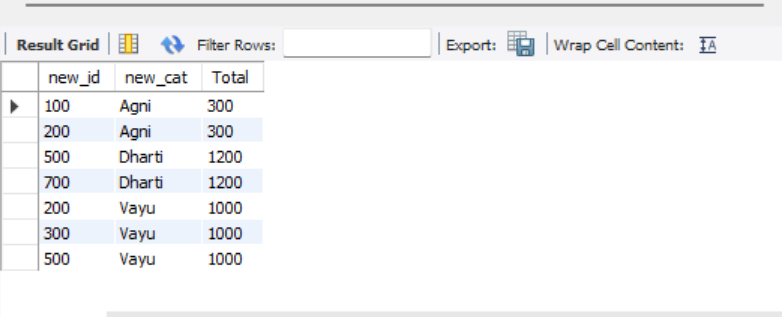
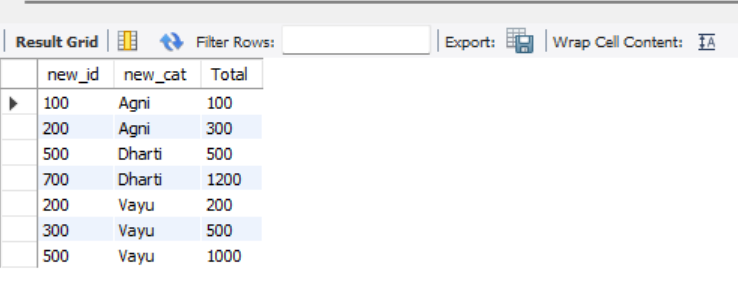
**WINDOW FUNCTIONS**

* create database new;
* use new;

1. select new\_id,new\_cat,Sum(new\_id) Over (partition by new\_cat) As "Total" from newid;



1. select new\_id,new\_cat,Sum(new\_id) Over (partition by new\_cat order by new\_id) As "Total" from newid;



1. select new\_id,new\_cat,

sum(new\_id) OVER (PARTITION BY new\_cat ORDER BY new\_id) AS "TOTAL",

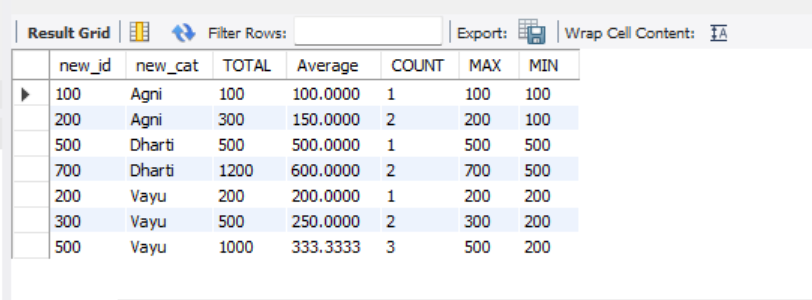
AVG (new\_id) OVER (PARTITION BY new\_cat ORDER BY new\_id) AS "Average" ,

count(new\_id) OVER (PARTITION BY new\_cat ORDER BY new\_id) AS "COUNT",

max(new\_id) OVER (PARTITION BY new\_cat ORDER BY new\_id) AS "MAX",

min(new\_id) OVER (PARTITION BY new\_cat ORDER BY new\_id) AS "MIN"

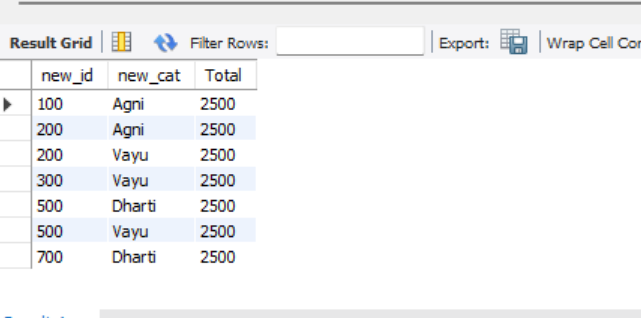
from newid;



1. SELECT new\_id,new\_cat,

sum(new\_id)over(order by new\_id rows between unbounded preceding and unbounded following)

as "Total" from newid;



1. select new\_id, row\_number() over(order by new\_id)

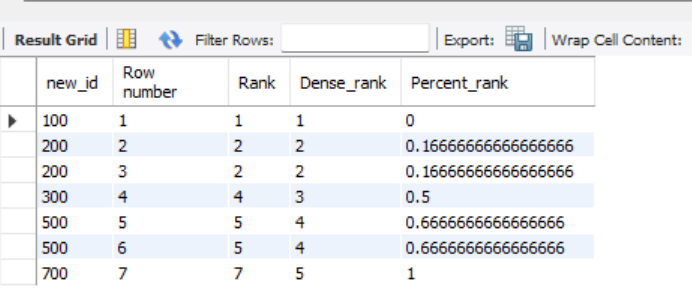
as "Row number",

rank() over(order by new\_id ) as "Rank",

dense\_rank() over(order by new\_id) as "Dense\_rank",

percent\_rank() over(order by new\_id) as "Percent\_rank"

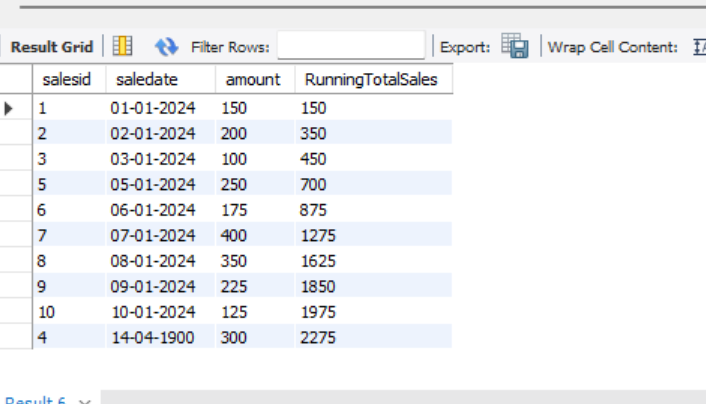
from newid;



1. Calculate the running total of sales amount

SELECT salesid,saledate,amount,SUM(amount) OVER (ORDER BY saledate) AS "RunningTotalSales"

FROM customer;



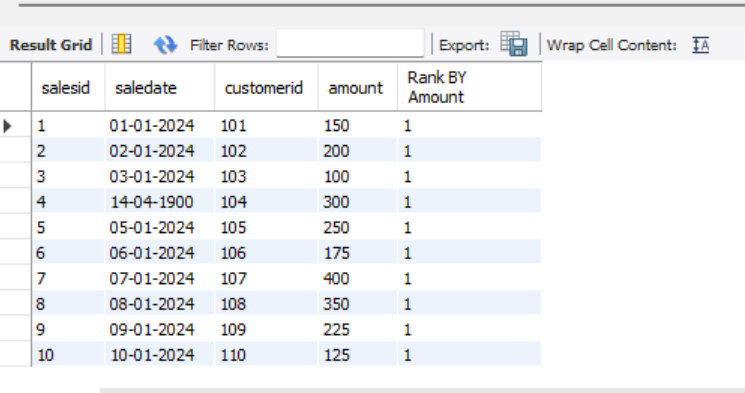
1. Rank the sales by amount for each customer.

SELECT salesid, saledate,customerid,amount,

rank() over(partition by customerid order by amount DESc)

AS "Rank BY Amount"

FROM customer;



1. Calculate avg sales amount over the last 3 sales.

Select

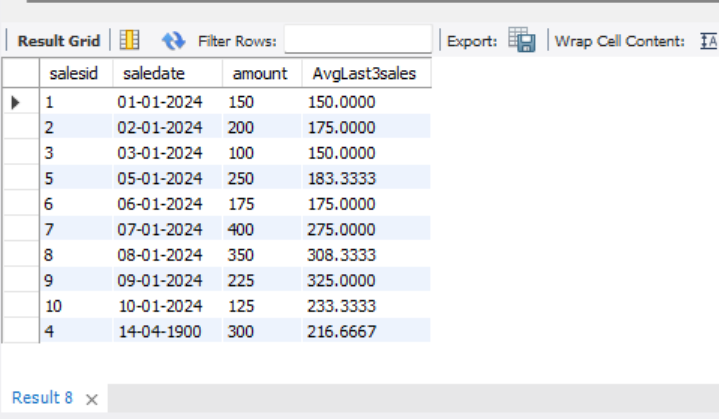
salesid,saledate,

amount,

avg(amount) over(order by saledate Rows between 2 preceding and current row)

As "AvgLast3sales"

FROM customer;



1. Calculate the cumulative distribution sales amount.

SELECT

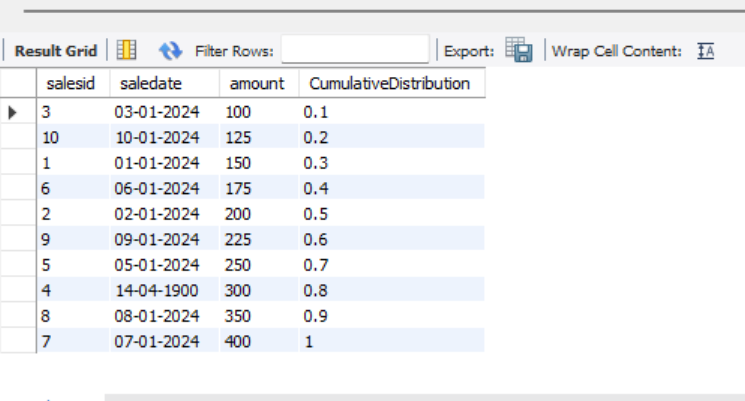
salesid,

saledate,

amount,

CUME\_DIST() OVER (ORDER BY Amount) AS CumulativeDistribution

FROM customer;



1. Calculate the lead sales amount for the next sale.

SELECT

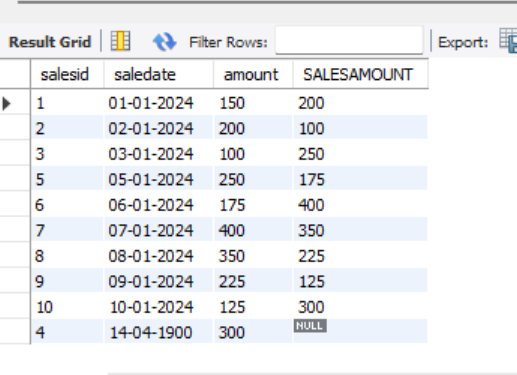
salesid,

saledate,

amount,

LEAD(amount,1) OVER (ORDER BY saledate) AS SALESAMOUNT

FROM customer;



SELECT

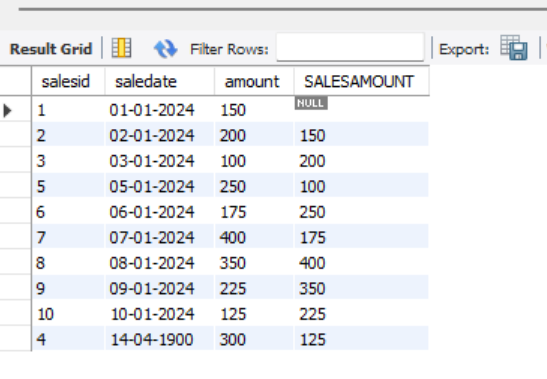
salesid,

saledate,

amount,

Lag(amount,1) OVER (ORDER BY saledate) AS SALESAMOUNT

FROM customer;



1. Find the first sale amount fir each customer.

select salesid,

saledate,

customerid,

amount,

first\_value(amount) over (partition by customerid order by saledate)

as "FirstSaleAmount"

from customer;

