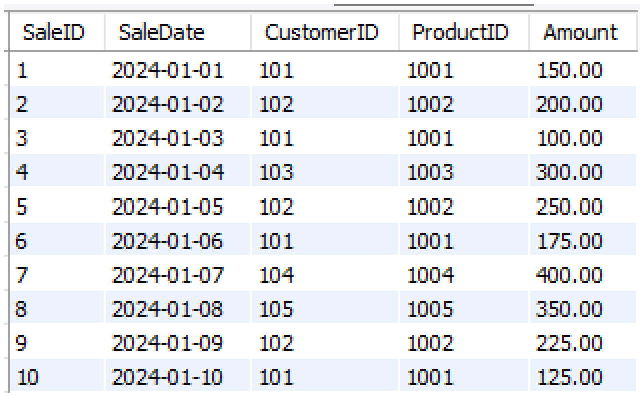
MySQL WINDOW FUNCTIONS

```
USE company;

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
```

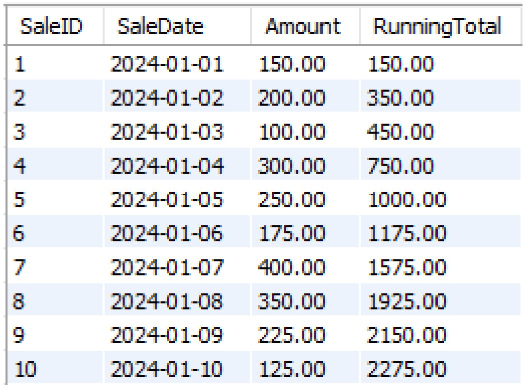


SaleID	SaleDate	CustomerID	ProductID	Amount
1	2024-01-01	101	1001	150.00
2	2024-01-02	102	1002	200.00
3	2024-01-03	101	1001	100.00
4	2024-01-04	103	1003	300.00
5	2024-01-05	102	1002	250.00
6	2024-01-06	101	1001	175.00
7	2024-01-07	104	1004	400.00
8	2024-01-08	105	1005	350.00
9	2024-01-09	102	1002	225.00
10	2024-01-10	101	1001	125.00

Question 1: Calculate the running total of sales amount.

Solution 1:

```
SELECT
    SaleID,
    SaleDate,
    Amount,
    SUM(Amount) OVER (ORDER BY SaleDate) AS RunningTotal
FROM Sales;
```

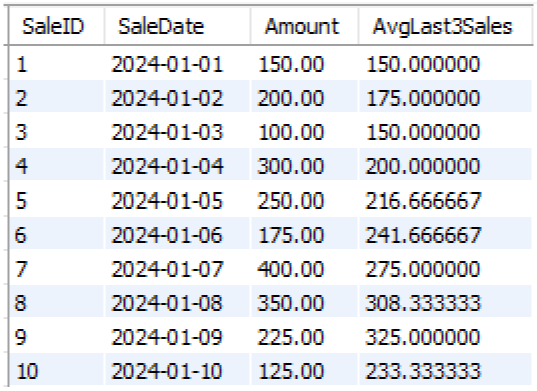


SaleID	SaleDate	Amount	RunningTotal
1	2024-01-01	150.00	150.00
2	2024-01-02	200.00	350.00
3	2024-01-03	100.00	450.00
4	2024-01-04	300.00	750.00
5	2024-01-05	250.00	1000.00
6	2024-01-06	175.00	1175.00
7	2024-01-07	400.00	1575.00
8	2024-01-08	350.00	1925.00
9	2024-01-09	225.00	2150.00
10	2024-01-10	125.00	2275.00

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

Solution 2:

```
SELECT
    SaleID,
    SaleDate,
    Amount,
    AVG(Amount) OVER (ORDER BY SaleDate ROWS BETWEEN 2 PRECEDING AND CURRENT
ROW) AS AvgLast3Sales
FROM Sales;
```

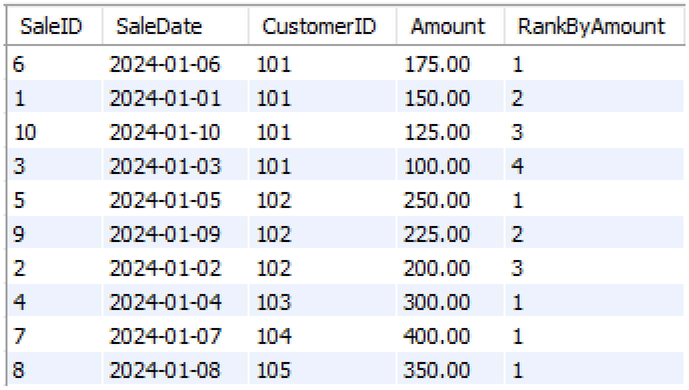


SaleID	SaleDate	Amount	AvgLast3Sales
1	2024-01-01	150.00	150.000000
2	2024-01-02	200.00	175.000000
3	2024-01-03	100.00	150.000000
4	2024-01-04	300.00	200.000000
5	2024-01-05	250.00	216.666667
6	2024-01-06	175.00	241.666667
7	2024-01-07	400.00	275.000000
8	2024-01-08	350.00	308.333333
9	2024-01-09	225.00	325.000000
10	2024-01-10	125.00	233.333333

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

Solution 3:

```
SELECT
    SaleID,
    SaleDate,
    CustomerID,
    Amount,
    RANK() OVER (PARTITION BY CustomerID ORDER BY Amount DESC) AS
RankByAmount
FROM Sales;
```

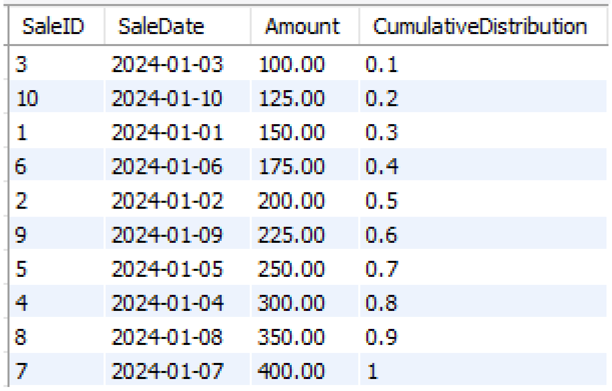


SaleID	SaleDate	CustomerID	Amount	RankByAmount
6	2024-01-06	101	175.00	1
1	2024-01-01	101	150.00	2
10	2024-01-10	101	125.00	3
3	2024-01-03	101	100.00	4
5	2024-01-05	102	250.00	1
9	2024-01-09	102	225.00	2
2	2024-01-02	102	200.00	3
4	2024-01-04	103	300.00	1
7	2024-01-07	104	400.00	1
8	2024-01-08	105	350.00	1

Question 4: Calculate the cumulative distribution of sales amount.

Solution 4:

```
SELECT
    SaleID,
    SaleDate,
    Amount,
    CUME_DIST() OVER (ORDER BY Amount) AS CumulativeDistribution
FROM Sales;
```

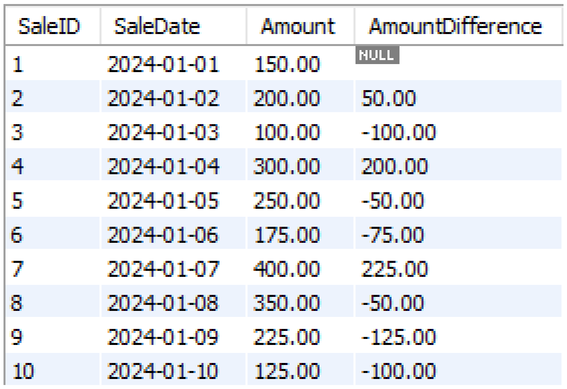


SaleID	SaleDate	Amount	CumulativeDistribution
3	2024-01-03	100.00	0.1
10	2024-01-10	125.00	0.2
1	2024-01-01	150.00	0.3
6	2024-01-06	175.00	0.4
2	2024-01-02	200.00	0.5
9	2024-01-09	225.00	0.6
5	2024-01-05	250.00	0.7
4	2024-01-04	300.00	0.8
8	2024-01-08	350.00	0.9
7	2024-01-07	400.00	1

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

Solution 5:

```
SELECT
    SaleID,
    SaleDate,
    Amount,
    Amount - LAG(Amount, 1) OVER (ORDER BY SaleDate) AS AmountDifference
FROM Sales;
```

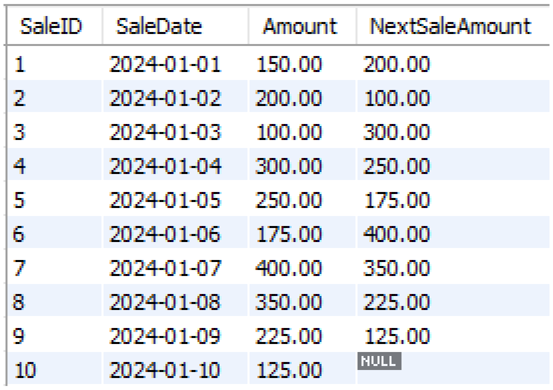


SaleID	SaleDate	Amount	AmountDifference
1	2024-01-01	150.00	NULL
2	2024-01-02	200.00	50.00
3	2024-01-03	100.00	-100.00
4	2024-01-04	300.00	200.00
5	2024-01-05	250.00	-50.00
6	2024-01-06	175.00	-75.00
7	2024-01-07	400.00	225.00
8	2024-01-08	350.00	-50.00
9	2024-01-09	225.00	-125.00
10	2024-01-10	125.00	-100.00

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

Solution 6:

```
SELECT
    SaleID,
    SaleDate,
    Amount,
    LEAD(Amount, 1) OVER (ORDER BY SaleDate) AS NextSaleAmount
FROM Sales;
```

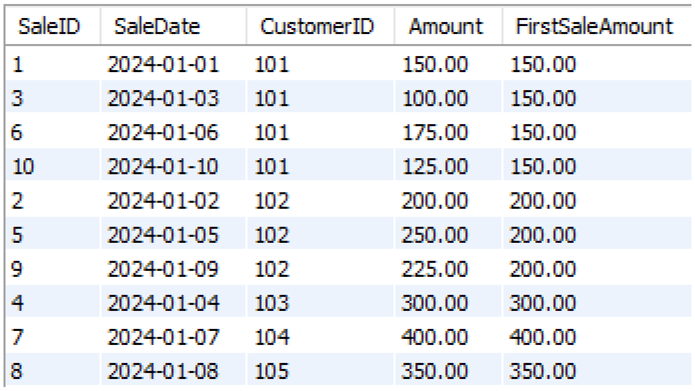


SaleID	SaleDate	Amount	NextSaleAmount
1	2024-01-01	150.00	200.00
2	2024-01-02	200.00	100.00
3	2024-01-03	100.00	300.00
4	2024-01-04	300.00	250.00
5	2024-01-05	250.00	175.00
6	2024-01-06	175.00	400.00
7	2024-01-07	400.00	350.00
8	2024-01-08	350.00	225.00
9	2024-01-09	225.00	125.00
10	2024-01-10	125.00	NULL

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

Solution 7:

```
SELECT
    SaleID,
    SaleDate,
    CustomerID,
    Amount,
    FIRST_VALUE(Amount) OVER (PARTITION BY CustomerID ORDER BY SaleDate) AS
FirstSaleAmount
FROM Sales;
```

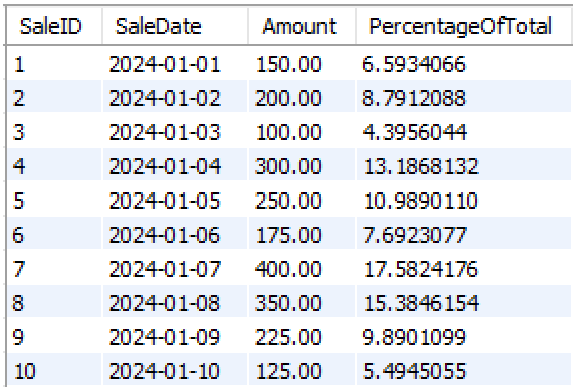


SaleID	SaleDate	CustomerID	Amount	FirstSaleAmount
1	2024-01-01	101	150.00	150.00
3	2024-01-03	101	100.00	150.00
6	2024-01-06	101	175.00	150.00
10	2024-01-10	101	125.00	150.00
2	2024-01-02	102	200.00	200.00
5	2024-01-05	102	250.00	200.00
9	2024-01-09	102	225.00	200.00
4	2024-01-04	103	300.00	300.00
7	2024-01-07	104	400.00	400.00
8	2024-01-08	105	350.00	350.00

Question 8: Find the percentage that each sale amount contributes to the total sales.

Solution 8:

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

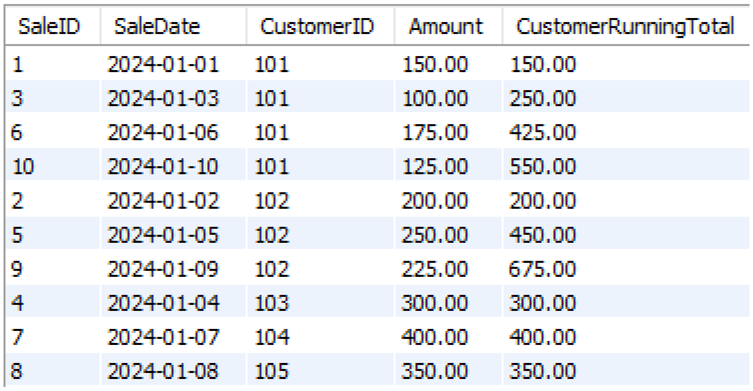


SaleID	SaleDate	Amount	PercentageOfTotal
1	2024-01-01	150.00	6.5934066
2	2024-01-02	200.00	8.7912088
3	2024-01-03	100.00	4.3956044
4	2024-01-04	300.00	13.1868132
5	2024-01-05	250.00	10.9890110
6	2024-01-06	175.00	7.6923077
7	2024-01-07	400.00	17.5824176
8	2024-01-08	350.00	15.3846154
9	2024-01-09	225.00	9.8901099
10	2024-01-10	125.00	5.4945055

Question 9: For each customer, calculate the total sales up to the current sale date (cumulative total per customer).

Solution 9:

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

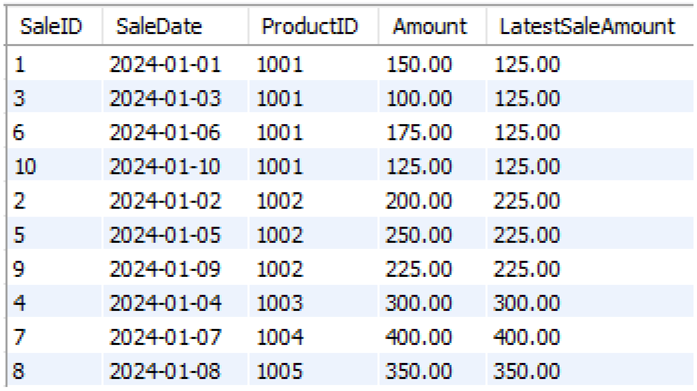


SaleID	SaleDate	CustomerID	Amount	CustomerRunningTotal
1	2024-01-01	101	150.00	150.00
3	2024-01-03	101	100.00	250.00
6	2024-01-06	101	175.00	425.00
10	2024-01-10	101	125.00	550.00
2	2024-01-02	102	200.00	200.00
5	2024-01-05	102	250.00	450.00
9	2024-01-09	102	225.00	675.00
4	2024-01-04	103	300.00	300.00
7	2024-01-07	104	400.00	400.00
8	2024-01-08	105	350.00	350.00

Question 10: For each product, find the latest sale amount.

Solution 10:

```
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;
```



SaleID	SaleDate	ProductID	Amount	LatestSaleAmount
1	2024-01-01	1001	150.00	125.00
3	2024-01-03	1001	100.00	125.00
6	2024-01-06	1001	175.00	125.00
10	2024-01-10	1001	125.00	125.00
2	2024-01-02	1002	200.00	225.00
5	2024-01-05	1002	250.00	225.00
9	2024-01-09	1002	225.00	225.00
4	2024-01-04	1003	300.00	300.00
7	2024-01-07	1004	400.00	400.00
8	2024-01-08	1005	350.00	350.00