SQL Assignment-4

**1: Create a stored procedure in the Northwind database that will calculate the average value of Freight for a specified customer.Then, a business rule will be added that will be triggered before every Update and Insert command in the Orders controller,and will use the stored procedure to verify that the Freight does not exceed the average freight. If it does, a message will be displayed and the command will be cancelled.**

**Query1:**

**TriggerName:** trigger\_orders\_beforeInsertionAvgFreight

CREATE DEFINER=`root`@`localhost` TRIGGER `trigger\_orders\_beforeInsertionAvgFreight` BEFORE INSERT ON `orders` FOR EACH ROW BEGIN

DECLARE avgFreight DOUBLE;

DECLARE counter INT;

SET counter = (SELECT count(CustomerID) FROM orders WHERE CustomerID=new.CustomerID);

SET avgFreight = (SELECT avg(Freight) FROM orders where CustomerID=new.CustomerID);

if new.Freight > avgFreight and counter > 0 then SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Freight error due to add more than average value of freight!!';

END IF;

END

**TriggerName:** trigger\_orders\_beforeUpdationFreight

CREATE DEFINER=`root`@`localhost` TRIGGER `trigger\_orders\_beforeUpdationFreight` BEFORE UPDATE ON `orders` FOR EACH ROW BEGIN

DECLARE avgFreight DOUBLE;

SET avgFreight = (SELECT avg(Freight) FROM orders where CustomerID = new.CustomerID);

if new.Freight > avgFreight then SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Freight error due to add more than average value of freight!!';

END IF;

END

**Output:**

**Successful Update & insertion**

****

**2: Write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country.**

**Query2:** call northwind.sp\_orders\_salesByCountry();

**Procedure Name:** sp\_orders\_salesByCountry

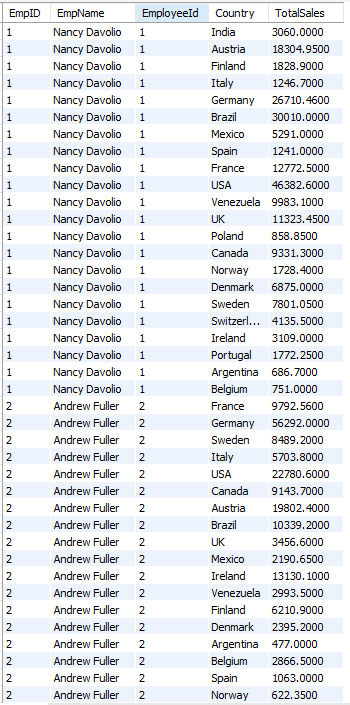
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orders\_salesByCountry`()

BEGIN

SELECT e.EmployeeID EmpID, CONCAT(e.FirstName, " ", e.LastName) EmpName, ordDetails.\* FROM employees e INNER JOIN (SELECT o.EmployeeId, o.shipCountry Country, sum(od.UnitPrice \* od.Quantity) TotalSales FROM orders o INNER JOIN orderDetails od ON o.OrderID = od.OrderID GROUP BY ShipCountry,EmployeeID) AS ordDetails ON e.employeeID = ordDetails.employeeID;

END

**Output:**

****

**3: Write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year.**

**Query3:** call northwind.sp\_orders\_salesByCountry();

**Procedure Name:** sp\_orders\_salesByYears

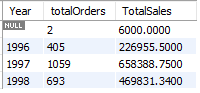
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orders\_salesByYears`()

BEGIN

SELECT YEAR(o.OrderDate) 'Year',count(o.orderID) totalOrders,sum(od.UnitPrice \* od.Quantity) TotalSales FROM orders o INNER JOIN orderDetails od ON o.orderID = od.orderID GROUP BY YEAR(OrderDate);

END

**Output:**

****

**4: Write a SQL query to create a stored procedure in the Northwind database to retrieve sales by category.**

**Query4:** call northwind.sp\_orders\_salesByCategories();

**Procedure Name:** sp\_orderdetails\_salesByCategories

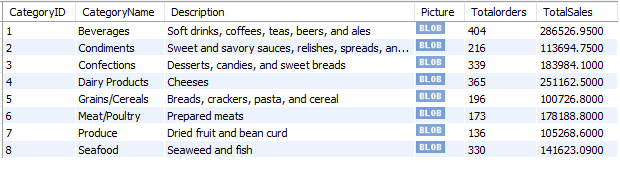
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orderdetails\_salesByCategories`()

BEGIN

select c.\*,count(o.OrderID ) Totalorders, sum(od.UnitPrice \* od.Quantity) TotalSales from orders o INNER JOIN orderDetails od ON o.OrderID = od.OrderID INNER JOIN products p ON p.ProductID = od.ProductID INNER JOIN categories c ON p.CategoryID = c.CategoryID group by c.categoryID ;

END

**Output:**

****

**5: Write a SQL query to create a stored procedure in the Northwind database to retrieve ten most expensive products.**

**Query5:** call northwind.sp\_product\_maxPrice10Retrive();

**Procedure Name:** sp\_product\_maxPrice10Retrive

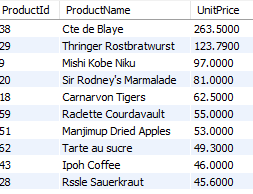
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_product\_maxPrice10Retrive`()

BEGIN

select ProductId, ProductName, UnitPrice from products as productList order by UnitPrice desc Limit 10 ;

END

**Output:**

****

**6: Write a SQL query to create a stored procedure in the Northwind database to insert customer order details.**

**Query6:** call northwind.sp\_orderDetails\_insertstatement(123, 25, 5000, 2, 1, 'WARTH', 6, null, null, null, 1, 0, 'DJShip', 'Bhavnagar', 'Bhavnagar', 'Asia', '364260', 'INDIA');

**Procedure Name:** sp\_orderDetails\_insertstatement

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orderDetails\_insertstatement`(IN OrderID INT, IN ProductID INT, IN UnitPrice DECIMAL(10,4), IN Quantity SMALLINT, IN Discount DOUBLE(8,0),IN CustomerID varchar(5), IN EmployeeID INT, IN OrderDate DATETIME, IN RequireDate DATETIME, IN ShippedDate DATETIME, IN ShipVia INT , IN Freight DECIMAL(10,4), IN ShipName VARCHAR(40), IN ShipAddress VARCHAR(60), IN ShipCity VARCHAR(15), IN ShipRegion VARCHAR(15), IN ShipPostalCode VARCHAR(10), IN ShipCountry VARCHAR(15))

BEGIN

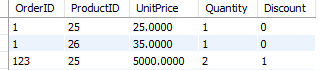
INSERT INTO orders VALUES(OrderID, CustomerID, EmployeeID, OrderDate, RequireDate, ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry);

INSERT INTO orderdetails VALUES (OrderID, ProductID, UnitPrice, Quantity, Discount);

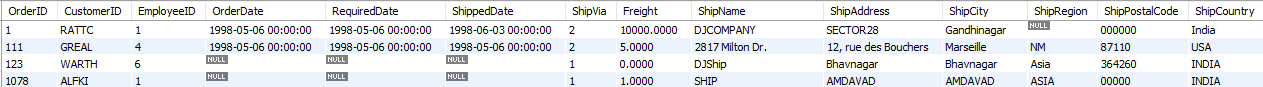
END

**Output:**

**OrderDetails**

****

**Orders**

****

**7: Write a SQL query to create a stored procedure in the Northwind database to update customer order details.**

**Query7:** call northwind.sp\_orderDetails\_updateStatement(123, 25, 123, 25, 1000, 3, 1, 'WARTH', 6, null, null, null, 1, 0, 'DJShip', 'Bhavnagar', 'Bhavnagar', 'Asia', '364260', 'INDIA');

**Procedure Name:** sp\_orderDetails\_updateStatement

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orderDetails\_updateStatement`(IN oldOrderId INT, In oldProductId INT, IN AtOrderID INT, IN AtProductID INT, IN AtUnitPrice DECIMAL(10,4), IN AtQuantity SMALLINT, IN AtDiscount DOUBLE(8,0), IN AtCustomerID varchar(5), IN AtEmployeeID INT, IN AtOrderDate DATETIME, IN AtRequireDate DATETIME, IN AtShippedDate DATETIME, IN AtShipVia INT , IN AtFreight DECIMAL(10,4), IN AtShipName VARCHAR(40), IN AtShipAddress VARCHAR(60), IN AtShipCity VARCHAR(15), IN AtShipRegion VARCHAR(15), IN AtShipPostalCode VARCHAR(10), IN AtShipCountry VARCHAR(15))

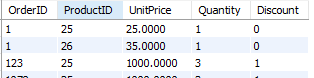
BEGIN

UPDATE orders SET OrderID = AtOrderID, CustomerID = AtCustomerID, EmployeeID = AtEmployeeId, OrderDate = AtOrderDate, RequireDate = AtRequireDate, ShippedDate = AtShippedDate , ShipVia = AtShipVia , Freight = AtFreight, ShipName = AtShipName,ShipAddress = AtShipAddress, ShipCity = AtShipCity,ShipRegion = AtShipRegion, ShipPostalCode = AtShipPostalCode, ShipCountry = AtShipCountry WHERE OrderId = oldOrderID;

UPDATE orderDetails SET OrderID = AtOrderID, ProductID = AtProductId, UnitPrice = AtUnitPrice, Quantity = AtQuantity, Discount = AtDiscount WHERE OrderId = oldOrderID and ProductId = oldProductId;

END

**Output: OrderDetails**

****