**Use nortiwind database for these exercises**

**Exercise 1: Retrieve the names of customers who have placed orders.**

-- Solution: Using UNION

SELECT CustomerID, CompanyName FROM Customers

WHERE CustomerID IN (SELECT CustomerID FROM Orders)

UNION

SELECT CustomerID, ' ' AS CompanyName FROM Orders

WHERE CustomerID NOT IN (SELECT CustomerID FROM Customers);

**Exercise 2: Find the common customers between Customers and Employees.**

-- Solution: Using INTERSECT

SELECT CustomerID, CompanyName FROM Customers

INTERSECT

SELECT EmployeeID, '' AS CompanyName FROM Employees;

-- but it will not work in mysql

-- Solution: Using INNER JOIN

SELECT Customers.CustomerID, CompanyName

FROM Customers

INNER JOIN Employees ON Customers.CustomerID = Employees.EmployeeID;

**Exercise 3: Retrieve the names of customers who have not placed any orders.**

-- Solution: Using EXCEPT (or MINUS in some databases)

SELECT CustomerID, CompanyName FROM Customers

EXCEPT

SELECT CustomerID, '' AS CompanyName FROM Orders;

-- but it will not work in mysql

-- Solution: Using NOT EXISTS

SELECT CustomerID, CompanyName FROM Customers

WHERE NOT EXISTS (SELECT 1 FROM Orders WHERE Orders.CustomerID = Customers.CustomerID);

**Exercise 4: Get the list of all unique cities involved in either shipping or billing.**

-- Solution: Using UNION

SELECT ShipCity AS City FROM Orders

UNION

SELECT City FROM Customers;

**Exercise 5: Retrieve the unique product categories that have been ordered.**

-- Solution: Using UNION

SELECT CategoryName FROM Categories

WHERE CategoryID IN (SELECT CategoryID FROM Products)

UNION

SELECT '' AS CategoryName FROM Products

WHERE CategoryID NOT IN (SELECT CategoryID FROM Categories);

**Exercise 6: Find the customers who have either placed an order or have a credit limit greater than $50,000.**

-- Solution: Using UNION

SELECT CustomerID, CompanyName FROM Customers

WHERE CustomerID IN (SELECT CustomerID FROM Orders)

UNION

SELECT CustomerID, CompanyName FROM Customers

WHERE CreditLimit > 50000;

**Exercise 7: Retrieve the unique ship countries and bill countries from orders.**

-- Solution: Using UNION

SELECT ShipCountry AS Country FROM Orders

UNION

SELECT Country FROM Customers;

**Exercise 8: Find employees who are also customers (based on matching names).**

-- Solution: Using UNION

SELECT EmployeeID, FirstName, LastName FROM Employees

WHERE CONCAT(FirstName, ' ', LastName) IN (SELECT CONCAT(ContactName) FROM Customers)

UNION

SELECT EmployeeID, FirstName, LastName FROM Employees

WHERE EmployeeID IN (SELECT EmployeeID FROM Customers);

**On MINUS**

**Exercise 9: Retrieve the customers who have not placed any orders.**

-- Solution: Using NOT IN

SELECT CustomerID, CompanyName FROM Customers

WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders);

**Exercise 10: Find the employees who are not customers.**

-- Solution: Using NOT IN

SELECT EmployeeID, FirstName, LastName FROM Employees

WHERE CONCAT(FirstName, ' ', LastName) NOT IN (SELECT CONCAT(ContactName) FROM Customers);

**Exercise 11: Retrieve the products that have not been ordered.**

-- Solution: Using NOT IN

SELECT ProductID, ProductName FROM Products

WHERE ProductID NOT IN (SELECT ProductID FROM OrderDetails);

**Exercise 12: Find the customers who have not placed any orders and have a credit limit greater than $50,000.**

-- Solution: Using NOT IN

SELECT CustomerID, CompanyName FROM Customers

WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders)

AND CreditLimit > 50000;

**On Intersection**

**Exercise 13: Retrieve the customers who are also employees (based on matching names).**

-- Solution: Using INNER JOIN

SELECT C.CustomerID, C.CompanyName

FROM Customers C

INNER JOIN Employees E ON CONCAT(E.FirstName, ' ', E.LastName) = C.ContactName;

**Exercise 14: Find the products that are both in the "Beverages" category and have been ordered.**

-- Solution: Using INNER JOIN

SELECT P.ProductID, P.ProductName

FROM Products P

INNER JOIN OrderDetails OD ON P.ProductID = OD.ProductID

INNER JOIN Orders O ON OD.OrderID = O.OrderID

WHERE P.CategoryID = 1; -- Assuming "Beverages" has CategoryID 1

**Exercise 15: Retrieve the employees who are also customers (based on matching names).**

-- Solution: Using EXISTS

SELECT E.EmployeeID, E.FirstName, E.LastName

FROM Employees E

WHERE EXISTS (

SELECT 1

FROM Customers C

WHERE CONCAT(E.FirstName, ' ', E.LastName) = C.ContactName

);

**Exercise 16: Find the customers who have placed orders and have a credit limit greater than $30,000.**

-- Solution: Using INNER JOIN

SELECT C.CustomerID, C.CompanyName

FROM Customers C

INNER JOIN Orders O ON C.CustomerID = O.CustomerID

WHERE C.CreditLimit > 30000;