

Level: 2nd Year Bachelor's in Computer Science (2025/2026)
MODULE: DATABASES
TD No 4: SQL

Consider the following relational schema for sales

PRODUCT (**ProductID**, Name, Brand, Price, Quantity_in_Stock)

CLIENT (**ClientID**, F_Name, L_Name, Phone_Number, E-Mail, City)

ORDER (**OrderID**, # ClientID, OrderDate, TotalAmount)

ORDER_LINE (**# OrderID**, **#ProductID**, Quantity)

Express the following queries using SQL:

1. List all products with a stock quantity greater than 50.

```
SELECT * FROM PRODUCT
WHERE Quantity_in_Stock > 50;
```

2. Display the list of orders sorted by their total amount.

Note: In Microsoft Access, ORDER is a reserved keyword. Therefore, it must be enclosed in square brackets as **[ORDER]**.

```
SELECT * FROM [ORDER]
ORDER BY TotalAmount DESC;
```

3. Display orders placed between January 1, 2023, and March 31, 2023.

```
SELECT *
FROM [ORDER]
WHERE OrderDate BETWEEN #01/01/2023# AND #03/31/2023#;
```

4. Retrieve all orders placed during the year 2023.

```
SELECT *
FROM [ORDER]
WHERE YEAR(OrderDate) = 2023;
```

5. Display products of brand HP or Sumsung or Apple using `IN`.

```
SELECT *  
FROM PRODUCT  
WHERE Brand IN ('HP', 'Samsung', 'Apple');
```

6. Display the top three products with the highest quantities in stock.

```
SELECT TOP 3 *  
FROM PRODUCT  
ORDER BY Quantity_in_Stock DESC;
```

7. Display the number of products per brand.

```
SELECT Brand, COUNT(*) AS NumberOfProducts  
FROM PRODUCT  
GROUP BY Brand;
```

8. Find the average price of Laptops of brand 'DELL'.

```
SELECT AVG(Price) AS AveragePrice  
FROM PRODUCT  
WHERE Brand = 'DELL' AND Name LIKE '*Laptop*';
```

9. Show the number of orders per year in descending order.

```
SELECT YEAR(OrderDate) AS OrderYear, COUNT(*) AS NumberOfOrders  
FROM [ORDER]  
GROUP BY YEAR(OrderDate)  
ORDER BY NumberOfOrders DESC;
```

10. Display the number of clients for each city.

```
SELECT City, COUNT(*) AS Nbr_Clients  
FROM CLIENT  
GROUP BY City;
```

11. Find the names of clients who live in the same city as 'Karim Abadi'.

```
SELECT F_Name, L_Name FROM CLIENT
WHERE City = ( SELECT City FROM CLIENT
WHERE F_Name = 'Karim' AND L_Name = 'Abadi' );
```

12. Display the number of orders per month for the year 2022.

```
SELECT MONTH(OrderDate) AS OrderMonth, COUNT(*) AS Nbr_Orders
FROM [ORDER]
WHERE YEAR(OrderDate) = 2022
GROUP BY MONTH(OrderDate)
ORDER BY MONTH(OrderDate);
```

13. Retrieve the IDs of clients who have ordered at least one product of brand 'HP'.

```
SELECT DISTINCT o.ClientID FROM [ORDER] AS o
INNER JOIN ORDER_LINE AS ol ON o.OrderID = ol.OrderID
INNER JOIN PRODUCT AS p ON ol.ProductID = p.ProductID
WHERE p.Brand = 'HP';
```

14. Find the date with the highest purchase peak (in terms of quantities ordered).

```
SELECT TOP 1 o.OrderDate, SUM(ol.Quantity) AS TotalQuantity
FROM [ORDER] AS o INNER JOIN ORDER_LINE AS ol
ON o.OrderID = ol.OrderID
GROUP BY o.OrderDate
ORDER BY SUM(ol.Quantity) DESC;
```

15. Display clients who have placed at least three orders.

```
SELECT c.ClientID, c.F_Name, c.L_Name, COUNT(o.OrderID) AS Nbr_Orders
FROM CLIENT AS c INNER JOIN [ORDER] AS o
ON c.ClientID = o.ClientID
GROUP BY c.ClientID, c.F_Name, c.L_Name
HAVING COUNT(o.OrderID) >= 3;
```

16. For each product (showing its name), display the number of times it has been purchased.

```
SELECT p.ProductID, p.Name, SUM(ol.Quantity) AS TimesPurchased
FROM PRODUCT AS p LEFT JOIN ORDER_LINE AS ol
    ON p.ProductID = ol.ProductID
GROUP BY p.ProductID, p.Name;
```

17. Retrieve the IDs of clients who have not placed any orders for products of brand 'HP'.

```
SELECT c.ClientID FROM CLIENT AS c
WHERE NOT EXISTS (
    SELECT 1
    FROM [ORDER] AS o
    INNER JOIN ORDER_LINE AS ol ON o.OrderID = ol.OrderID
    INNER JOIN PRODUCT AS p ON ol.ProductID = p.ProductID
    WHERE o.ClientID = c.ClientID AND p.Brand = 'HP' );
```

18. Display brands whose average product price is greater than 30,000 DZD.

```
SELECT Brand, AVG(Price) AS AvgPrice
FROM PRODUCT
GROUP BY Brand
HAVING AVG(Price) > 30000;
```

19. Display clients who have placed at least one order

```
SELECT c.ClientID, c.F_Name, c.L_Name
FROM CLIENT AS c
WHERE EXISTS ( SELECT 1 FROM [ORDER] AS o WHERE o.ClientID = c.ClientID
);
```

20. Retrieve the IDs of clients who have ordered only products of brand 'Sony'.

```
SELECT c.ClientID FROM CLIENT AS c
WHERE EXISTS ( SELECT 1 FROM [ORDER] AS o
WHERE o.ClientID = c.ClientID )
AND NOT EXISTS (SELECT 1 FROM [ORDER] AS o
INNER JOIN ORDER_LINE AS ol ON o.OrderID = ol.OrderID
INNER JOIN PRODUCT AS p ON ol.ProductID = p.ProductID
WHERE o.ClientID = c.ClientID AND p.Brand <> 'Sony'
);
```

21. For each order, display the product ID, product name, price, client name, order date, and quantity.

```
SELECT o.OrderID, p.ProductID, p.Name AS ProductName, p.Price,
c.F_Name, c.L_Name AS ClientName, o.OrderDate, ol.Quantity
FROM ((([ORDER] AS o
INNER JOIN CLIENT AS c ON o.ClientID = c.ClientID)
INNER JOIN ORDER_LINE AS ol ON o.OrderID = ol.OrderID)
INNER JOIN PRODUCT AS p ON ol.ProductID = p.ProductID)
ORDER BY o.OrderDate;
```