SQL case-based assignment with a scenario and 10 queries for a CRM (Customer Relationship Management) application.Scenario

```
CREATE DATABASE CUSTOMER_RELATIONSHIP ;

USE CUSTOMER_RELATIONSHIP ;
```

CUSTOMERS TABLE

```
CREATE TABLE CUSTOMERS (

CUSTOMER_ID VARCHAR (20) PRIMARY KEY,

FIRST_NAME VARCHAR(100),

LAST_NAME VARCHAR(100),

EMAIL VARCHAR(100),

PHONE VARCHAR(20),

CITY VARCHAR(100),

COUNTRY VARCHAR(100),

JOIN_DATE DATE );


INSERT INTO CUSTOMERS (CUSTOMER_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE, CITY, COUNTRY, JOIN_DATE)

VALUES

   ('CUST001', 'Alice', 'Johnson', 'alice.johnson@email.com', '555-1001', 'New York', 'USA', '2024-01-15'),

   ('CUST002', 'Bob', 'Smith', 'bob.smith@email.com', '555-1002', 'London', 'UK', '2024-02-10'),

   ('CUST003', 'Carlos', 'Garcia', 'carlos.garcia@email.com', '555-1003', 'Madrid', 'Spain', '2024-03-05'),

   ('CUST004', 'Diana', 'Wang', 'diana.wang@email.com', '555-1004', 'Beijing', 'China', '2024-04-20'),

   ('CUST005', 'Ethan', 'Patel', 'ethan.patel@email.com', '555-1005', 'Mumbai', 'India', '2024-05-01');
```

ORDERS TABLE

```
CREATE TABLE ORDERS (
ORDER_ID VARCHAR(20) PRIMARY KEY,
CUSTOMER_ID VARCHAR(20),
ORDER_DATE DATETIME,
AMOUNT DECIMAL(10,2),
ORDER_STATUS ENUM ('COMPLETED','PENDING','CANCELLED')
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID) );


INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_DATE, AMOUNT, ORDER_STATUS)
VALUES
    ('ORD001', 'CUST001', '2025-01-10 14:30:00', 150.50, 'COMPLETED'),
    ('ORD002', 'CUST002', '2025-01-12 09:15:00', 200.00, 'PENDING'),
    ('ORD003', 'CUST003', '2025-01-15 16:45:00', 99.99, 'CANCELLED'),
    ('ORD004', 'CUST004', '2025-01-18 11:00:00', 320.00, 'COMPLETED'),
    ('ORD005', 'CUST005', '2025-01-20 13:20:00', 250.75, 'PENDING');
```

SUPPORT TABLE

```
CREATE TABLE SUPPORT_TICKETS (
TICKET_ID VARCHAR(20) PRIMARY KEY,
CUSTOMER_ID VARCHAR(20),
ISSUE_DESCRIPTION TEXT,
STATUS ENUM ('OPEN','IN-PROGRESS','RESOLVED'),
CREATED DATE DATE,
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID) );
```

```
INSERT INTO SUPPORT_TICKETS (TICKET_ID, CUSTOMER_ID, ISSUE_DESCRIPTION, STATUS,
CREATED_DATE)

VALUES

    ('TKT001', 'CUST001', 'Login issues with account', 'OPEN', '2025-05-01'),

    ('TKT002', 'CUST002', 'Order not delivered', 'IN-PROGRESS', '2025-05-02'),

    ('TKT003', 'CUST003', 'Incorrect billing amount', 'RESOLVED', '2025-05-03'),

    ('TKT004', 'CUST004', 'Unable to reset password', 'RESOLVED', '2025-05-04'),

    ('TKT005', 'CUST005', 'Website not loading on mobile', 'OPEN', '2025-05-05');
```

INTERACTIONS TABLE

```
CREATE TABLE INTERACTIONS (

INTERACTION_ID VARCHAR(20) PRIMARY KEY,

CUSTOMER_ID VARCHAR(20),

INTERACTION_TYPE ENUM ( 'EMAIL','PHONE', 'IN-PERSON' ),

INTERACTION_DATE DATETIME,

FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID) );


INSERT INTO INTERACTIONS (INTERACTION_ID, CUSTOMER_ID, INTERACTION_TYPE,
INTERACTION_DATE)

VALUES

    ('INT001', 'CUST001', 'EMAIL', '2025-05-01 10:30:00'),

    ('INT002', 'CUST002', 'PHONE', '2025-05-02 14:00:00'),

    ('INT003', 'CUST003', 'IN-PERSON', '2025-05-03 11:15:00'),

    ('INT004', 'CUST004', 'EMAIL', '2025-05-04 16:45:00'),

    ('INT005', 'CUST005', 'PHONE', '2025-05-05 09:50:00');
```

Assignment Queries

1. Full name & email of customers who joined after January 1, 2023

```
SELECT FIRST_NAME, LAST_NAME, EMAIL
FROM CUSTOMERS
WHERE JOIN_DATE > '2023-01-01';
```

2. Total amount of completed orders for each customer

```
SELECT CUSTOMER_ID, SUM(AMOUNT) AS TOTAL_COMPLETED_AMOUNT
FROM ORDERS
WHERE ORDER_STATUS = 'COMPLETED'
GROUP BY CUSTOMER_ID;
```

3. All unresolved support tickets with customer's name

```
SELECT ST.TICKET_ID, C.FIRST_NAME, C.LAST_NAME, ST.STATUS
FROM SUPPORT_TICKETS ST
JOIN CUSTOMERS C ON ST.CUSTOMER_ID = C.CUSTOMER_ID
WHERE ST.STATUS IN ('OPEN', 'IN-PROGRESS');
```

4. Most recent interaction for each customer

```
SELECT I1.*
FROM INTERACTIONS I1
JOIN (
  SELECT CUSTOMER_ID, MAX(INTERACTION_DATE) AS LATEST
  FROM INTERACTIONS
  GROUP BY CUSTOMER_ID
) I2 ON I1.CUSTOMER_ID = I2.CUSTOMER_ID AND I1.INTERACTION_DATE = I2.LATEST;
```

5. Number of customers in each country

```
SELECT COUNTRY, COUNT(*) AS CUSTOMER_COUNT
FROM CUSTOMERS
GROUP BY COUNTRY;
```

6. Customers who have placed no orders

```
SELECT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME

FROM CUSTOMERS C

LEFT JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID

WHERE O.ORDER_ID IS NULL;
```

7. Customers with unresolved support tickets & pending orders

```
SELECT DISTINCT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME

FROM CUSTOMERS C

JOIN SUPPORT_TICKETS ST ON C.CUSTOMER_ID = ST.CUSTOMER_ID

JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID

WHERE ST.STATUS IN ('OPEN', 'IN-PROGRESS')

  AND O.ORDER_STATUS = 'PENDING';
```

8. Average order amount for each order status

```
SELECT ORDER_STATUS, AVG(AMOUNT) AS AVERAGE_AMOUNT
FROM ORDERS
GROUP BY ORDER_STATUS;
```

8. Customers who interacted via email in the past month

```
SELECT DISTINCT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME

FROM CUSTOMERS C

JOIN INTERACTIONS I ON C.CUSTOMER_ID = I.CUSTOMER_ID

WHERE I.INTERACTION_TYPE = 'EMAIL'

  AND I.INTERACTION_DATE >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
```

10. Top 5 customers by total order amount

```
SELECT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME, SUM(O.AMOUNT) AS TOTAL_SPENT

FROM CUSTOMERS C

JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID

GROUP BY C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME

ORDER BY TOTAL_SPENT DESC

LIMIT 5;
```