CASE…ELSE in SQL Server

CASE is a **conditional expression** in SQL Server.
It works like **IF–ELSE** logic and is used inside:

- SELECT
- WHERE
- ORDER BY
- UPDATE
- HAVING

The **ELSE** part is **optional**, but **recommended** to handle unmatched conditions.

Below is a **Customer–Product based CASE…ELSE example**, similar to your previous requirement, with **table creation, sample data, queries, explanation, and output**. This is very commonly asked in **SQL Server interviews**.

Scenario: Customer & Product Example

Business Requirement

Classify customers based on **total purchase amount** of products.

Total Purchase Amount	Customer Category
< 10,000	Regular
10,000 – 25,000	Silver
> 25,000	Gold

Step 1: Create Tables

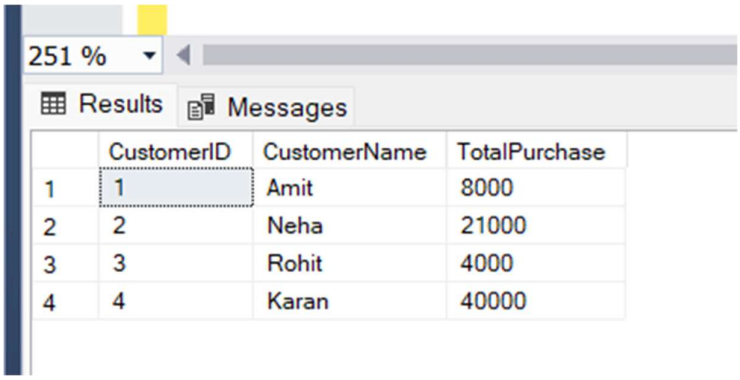
```
CREATE TABLE Customer
(
    CustomerID INT,
    CustomerName VARCHAR(50),
    City VARCHAR(30)
);
CREATE TABLE Product
(
    ProductID INT,
    ProductName VARCHAR(50),
    Price INT
);
CREATE TABLE Orders
(
    OrderID INT,
    CustomerID INT,
    ProductID INT,
    Quantity INT
);
```

Step 2: Insert Sample Data

```
INSERT INTO Customer VALUES
(1, 'Amit', 'Delhi'),
(2, 'Neha', 'Mumbai'),
(3, 'Rohit', 'Pune'),
(4, 'Karan', 'Delhi');
INSERT INTO Product VALUES
(101, 'Laptop', 40000),
(102, 'Mobile', 15000),
(103, 'Headphones', 3000),
(104, 'Keyboard', 2000);
INSERT INTO Orders VALUES
(1, 1, 103, 2),
(2, 1, 104, 1),
(3, 2, 102, 1),
(4, 2, 103, 2),
(5, 3, 104, 2),
(6, 4, 101, 1);
```

Step 3: Calculate Total Purchase per Customer

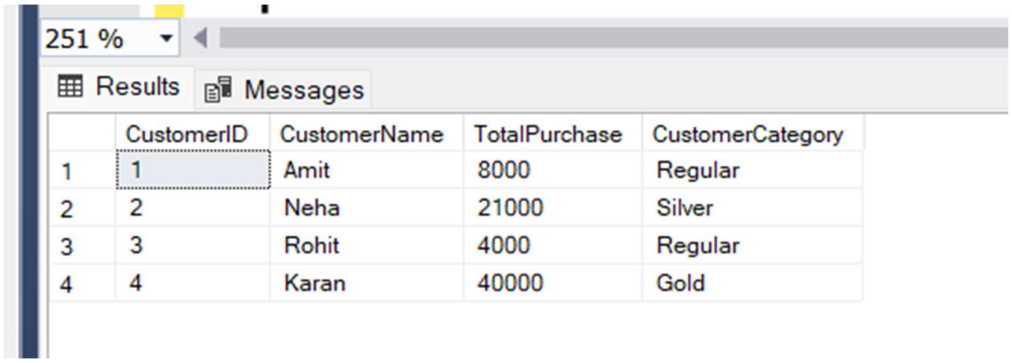
```
SELECT
    c.CustomerID,
    c.CustomerName,
    SUM(p.Price * o.Quantity) AS TotalPurchase
FROM Customer c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN Product p ON o.ProductID = p.ProductID
GROUP BY c.CustomerID, c.CustomerName;
```



	CustomerID	CustomerName	TotalPurchase
1	1	Amit	8000
2	2	Neha	21000
3	3	Rohit	4000
4	4	Karan	40000

Step 4: CASE…ELSE to Categorize Customers

```
SELECT
    c.CustomerID,
    c.CustomerName,
    SUM(p.Price * o.Quantity) AS TotalPurchase,
    CASE
        WHEN SUM(p.Price * o.Quantity) < 10000 THEN 'Regular'
        WHEN SUM(p.Price * o.Quantity) BETWEEN 10000 AND 25000 THEN 'Silver'
        ELSE 'Gold'
    END AS CustomerCategory
FROM Customer c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN Product p ON o.ProductID = p.ProductID
GROUP BY c.CustomerID, c.CustomerName;
```



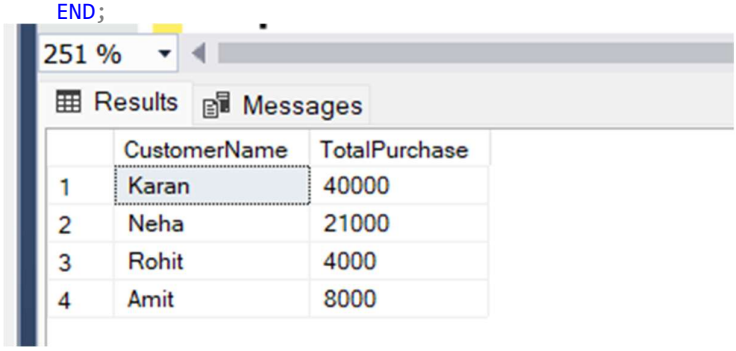
	CustomerID	CustomerName	TotalPurchase	CustomerCategory
1	1	Amit	8000	Regular
2	2	Neha	21000	Silver
3	3	Rohit	4000	Regular
4	4	Karan	40000	Gold

Explanation (Important)

- SUM(p.Price * o.Quantity) calculates total spending per customer
- CASE evaluates the **aggregated value**
- ELSE ensures **no customer is left uncategorized**
- Works only with GROUP BY when using aggregate functions

Another Example: CASE with ELSE in ORDER BY (Customer Priority)

```
SELECT
    c.CustomerName,
    SUM(p.Price * o.Quantity) AS TotalPurchase
FROM Customer c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN Product p ON o.ProductID = p.ProductID
GROUP BY c.CustomerName
ORDER BY
    CASE
        WHEN SUM(p.Price * o.Quantity) > 25000 THEN 1
        WHEN SUM(p.Price * o.Quantity) BETWEEN 10000 AND 25000 THEN 2
        ELSE 3
    END;
```



	CustomerName	TotalPurchase
1	Karan	40000
2	Neha	21000
3	Rohit	4000
4	Amit	8000

Order Result:

1. Gold customers
2. Silver customers
3. Regular customers