

Task:

1. Consider two or more related datasets from your chosen software (e.g., customer data and order data).

```
CREATE TABLE Ordercus (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT,  
    OrderDate DATE,  
    TotalAmount DECIMAL(10, 2),  
    FOREIGN KEY (CustomerID) REFERENCES Customers1(CustomerID)  
);
```

```
select * from customers1
```

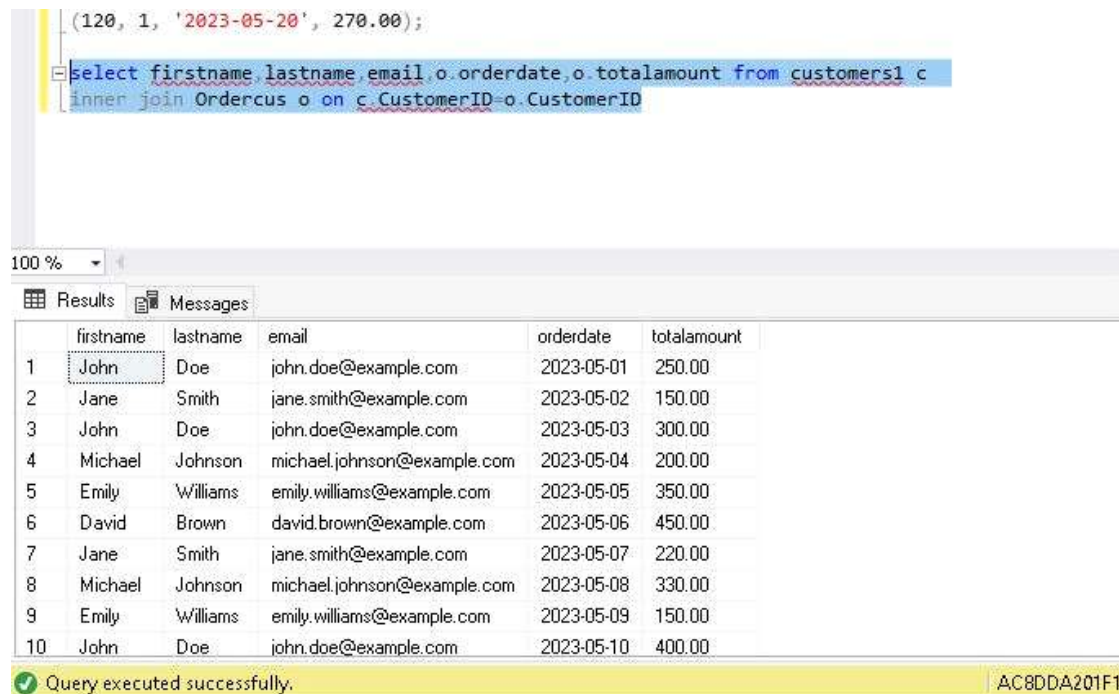
```
INSERT INTO Ordercus(OrderID, CustomerID, OrderDate, TotalAmount) VALUES  
(101, 1, '2023-05-01', 250.00),  
(102, 2, '2023-05-02', 150.00),  
(103, 1, '2023-05-03', 300.00),  
(104, 3, '2023-05-04', 200.00),  
(105, 4, '2023-05-05', 350.00),  
(106, 5, '2023-05-06', 450.00),  
(107, 2, '2023-05-07', 220.00),  
(108, 3, '2023-05-08', 330.00),  
(109, 4, '2023-05-09', 150.00),  
(110, 1, '2023-05-10', 400.00),  
(111, 2, '2023-05-11', 180.00),  
(112, 5, '2023-05-12', 275.00),  
(113, 3, '2023-05-13', 320.00),  
(114, 4, '2023-05-14', 290.00),  
(115, 1, '2023-05-15', 500.00),
```

```
(116, 2, '2023-05-16', 120.00),
(117, 3, '2023-05-17', 360.00),
(118, 5, '2023-05-18', 410.00),
(119, 4, '2023-05-19', 380.00),
(120, 1, '2023-05-20', 270.00);
```

2. Practice data integration:

Merge: Combine datasets based on a common key (e.g., customer ID) using

MERGE or similar functionality.



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

```
(120, 1, '2023-05-20', 270.00);

select firstname, lastname, email, o.orderdate, o.totalamount from customers1 c
inner join Ordercus o on c.CustomerID=o.CustomerID
```

The results window displays a table with 10 rows and 6 columns: **firstname**, **lastname**, **email**, **orderdate**, and **totalamount**. The data is as follows:

| | firstname | lastname | email | orderdate | totalamount |
|----|-----------|----------|-----------------------------|------------|-------------|
| 1 | John | Doe | john.doe@example.com | 2023-05-01 | 250.00 |
| 2 | Jane | Smith | jane.smith@example.com | 2023-05-02 | 150.00 |
| 3 | John | Doe | john.doe@example.com | 2023-05-03 | 300.00 |
| 4 | Michael | Johnson | michael.johnson@example.com | 2023-05-04 | 200.00 |
| 5 | Emily | Williams | emily.williams@example.com | 2023-05-05 | 350.00 |
| 6 | David | Brown | david.brown@example.com | 2023-05-06 | 450.00 |
| 7 | Jane | Smith | jane.smith@example.com | 2023-05-07 | 220.00 |
| 8 | Michael | Johnson | michael.johnson@example.com | 2023-05-08 | 330.00 |
| 9 | Emily | Williams | emily.williams@example.com | 2023-05-09 | 150.00 |
| 10 | John | Doe | john.doe@example.com | 2023-05-10 | 400.00 |

At the bottom of the results window, a status bar indicates: **Query executed successfully.** AC8DDA201F1

3. Enhance data organization:

Sort: Apply **ORDER BY** clauses to sort data based on specific columns (e.g.,

sorting orders by date).

```
select * from customers1 order by firstname
```

| | CustomerID | FirstName | LastName | Email | BirthDate | JoinDate | City | Salary | PurchaseTotal |
|----|------------|-------------|-----------|----------------------------------|------------|------------|---------------|--------|---------------|
| 1 | 12 | Barbara | Lopez | barbara.lopez@example.com | 1979-12-10 | 2020-12-10 | Jacksonville | 67000 | 1800.25 |
| 2 | 11 | Charles | Hernandez | charles.hernandez@example.com | 1986-11-05 | 2020-11-05 | Austin | 69000 | 1950.50 |
| 3 | 13 | Christopher | Gonzalez | christopher.gonzalez@example.com | 1994-01-15 | 2021-01-15 | San Francisco | 76000 | 2600.75 |
| 4 | 17 | Daniel | Taylor | daniel.taylor@example.com | 1976-05-05 | 2021-05-05 | Detroit | 80000 | 3000.50 |
| 5 | 5 | David | Brown | david.brown@example.com | 1975-05-30 | 2020-05-30 | Phoenix | 80000 | 3000.50 |
| 6 | 4 | Emily | Williams | emily.williams@example.com | 1982-04-25 | 2020-04-25 | Houston | 65000 | 1800.00 |
| 7 | 9 | James | Rodriguez | james.rodriguez@example.com | 1983-09-25 | 2020-09-25 | Dallas | 75000 | 2500.75 |
| 8 | 2 | Jane | Smith | jane.smith@example.com | 1985-02-15 | 2020-02-15 | Los Angeles | 70000 | 2000.75 |
| 9 | 14 | Jessica | Wilson | jessica.wilson@example.com | 1981-02-20 | 2021-02-20 | Columbus | 70000 | 2000.50 |
| 10 | 1 | John | Doe | john.doe@example.com | 1980-01-01 | 2020-01-01 | New York | 60000 | 1500.50 |

Query executed successfully.

Filtering: Utilize the WHERE clause with various operators to filter and focus on specific subsets of data (e.g., finding orders placed in a particular month).

```
select Firstname, lastname, email, o.orderdate, o.totalamount from customers1 c
inner join Ordercus o on c.CustomerID=o.CustomerID
where month(orderdate)=5 and day(orderdate)=3
```

| | Firstname | lastname | email | orderdate | totalamount |
|---|-----------|----------|----------------------|------------|-------------|
| 1 | John | Doe | john.doe@example.com | 2023-05-03 | 300.00 |