USE Northwind;

-- 1.1 Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.

SELECT CustomerID "Customer ID", CompanyName "Company", [Address], PostalCode "Post Code", City + ', ' + Country "City"

FROM Customers

WHERE City = 'Paris' OR City = 'London'

;

A screenshot of a cell phone

Description automatically generated

-- 1.2 List all products stored in bottles.

SELECT ProductName "Product Name"

FROM Products

WHERE QuantityPerUnit LIKE '%bottle%'

;

A screenshot of a cell phone

Description automatically generated

--1.3 Repeat question above, but add in the Supplier Name and Country.

SELECT ProductName "Product Name", CompanyName "Company Name", Country "Country"

FROM Products

INNER JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID

WHERE Products.QuantityPerUnit LIKE '%bottle%'

;

A screenshot of a cell phone

Description automatically generated

--1.4 Write an SQL Statement that shows how many products there are in each category.

-- Include Category Name in result set and list the highest number first.

SELECT CategoryName, COUNT(\*) AS "Number of Products in Each Category"

FROM Products

INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID

GROUP BY Categories.CategoryName

;

A screenshot of a computer

Description automatically generated

--1.5 List all UK employees using concatenation to join their title of courtesy, first name and last name together.

-- Also include their city of residence.

SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName AS "Name", City AS "City of Residence"

FROM Employees

WHERE Country = 'UK'

;

A screenshot of a cell phone

Description automatically generated

--1.6 List Sales Totals for all Sales Regions (via the Territories table using 4 joins) with a Sales Total greater than 1,000,000.

-- Use rounding or FORMAT to present the numbers.

SELECT ROUND(SUM((od.UnitPrice\*od.Quantity)\*(1-od.Discount)),1) AS "Sales Totals", r.RegionDescription "Region"

FROM [Order Details] od

INNER JOIN Orders o ON o.OrderID = od.OrderID

INNER JOIN EmployeeTerritories et ON et.EmployeeID = o.EmployeeID

INNER JOIN Territories t ON t.TerritoryID = et.TerritoryID

INNER JOIN Region r ON r.RegionID = t.RegionID

GROUP BY r.RegionDescription

HAVING ROUND(SUM((od.UnitPrice\*od.Quantity)\*(1-od.Discount)),1) > 1000000

;

A screenshot of a cell phone

Description automatically generated

-- 1.7 Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.

SELECT COUNT(\*) AS "Number of Orders with Freight amount greater than 100"

FROM Orders

WHERE ShipCountry = 'USA' OR ShipCountry = 'UK' AND Freight > 100.00

;

A screenshot of a cell phone

Description automatically generated

-- 1.8 Write an SQL Statement to identify the Order Number of the Order with the highest amount of discount applied to that order.

SELECT TOP 1 OrderID "Order Number with the Highest Amount of Discount", Discount

FROM [Order Details]

ORDER BY Discount DESC

;

A screenshot of a cell phone

Description automatically generated

-------------------------------------------------------------- Question 2 ------------------------------------------------------------

---- Table 1 -----

DROP TABLE Rooms;

CREATE TABLE Rooms

(

Room\_id INT NOT NULL IDENTITY PRIMARY KEY,

[AcademyID] INT,

[RoomName] VARCHAR(20),

[Description] VARCHAR(255),

[Capacity] INT

)

;

---- Table 2 -----

DROP TABLE Academies;

CREATE TABLE Academies

(

Academy\_id INT NOT NULL IDENTITY PRIMARY KEY,

[AcademyName] VARCHAR(255)

)

;

----- Table 3 ------

DROP TABLE CourseCatalog;

CREATE TABLE CourseCatalog

(

Course\_id INT NOT NULL IDENTITY PRIMARY KEY,

CourseName VARCHAR(255),

Duration INT

)

;

----- Table 4 ------

DROP TABLE Employees;

SET DATEFORMAT mdy;

CREATE TABLE Employees

(

Employee\_id INT NOT NULL IDENTITY PRIMARY KEY,

[FirstName] VARCHAR(255),

[LastName] VARCHAR(255),

[EmployeeType] VARCHAR(255),

[StartDate] DATE

)

;

----- Table 5 ------

DROP TABLE [dbo].[CourseSchedule];

CREATE TABLE [dbo].[CourseSchedule]

([CourseScheduleID] INT NOT NULL IDENTITY PRIMARY KEY,

[AcademyID] INT,

[RoomID] INT,

[CourseID] INT,

[StartDate] DATE,

[EndDate] DATE

)

;

----- Table 6 ------

DROP TABLE CourseScheduleTrainers;

CREATE TABLE CourseScheduleTrainers

(

[CourseScheduleID] INT,

[TrainerID] INT,

[TrainerType] VARCHAR(20)

)

;

----- Table 7 -----

DROP TABLE CourseScheduleAttendees

CREATE TABLE CourseScheduleAttendees

(

[CourseScheduleID] INT,

[AttendeeID] INT,

[Active] INT

)

;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------- Question 2.2. and 2.3--------------------------------------------------

------------ Question 2.3 & 2.3 -------------

----- Adding the London Academy -----

INSERT INTO [Academies]

([AcademyName])

VALUES

('London')

----- Adding the Data Stream -----

INSERT INTO [CourseCatalog]

([CourseName]

,[Duration])

VALUES

('Data',

8)

----- Adding New Trainer for Data -----

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Isabella','Jones','T')

----- Adding a new TA for Data ----

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Training','Assistant','A')

----- Adding Current Spartans in the Data Stream ----

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Sahal','Nurain','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Safal','Mukhia','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Oscar','China','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Viki','Patel','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Daniel','Hughes','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Jeevan','Sandhu','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Hudhayfee','Hassan','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Avnish','Hirani','S')

INSERT INTO [Employees]

([FirstName]

,[LastName]

,[EmployeeType]

)

VALUES

('Sarmi','Ranjan','S')

A screenshot of a cell phone

Description automatically generated

-------------------------------------------------------- Question 3 ------------------------------------------------------------------

------- Question 3.1 -------

SELECT ca.CourseName AS "Course Name",

cs.StartDate AS "Start Date",

cs.EndDate AS "End Date",

a.AcademyName AS "Academy",

r.RoomName AS "Room",

et.FirstName + ' ' + et.LastName AS "Trainer",

ea.FirstName + ' ' + ea.LastName AS "Spartan"

FROM CourseSchedule cs

INNER JOIN CourseCatalog ca ON ca.Course\_id = cs.CourseID

INNER JOIN CourseScheduleTrainers cst ON cs.CourseScheduleID = cst.CourseScheduleID

INNER JOIN CourseScheduleAttendees csa ON cs.CourseScheduleID = csa.CourseScheduleID

INNER JOIN Employees ea ON csa.AttendeeID = ea.Employee\_id

INNER JOIN Employees et ON cst.TrainerID = et.Employee\_id

INNER JOIN Rooms r ON cs.RoomID = r.Room\_id

INNER JOIN Academies a ON r.AcademyID = a.Academy\_id

;

A screenshot of a computer

Description automatically generated

--------- Question 3.2. -----------

SELECT ca.CourseName AS "Course Name",

cs.StartDate AS "Start Date",

cs.EndDate AS "End Date",

a.AcademyName AS "Academy",

r.RoomName AS "Room",

et.FirstName + ' ' + et.LastName AS "Trainer",

LEFT(ea.FirstName,1) + '. ' + LEFT(ea.LastName,1) AS "Spartan"

FROM CourseSchedule cs

INNER JOIN CourseCatalog ca ON ca.Course\_id = cs.CourseID

INNER JOIN CourseScheduleTrainers cst ON cs.CourseScheduleID = cst.CourseScheduleID

INNER JOIN CourseScheduleAttendees csa ON cs.CourseScheduleID = csa.CourseScheduleID

INNER JOIN Employees ea ON csa.AttendeeID = ea.Employee\_id

INNER JOIN Employees et ON cst.TrainerID = et.Employee\_id

INNER JOIN Rooms r ON cs.RoomID = r.Room\_id

INNER JOIN Academies a ON r.AcademyID = a.Academy\_id

;

A screenshot of a computer

Description automatically generated

------- Exercise 3.3 ----------

ALTER TABLE CourseSchedule

ADD [Check Date] DATE;

UPDATE CourseSchedule

SET [Check Date] = DATEADD(mm,2,EndDate)

WHERE CourseScheduleID = 1

;

UPDATE CourseSchedule

SET [Check Date] = DATEADD(mm,3,EndDate)

WHERE CourseScheduleID != 1

;

SELECT ca.CourseName AS "Course Name",

cs.StartDate AS "Start Date",

cs.EndDate AS "End Date",

cs.[Check Date] AS "Check Date",

a.AcademyName AS "Academy",

r.RoomName AS "Room",

et.FirstName + ' ' + et.LastName AS "Trainer",

LEFT(ea.FirstName,1) + '. ' + LEFT(ea.LastName,1) AS "Spartan"

FROM CourseSchedule cs

INNER JOIN CourseCatalog ca ON ca.Course\_id = cs.CourseID

INNER JOIN CourseScheduleTrainers cst ON cs.CourseScheduleID = cst.CourseScheduleID

INNER JOIN CourseScheduleAttendees csa ON cs.CourseScheduleID = csa.CourseScheduleID

INNER JOIN Employees ea ON csa.AttendeeID = ea.Employee\_id

INNER JOIN Employees et ON cst.TrainerID = et.Employee\_id

INNER JOIN Rooms r ON cs.RoomID = r.Room\_id

INNER JOIN Academies a ON r.AcademyID = a.Academy\_id

;

A screenshot of a computer

Description automatically generated

-------------------------------------------------------------- Question 4 ------------------------------------------------------------

----------- Question 4.1 ------------

----------- Adding Foreign Keys ------

ALTER TABLE CourseScheduleTrainers

ADD CONSTRAINT [cst.CourseScheduleID] FOREIGN KEY(CourseScheduleID) REFERENCES CourseSchedule(CourseScheduleID),

CONSTRAINT [cst.TrainerID] FOREIGN KEY(TrainerID) REFERENCES Employees(Employee\_id)

;

ALTER TABLE CourseScheduleAttendees

ADD CONSTRAINT [csa.CourseScheduleID] FOREIGN KEY(CourseScheduleID) REFERENCES CourseSchedule(CourseScheduleID),

CONSTRAINT [csa.CourseScheduleID] FOREIGN KEY(CourseScheduleID) REFERENCES CourseSchedule(CourseScheduleID)

;

---------- Question 4.2 -------------

----- Adding further constraints for trainer type and employee type and maximum capacity for rooms to 25 ----

ALTER TABLE CourseScheduleTrainers

ADD CONSTRAINT [cst.TrainerType] CHECK (TrainerType = 'T' OR TrainerType = 'A')

;

ALTER TABLE Employees

ADD CONSTRAINT [e.EmployeeType] CHECK (EmployeeType = 'T' OR EmployeeType = 'S' OR EmployeeType = 'A')

;

ALTER TABLE Rooms

ADD CONSTRAINT [r.Capacity] CHECK (Capacity <= 25)

;

--------------------------------------------------------------- Question 5 -----------------------------------------------------------

--Q5.1 - List all Employees from the Employees table and who they report to.

SELECT E.EmployeeID,

E.FirstName + ' ' + E.LastName AS "Employee",

E2.FirstName + ' ' + E2.LastName AS "Reports To"

FROM Employees E

INNER JOIN Employees E2 ON E2.EmployeeID = E.ReportsTo

ORDER BY [Reports To]

;

A screenshot of a computer

Description automatically generated

-- Q5.2 -List all Suppliers with total sales over $10,000 in the Order Details table.

-- Include the Company Name from the Suppliers Table and present as a bar chart.

SELECT s.CompanyName AS "Suppliers With Total Sales Over $10,000",

ROUND(SUM((od.UnitPrice\*od.Quantity)\*(1-od.Discount)),0) AS "Total Sales"

FROM [Order Details] od

INNER JOIN Products p ON od.ProductID = p.ProductID

INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID

GROUP BY s.CompanyName

HAVING SUM((od.UnitPrice\*od.Quantity)\*(1-od.Discount)) > 10000

ORDER BY 2 DESC

;

A screenshot of a cell phone

Description automatically generated

-- Q5.3 - List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped.

SELECT TOP 10 c.CompanyName AS "Top 10 Customers",

COUNT(o.ShippedDate) AS "Total Value of Orders Shipped",

MAX(YEAR(o.OrderDate)) AS "Latest Year"

FROM Customers c

INNER JOIN Orders o ON c.CustomerID = o.CustomerID

GROUP BY c.CompanyName

ORDER BY 2 DESC

;

A screenshot of a computer

Description automatically generated

-- q5.4. - Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below.

SELECT YEAR(o.OrderDate) AS "Year", MONTH(O.OrderDate) AS "Month",

AVG(DATEDIFF(dd, OrderDate, ShippedDate)) AS "Average Ship Time By Month"

FROM Orders o

GROUP BY MONTH(OrderDate), YEAR(OrderDate)

ORDER BY "Year" ASC

A screenshot of a computer

Description automatically generated;