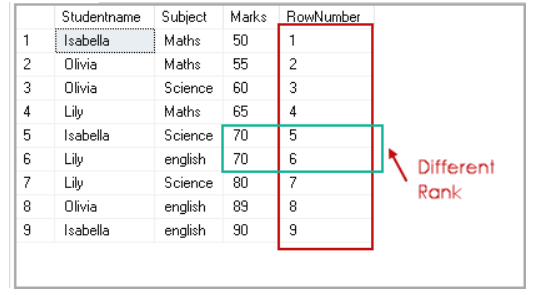
* ROW\_NUMBER()
* RANK()
* DENSE\_RANK()
* NTILE()

ROW\_NUMBER()

We use ROW\_Number() SQL RANK function to get a unique sequential number for each row in the specified data. It gives the rank one for the first row and then increments the value by one for each row. We get different ranks for the row having similar values as well.

SELECT Studentname, Subject, Marks, ROW\_NUMBER() OVER(ORDER BY Marks) RowNumber FROM ExamResult;



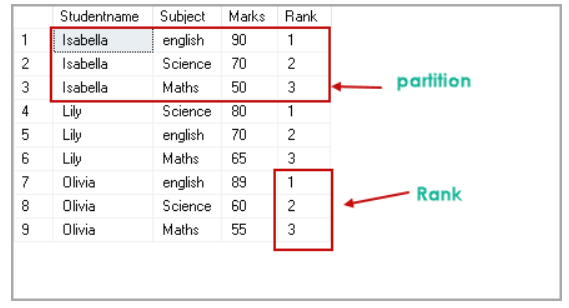
RANK()

We use RANK() SQL Rank function to specify rank for each row in the result set. We have student results for three subjects. We want to rank the result of students as per their marks in the subjects.

Execute the following query to get this result set. In this query, you can note the following things:

* We use PARTITION**BY Studentname** clause to perform calculations on each student group
* Each subset should get rank as per their Marks in descending order
* The result set uses Order By clause to sort results on Studentname and their rank

SELECT Studentname, Subject, Marks, RANK() OVER(PARTITION BY Studentname ORDER BY Marks DESC) Rank FROM ExamResult ORDER BY Studentname, Rank;

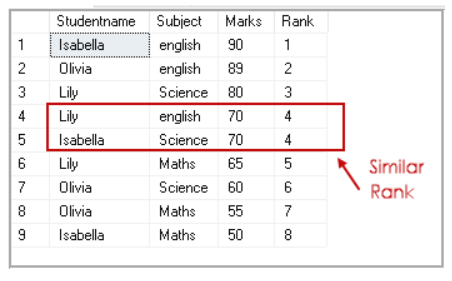


DENSE\_RANK()

We use DENSE\_RANK() function to specify a unique rank number within the partition as per the specified column value.

In the SQL RANK function DENSE\_RANK(), if we have duplicate values, SQL assigns different ranks to those rows as well.

SELECT Studentname, Subject, Marks, DENSE\_RANK() OVER(ORDER BY Marks DESC) Rank FROM ExamResult ORDER BY Rank;



NTILE(N)

We use the NTILE(N) function to distribute the number of rows in the specified (N) number of groups. Each row group gets its rank as per the specified condition. We need to specify the value for the desired number of groups.

SELECT \*, NTILE(2) OVER(ORDER BY Marks DESC) Rank FROM ExamResult ORDER BY rank;

