**MySQL WINDOW FUNCTIONS**

CREATE TABLE Sales (

SaleID INT,

SaleDate DATE,

CustomerID INT,

ProductID INT,

Amount DECIMAL(10, 2)

);

INSERT INTO Sales (SaleID, SaleDate, CustomerID, ProductID, Amount) VALUES

(1, '2024-01-01', 101, 1001, 150.00),

(2, '2024-01-02', 102, 1002, 200.00),

(3, '2024-01-03', 101, 1001, 100.00),

(4, '2024-01-04', 103, 1003, 300.00),

(5, '2024-01-05', 102, 1002, 250.00),

(6, '2024-01-06', 101, 1001, 175.00),

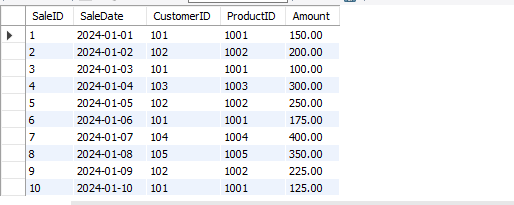
(7, '2024-01-07', 104, 1004, 400.00),

(8, '2024-01-08', 105, 1005, 350.00),

(9, '2024-01-09', 102, 1002, 225.00),

(10, '2024-01-10', 101, 1001, 125.00);

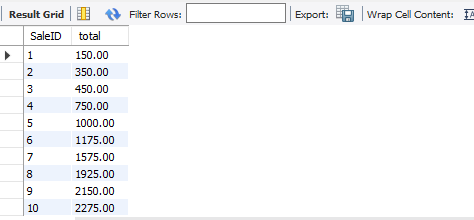
select \* from sales;



**Question 1: Calculate the running total of sales amount.**

**Solution :**

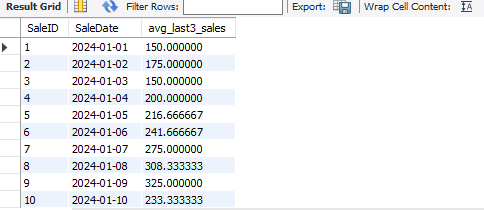
select SaleID, sum(amount)over(order by SaleDate ) as "total" from sales;



**Question 2: Calculate the average sales amount over the last 3 sales.**

**Solution :**

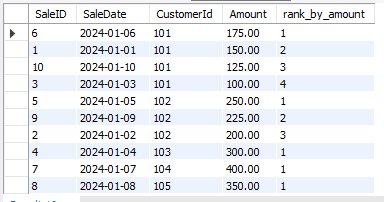
select SaleID,SaleDate, avg(amount)over(order by SaleDate rows between 2 preceding and current row ) as "avg\_last3\_sales" from sales;



**Question 3: Rank the sales by amount for each customer.**

**Solution :**

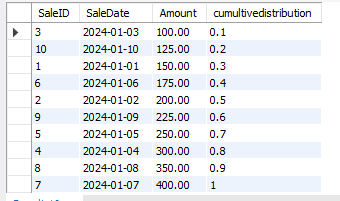
select SaleID,SaleDate,CustomerId,Amount,rank() over(partition by CustomerID order by Amount desc) as "rank\_by\_amount" from sales;



**Question 4: Calculate the cumulative distribution of sales amount.**

**Solution :**

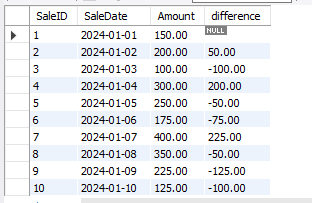
select SaleID, SaleDate, Amount, CUME\_DIST() OVER (ORDER BY AMOUNT) AS "cumultivedistribution" from Sales;



**Question 5: Calculate the difference in sales amount between the current sale and the previous sale.**

**Solution :**

select SaleID, SaleDate, Amount,Amount-lag(Amount,1)over(order by SaleDate) as "difference" from Sales;



**Question 6: Calculate the lead sales amount for the next sale.**

**Solution :**

SELECT

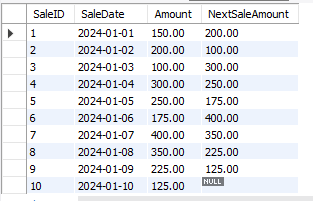
SaleID,

SaleDate,

Amount,

LEAD(Amount, 1) OVER (ORDER BY SaleDate) AS NextSaleAmount

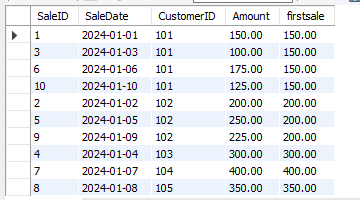
FROM Sales;



**Question 7: Find the first sale amount for each customer.**

**Solution :**

select SaleID, SaleDate,CustomerID, Amount, first\_value(Amount) over(partition by CustomerID order by SaleDate) as "firstsale" from SAles;



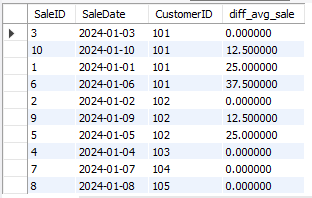
**Q8. Find the difference between each sale amount and the average sale amount.**

**Solution :**

select SaleID,

SaleDate,

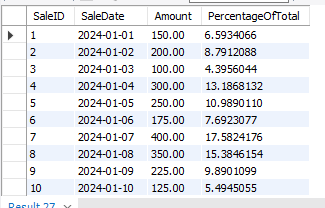
CustomerID,amount-avg(amount)over(partition by customerID order by amount) as "diff\_avg\_sale" from Sales;



**Question 9: Find the percentage that each sale amount contributes to the total sales**.

**Solution :**

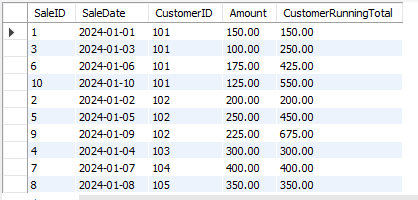
SELECT SaleID,SaleDate,Amount,Amount \* 100.0 / SUM(Amount) OVER () AS PercentageOfTotal FROM Sales;



**Question 10: For each customer, calculate the total sales up to the current saledate (cumulative total per customer).**

**Solution :**

SELECT SaleID,SaleDate,CustomerID,Amount,SUM(Amount) OVER (PARTITION BY CustomerID ORDER BY SaleDate) AS CustomerRunningTotal FROM Sales;



**Question 11: For each product, find the latest sale amount.**

**Solution :**

SELECT SaleID,SaleDate,ProductID,Amount,

LAST\_VALUE(Amount) OVER (PARTITION BY ProductID ORDER BY SaleDate ROWS

BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS LatestSaleAmount FROM Sales;

