#Run the below commands

CREATE TABLE student (

student\_id INT,

subject VARCHAR(50),

name VARCHAR(50),

marks INT

);

-- Insert the data

INSERT INTO student (student\_id, subject, name, marks)

VALUES

(1, 'English', 'Ross', 90),

(2, 'English', 'Nancy', 70),

(3, 'Maths', 'Rachel', 69),

(4, 'Maths', 'Joey', 59),

(5, 'Maths', 'Mike', 56),

(6, 'Science', 'Harvey', 60),

(7, 'Science', 'Ross', 75),

(8, 'Science', 'Nancy', 65),

(9, 'Maths', 'Jack', 45),

(10, 'English', 'Mary', 94),

(11, 'Maths', 'Nancy', 100),

(2, 'English', 'Nancy', 70);

-- Create the table

CREATE TABLE website(

id INT PRIMARY KEY,

name VARCHAR(50),

budget DECIMAL(10, 2),

launch\_date DATE

);

-- Insert the data

INSERT INTO website(id, name, budget, launch\_date)

VALUES

(1, 'Olympus', 2800.00, '2020-02-01'),

(2, 'Gamesville', 900.00, '2020-03-20'),

(3, 'Teletube', 600.00, '2020-04-12');

-- Create the table

CREATE TABLE website\_stats (

website\_id INT,

day DATE,

no\_users INT,

ad\_impressions INT,

ad\_clicks INT,

income DECIMAL(10, 2)

);

-- Insert the data

INSERT INTO website\_stats (website\_id, day, no\_users, ad\_impressions, ad\_clicks, income)

VALUES

(1, '2020-07-01', 39196, 109834, 237, 76.17),

(2, '2020-07-02', 27651, 287612, 793, 243.11),

(3, '2020-07-03', 8483, 501099, 1545, 410.99),

(1, '2020-07-04', 12763, 56990, 160, 40.01),

(2, '2020-07-05', 10666, 170555, 343, 98.78),

(3, '2020-07-06', 21189, 370101, 804, 121.12),

(1, '2020-07-07', 18998, 145783, 348, 134.91),

(2, '2020-07-08', 25987, 295633, 782, 144.71),

(3, '2020-07-09', 38616, 117825, 342, 95.6),

(1, '2020-07-10', 29715, 320101, 1029, 189.88),

(2, '2020-07-11', 22769, 143320, 423, 76.5),

(3, '2020-07-12', 18956, 168123, 378, 112.12),

(1, '2020-07-13', 21000, 178423, 526, 98.81),

(2, '2020-07-14', 28110, 172505, 407, 89.13),

(3, '2020-07-15', 34345, 57924, 135, 33.75),

(1, '2020-07-16', 32988, 311300, 609, 121.73),

(2, '2020-07-17', 25500, 187530, 383, 57.47),

(3, '2020-07-18', 33164, 236598, 594, 118.75),

(1, '2020-07-19', 34100, 345678, 857, 197.08),

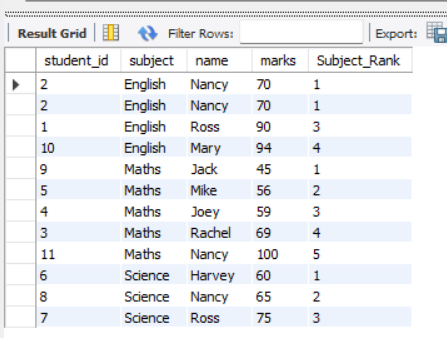
(2, '2020-07-20', 10799, 80012, 229, 52.58);

1) Rank the students on the basis of their marks subjectwise.

**CODE:**

SELECT \*,Rank() OVER (PARTITION BY subject ORDER BY marks ) AS Subject\_Rank from student;

**OUTPUT:**

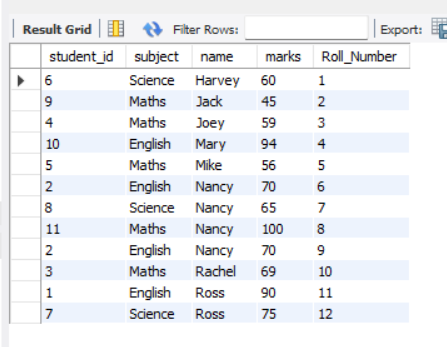
****

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

**CODE:**

SELECT\*,ROW\_NUMBER() OVER( ORDER BY name) AS Roll\_Number from student;

**OUTPUT:**

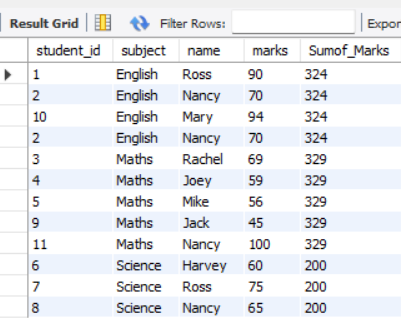


3) Use the aggregate window functions to display the sum of marks in each row within its partition (Subject).

**CODE:**

SELECT \*, SUM(Marks) OVER (PARTITION BY Subject) AS Sumof\_Marks FROM Student;

**OUTPUT:**

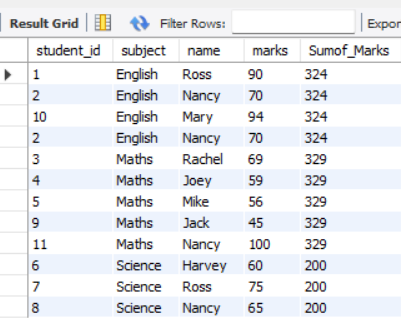


4) Display the records from the students table where partition should be done on subjects and use sum as a window function on marks

**CODE:**

SELECT \*, SUM(Marks) OVER (PARTITION BY Subject) AS Sumof\_Marks FROM Student;

**OUTPUT:**



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'

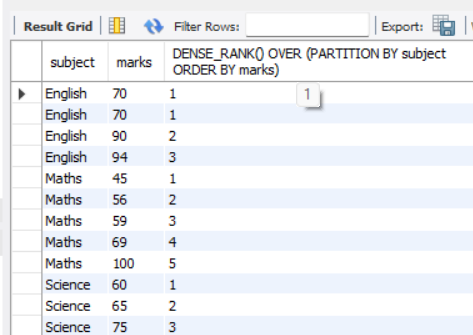
**CODE:**

CREATE VIEW Students\_Ranked AS

SELECT subject, marks, DENSE\_RANK() OVER (PARTITION BY subject ORDER BY marks) FROM Student ;

SELECT\*from Students\_Ranked;

**OUTPUT:**

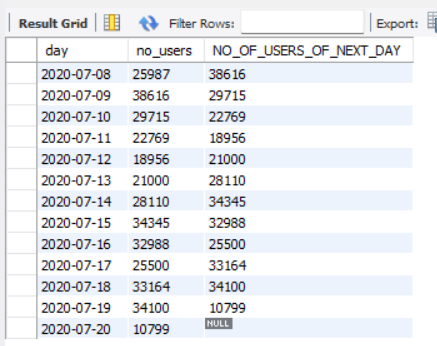


6)Show day, number of users and the number of users the next day (for all days when the website was used)

**CODE:**

SELECT day,no\_users,LEAD(no\_users) OVER( ORDER BY day) AS NO\_OF\_USERS\_OF\_NEXT\_DAY FROM Website\_stats;

**OUTPUT:**



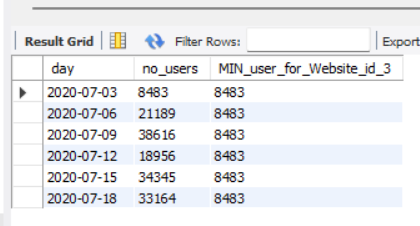
7)Write a query that displays the statistics for website\_id = 3 such that for each row, show the day, the number of users and the smallest number of users ever.

**CODE:**

SELECT day, no\_users, MIN(no\_users) OVER ( PARTITION BY Website\_id) AS MIN\_user\_for\_Website\_id\_3 FROM Website\_stats

WHERE Website\_id = 3;

**OUTPUT:**



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.

**CODE:**

SELECT

w.name AS website\_name,

w.launch\_date,

recent.launch\_date AS recently\_launched\_date

FROM

website w

JOIN (

SELECT

MAX(launch\_date) AS launch\_date

FROM

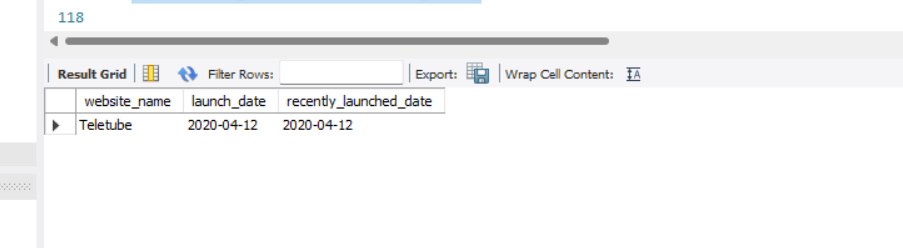
website

) AS recent

ON

w.launch\_date = recent.launch\_date;

**OUTPUT:**

****

9) Display the difference in ad\_clicks between the current day and the next day for the website 'Olympus'

**CODE:**

SELECT

ws1.day AS current\_day,

ws1.ad\_clicks AS current\_day\_ad\_clicks,

ws2.day AS next\_day,

ws2.ad\_clicks AS next\_day\_ad\_clicks,

ws2.ad\_clicks - ws1.ad\_clicks AS ad\_clicks\_difference

FROM

website\_stats ws1

JOIN

website\_stats ws2 ON ws1.day = DATE\_ADD(ws2.day, INTERVAL 1 DAY)

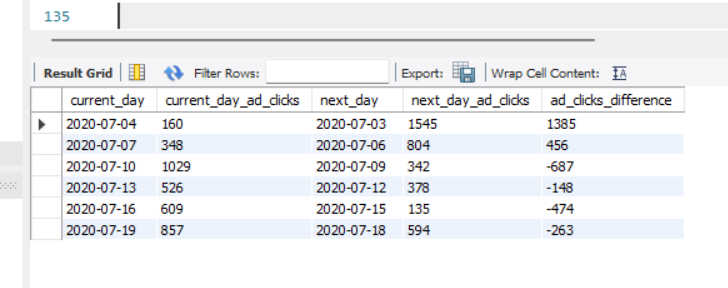
JOIN

website w ON ws1.website\_id = w.id

WHERE

w.name = 'Olympus';

**OUTPUT:**



10) Identify and remove the duplicate data from the student table.

**CODE:**

WITH DuplicateCTE AS (

SELECT

student\_id,

subject,

name,

marks,

ROW\_NUMBER() OVER (PARTITION BY student\_id, subject, name, marks ORDER BY student\_id) AS row\_num

FROM

student

)

## To Delete the duplicate records

DELETE FROM student

WHERE (student\_id, subject, name, marks) IN (

SELECT student\_id, subject, name, marks

FROM DuplicateCTE

WHERE row\_num > 1 );