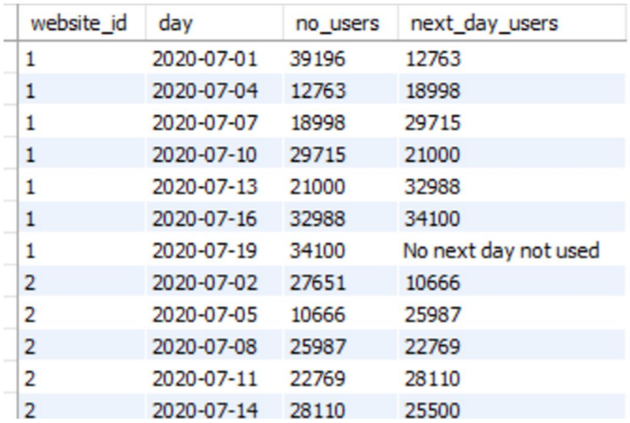
Query

```
CREATE OR REPLACE VIEW Students_Ranked as

select *,dense_rank() over(partition by subject order by marks DESC) as "rank" from student
```

Question 6

Show day, number of users and the number of users the next day (for all days when the website was used)

Output



website_id	day	no_users	next_day_users
1	2020-07-01	39196	12763
1	2020-07-04	12763	18998
1	2020-07-07	18998	29715
1	2020-07-10	29715	21000
1	2020-07-13	21000	32988
1	2020-07-16	32988	34100
1	2020-07-19	34100	No next day not used
2	2020-07-02	27651	10666
2	2020-07-05	10666	25987
2	2020-07-08	25987	22769
2	2020-07-11	22769	28110
2	2020-07-14	28110	25500

Query

select website_id,day , no_users ,lead(no_users,1,"No next day not used")

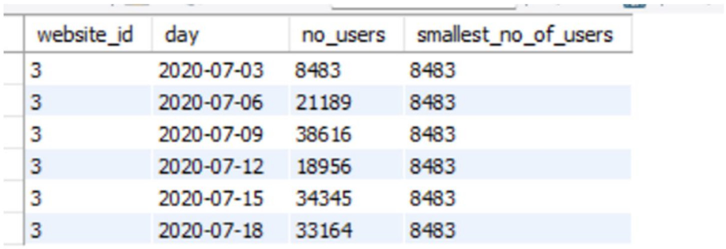
over (partition by website_id order by day asc) as "next_day_users"

from website_stats where no_users > 0

Question 7

Write a query that displays the statistics for website_id = 3 such that for each row, show the day, the number of users and the smallest number of users ever.

Output



website_id	day	no_users	smallest_no_of_users
3	2020-07-03	8483	8483
3	2020-07-06	21189	8483
3	2020-07-09	38616	8483
3	2020-07-12	18956	8483
3	2020-07-15	34345	8483
3	2020-07-18	33164	8483

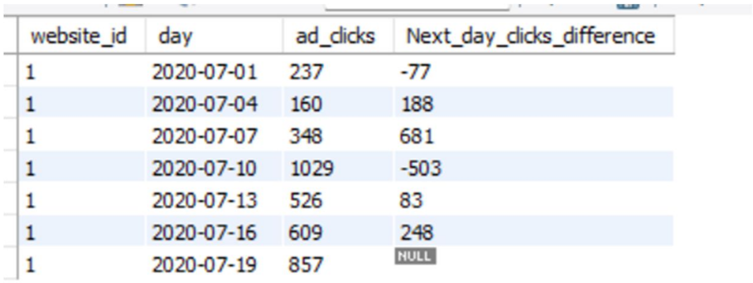
Query

```
select website_id,day,no_users , min(no_users) over (partition by website_id) as "smallest_no_of_users"

from website_stats where website_id = 3
```

Question 9

Display the difference in ad_clicks between the current day and the next day for the website 'Olympus'

Output



website_id	day	ad_clicks	Next_day_clicks_difference
1	2020-07-01	237	-77
1	2020-07-04	160	188
1	2020-07-07	348	681
1	2020-07-10	1029	-503
1	2020-07-13	526	83
1	2020-07-16	609	248
1	2020-07-19	857	NULL

Query

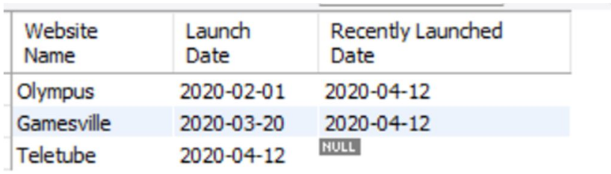
```
select a.website_id , a.day , a.ad_clicks ,

 CASE

       WHEN LEAD(a.ad_clicks) OVER (PARTITION BY a.website_id ORDER BY a.day) IS NULL THEN NULL

       ELSE LEAD(a.ad_clicks) OVER (PARTITION BY a.website_id ORDER BY a.day) - a.ad_clicks

   END AS "Next_day_clicks_difference"

from website_stats a where website_id=(select id from website where name = "Olympus")
```

Question 8

Write a query that displays name of the website and it's launch date. The query should also display the date of recently launched website in the third column.

Query

```
SELECT
   w1.name AS "Website Name",
   w1.launch_date AS "Launch Date",
   MAX(w2.launch_date) AS "Recently Launched Date"
FROM
   website w1
LEFT JOIN
   website w2 ON w1.la
unch_date < w2.launch_date
GROUP BY
   w1.name, w1.launch_date;
```



Website Name	Launch Date	Recently Launched Date
Olympus	2020-02-01	2020-04-12
Gamesville	2020-03-20	2020-04-12
Teletube	2020-04-12	NULL

Question 10

Remove duplication from student

Query

```
SELECT *,
    ROW_NUMBER() OVER (PARTITION BY subject ORDER BY marks DESC) AS "row_number",
    DENSE_RANK() OVER (PARTITION BY subject ORDER BY marks DESC) AS "dense_rank"
FROM student;
```

*After this process don’t know how to delete the records