1. Query to retrieve all customers who have placed orders but have never ordered a product with a unit price greater than $50:

SELECT Customers.CustomerID, Customers.CompanyName

FROM Customers

WHERE Customers.CustomerID IN (

SELECT Orders.CustomerID FROM Orders

)

EXCEPT

SELECT Customers.CustomerID, Customers.CompanyName

FROM Customers

WHERE Customers.CustomerID IN (

SELECT Orders.CustomerID FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Products.UnitPrice > 50

);

1. Query to retrieve all products that have been ordered more than once and also have a unit price less than $20:

SELECT Products.ProductID, Products.ProductName

FROM Products

WHERE Products.ProductID IN (

SELECT OrderDetails.ProductID FROM OrderDetails

GROUP BY OrderDetails.ProductID

HAVING COUNT(\*) > 1

)

INTERSECT

SELECT Products.ProductID, Products.ProductName

FROM Products

WHERE Products.UnitPrice < 20;

1. Query to retrieve all customers who have placed orders in both 1996 and 1997

SELECT Customers.CustomerID, Customers.CompanyName

FROM Customers

WHERE Customers.CustomerID IN (

SELECT Orders.CustomerID FROM Orders

WHERE Orders.OrderDate BETWEEN '1996-01-01' AND '1996-12-31'

)

INTERSECT

SELECT Customers.CustomerID, Customers.CompanyName

FROM Customers

WHERE Customers.CustomerID IN (

SELECT Orders.CustomerID FROM Orders

WHERE Orders.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'

);

1. Query to retrieve all employees who have sold products in both the USA and the UK

SELECT Employees.EmployeeID, Employees.FirstName || ' ' || Employees.LastName AS EmployeeName

FROM Employees

WHERE Employees.EmployeeID IN (

SELECT Orders.EmployeeID FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Customers.Country = 'USA'

)

INTERSECT

SELECT Employees.EmployeeID, Employees.FirstName || ' ' || Employees.LastName AS EmployeeName

FROM Employees

WHERE Employees.EmployeeID IN (

SELECT Orders.EmployeeID FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Customers.Country = 'UK'

);

1. Query to retrieve all products that have been ordered at least once and also have a unit price greater than $50:

SELECT Products.ProductID, Products.ProductName

FROM Products

WHERE Products.ProductID IN (

SELECT OrderDetails.ProductID FROM OrderDetails

)

INTERSECT

SELECT Products.ProductID, Products.ProductName

FROM Products

WHERE Products.UnitPrice > 50;

AGG FUNCTIONS:

1. Top 10 customers by total revenue
2. SELECT Customers.CustomerID, Customers.CompanyName, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity \* (1 - OrderDetails.Discount)) AS TotalRevenue
3. FROM Customers
4. JOIN Orders ON Customers.CustomerID = Orders.CustomerID
5. JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
6. GROUP BY Customers.CustomerID, Customers.CompanyName
7. ORDER BY TotalRevenue DESC

LIMIT 10

1. Total revenue by year

SELECT YEAR(OrderDate) AS OrderYear, SUM(UnitPrice \* Quantity \* (1 - Discount)) AS TotalRevenue

FROM OrderDetails

JOIN Orders ON OrderDetails.OrderID = Orders.OrderID

GROUP BY YEAR(OrderDate)

ORDER BY YEAR(OrderDate)

1. Retrieve the average order quantity for each category

SELECT

Categories.CategoryName,

AVG(OrderDetails.Quantity) AS AvgOrderQuantity

FROM Categories

JOIN Products ON Categories.CategoryID = Products.CategoryID

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Categories.CategoryName;

1. Find all customers who have placed orders in 1997 but not in 1996

SELECT \*

FROM Customers

WHERE CustomerID NOT IN (

SELECT CustomerID

FROM Orders

WHERE YEAR(OrderDate) = 1996

) AND CustomerID IN (

SELECT CustomerID

FROM Orders

WHERE YEAR(OrderDate) = 1997

)

1. Find the average order total for each customer, and show only customers with an average order total greater than $1,000

SELECT Customers.CustomerID, Customers.CompanyName, AVG(OrderDetails.Quantity \* OrderDetails.UnitPrice) AS AvgOrderTotal

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

1. GROUP BY Customers.CustomerID, Customers.CompanyName

HAVING AvgOrderTotal > 1000

ORDER BY AvgOrderTotal DESC;

Proposed schema

**Users**: userID, username, email, password, firstName, lastName, bio, profilePicture

**Posts**: postID, userID, postText, postImage, postDate, postTime

**Likes**: likeID, userID, postID, likeDate, likeTime

**Comments**: commentID, userID, postID, commentText, commentDate, commentTime

**Followers**: followerID, userID, followerUserID, followDate, followTime

Proposed schema

**Patients**: patientID, firstName, lastName, dateOfBirth, address, phoneNumber, email

**Appointments**: appointmentID, patientID, doctorID, appointmentDate, appointmentTime

**Doctors**: doctorID, firstName, lastName, specialty

**Procedures**: procedureID, procedureName, procedureDescription, cost

**Medications**: medicationID, medicationName, medicationDescription, cost

**Patient\_Procedures**: patientProcedureID, patientID, procedureID, procedureDate, procedureTime

**Patient\_Medications**: patientMedicationID, patientID, medicationID, medicationDate, medicationTime