**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.**

**Query:**

\\ Trigger for Before Insert

CREATE DEFINER=`root`@`localhost` TRIGGER `orders\_BEFORE\_INSERT` BEFORE INSERT ON `orders` FOR EACH ROW BEGIN  
declare msg varchar(100);  
 SET msg = 'Here value is greater than the average value of freight, thus insert command cannot be executed.';  
 call AvgFreight(new.customerid,@AverageFreight);  
 if @AverageFreight < new.freight then   
 SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;  
 end if ;  
END

\\ Trigger for Before Update

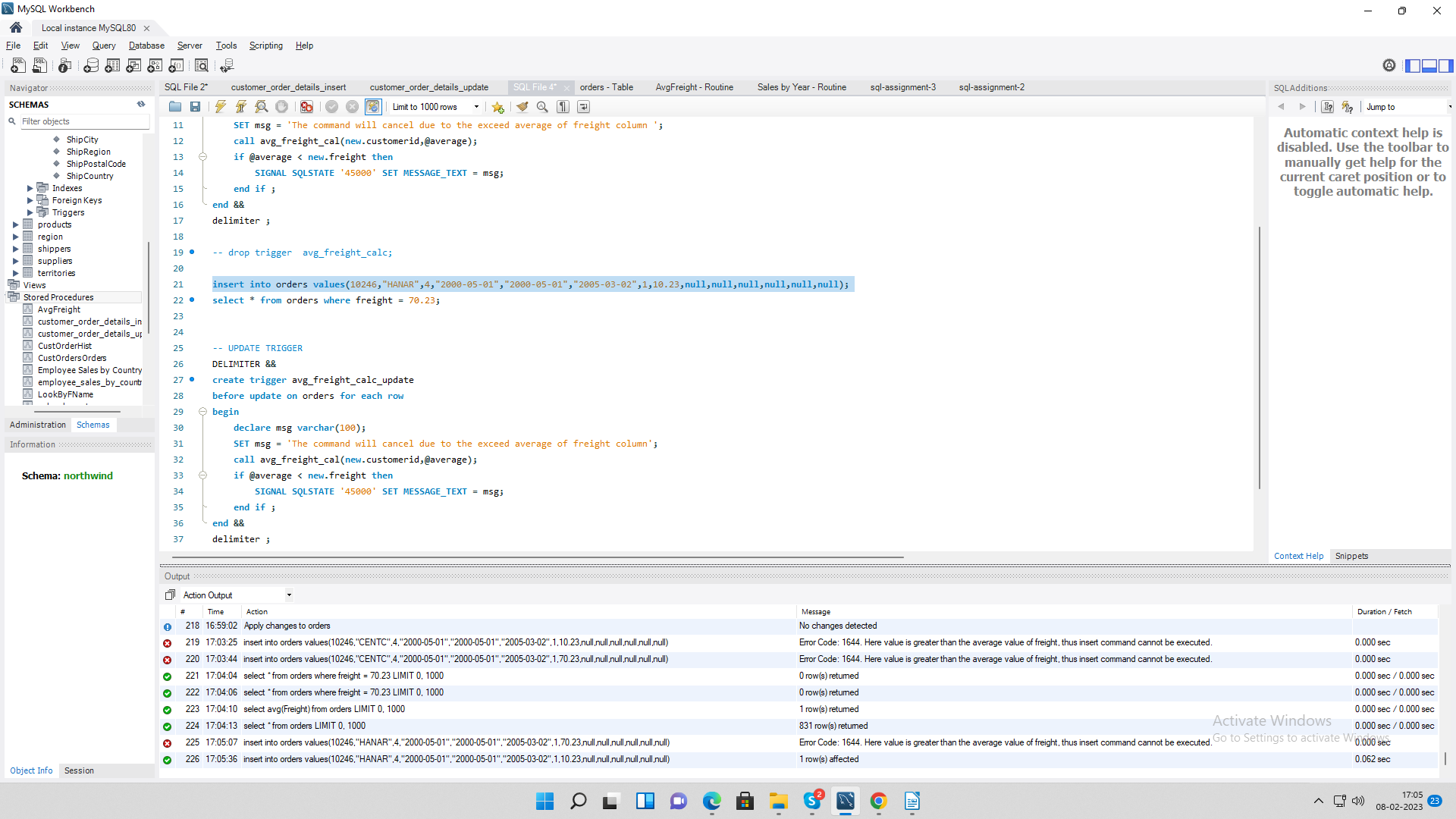
CREATE DEFINER=`root`@`localhost` TRIGGER `orders\_BEFORE\_UPDATE` BEFORE UPDATE ON `orders` FOR EACH ROW BEGIN  
declare msg varchar(100);  
 SET msg = 'Here value is greater than the average value of freight, thus update command cannot be executed.';  
 call AvgFreight(new.customerid,@AverageFreight);  
 if @AverageFreight < new.freight then   
 SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;  
 end if ;  
END

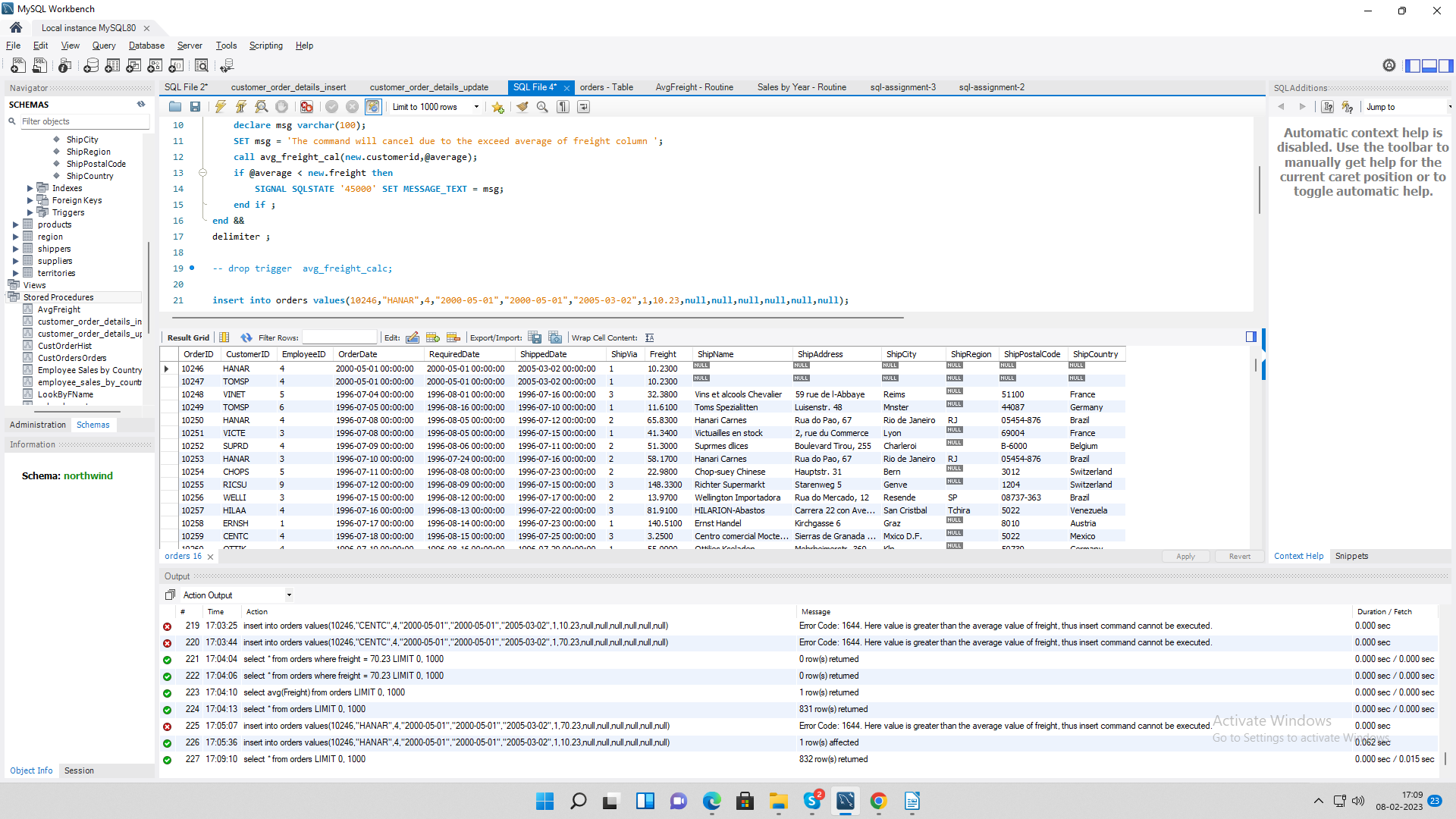
\\ Stored Procedure AvgFreight

CREATE DEFINER=`root`@`localhost` PROCEDURE `AvgFreight`(in customer\_id varchar(5),out AverageFreight float)  
BEGIN  
select avg(freight) as AverageFreight into AverageFreight from orders where customerid=customer\_id;  
END

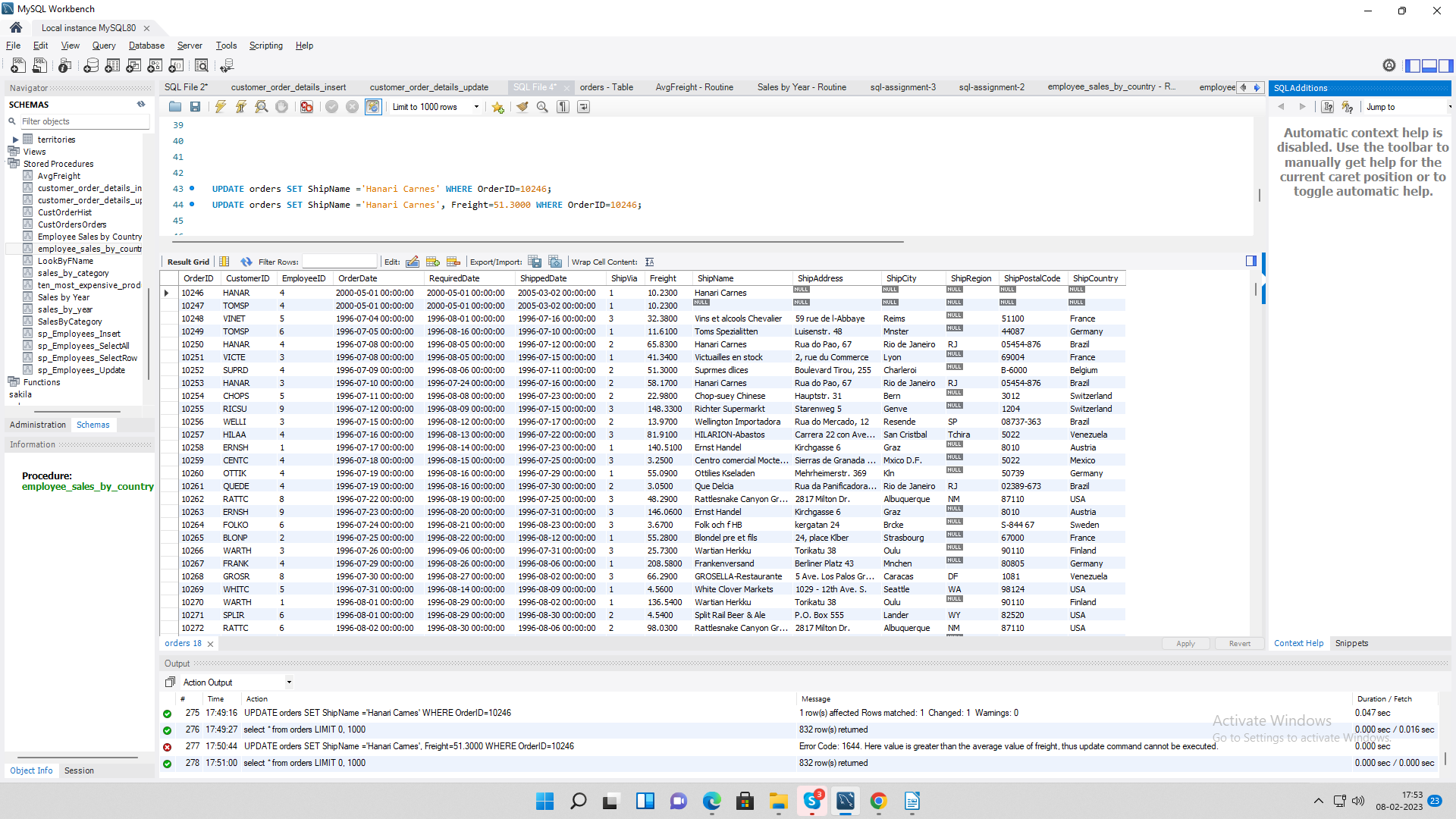
**Output:**

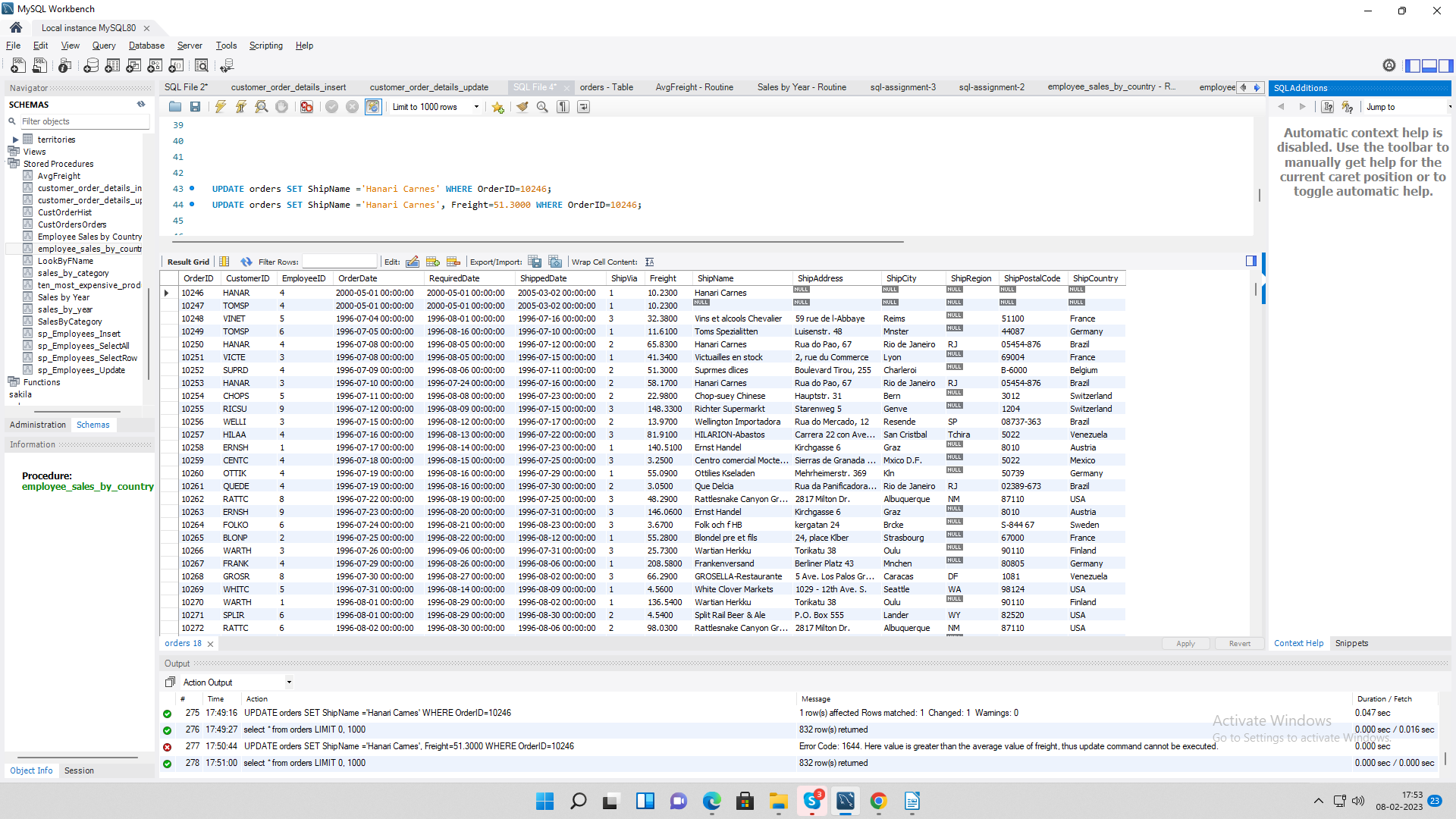
\\ For Insert





\\ For Update



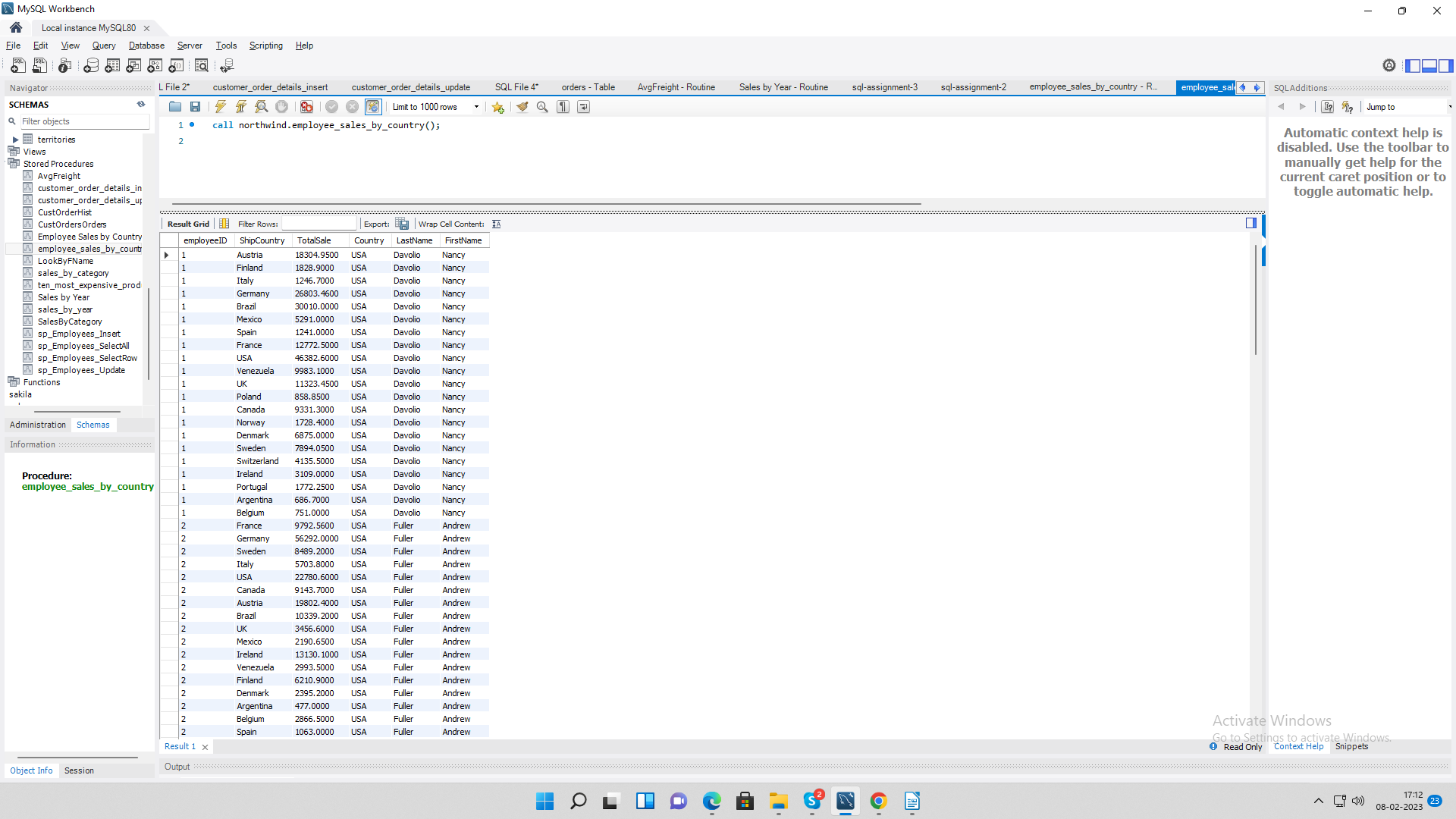


**2. Write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country**

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `employee\_sales\_by\_country`()  
BEGIN  
 SELECT OrderAttributes.\*,   
 e.Country,  
 e.LastName,  
 e.FirstName  
 FROM employees e   
 JOIN (SELECT o.employeeID, o.ShipCountry, sum(od.UnitPrice \* od.Quantity) AS TotalSale FROM orders o INNER JOIN `order details` od ON o.orderid = od.orderId   
 GROUP BY o.ShipCountry,o.EmployeeID) AS OrderAttributes ON e.employeeID = OrderAttributes.employeeID;   
END

**Output:**

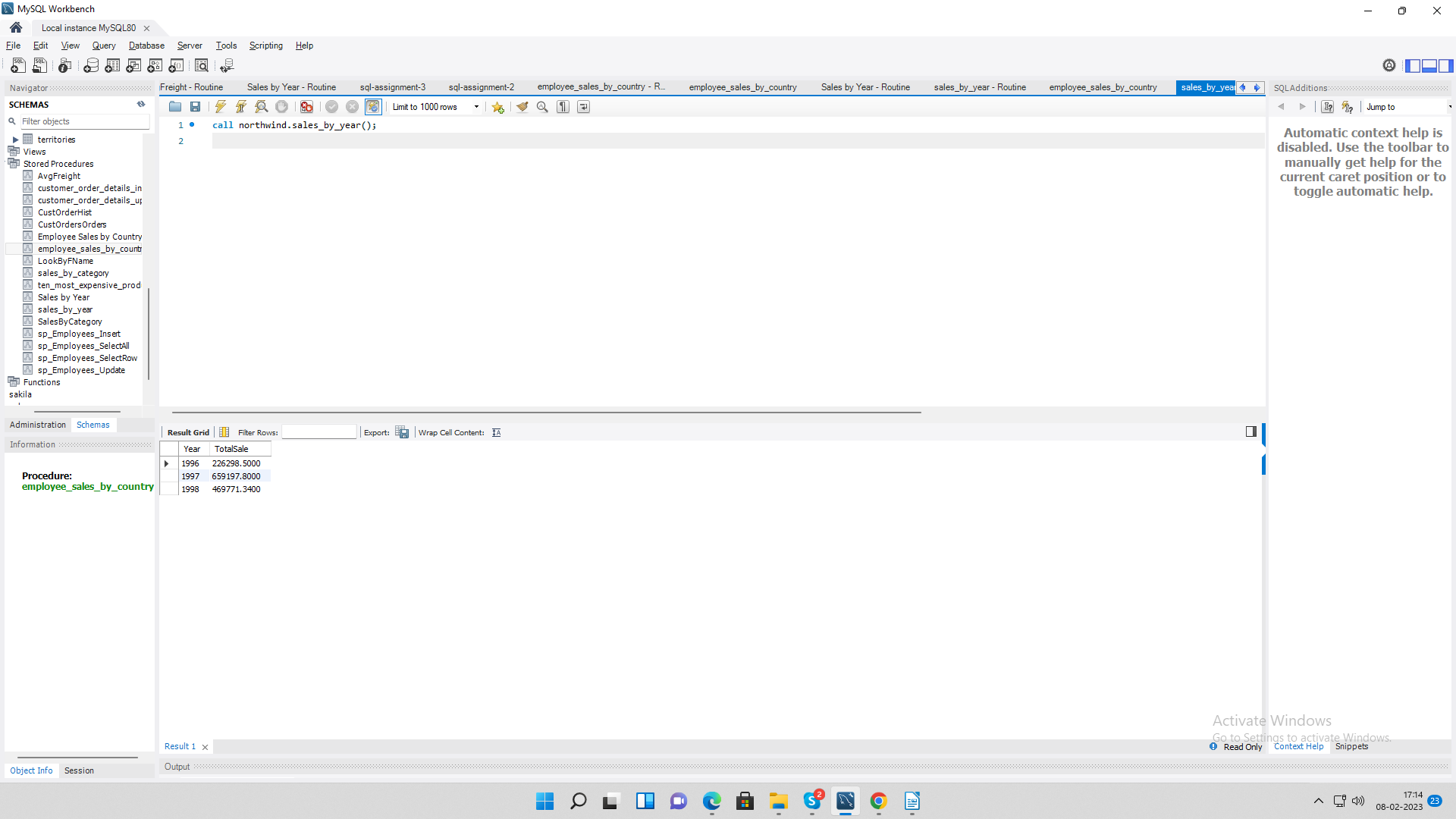
****

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

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `sales\_by\_year`()  
BEGIN  
SELECT YEAR(o.orderdate) AS Year,   
SUM(od.UnitPrice \* od.Quantity) AS TotalSale   
FROM orders o INNER JOIN `order details` od ON o.orderid = od.orderId GROUP BY YEAR(o.OrderDate);  
END

**Output:**

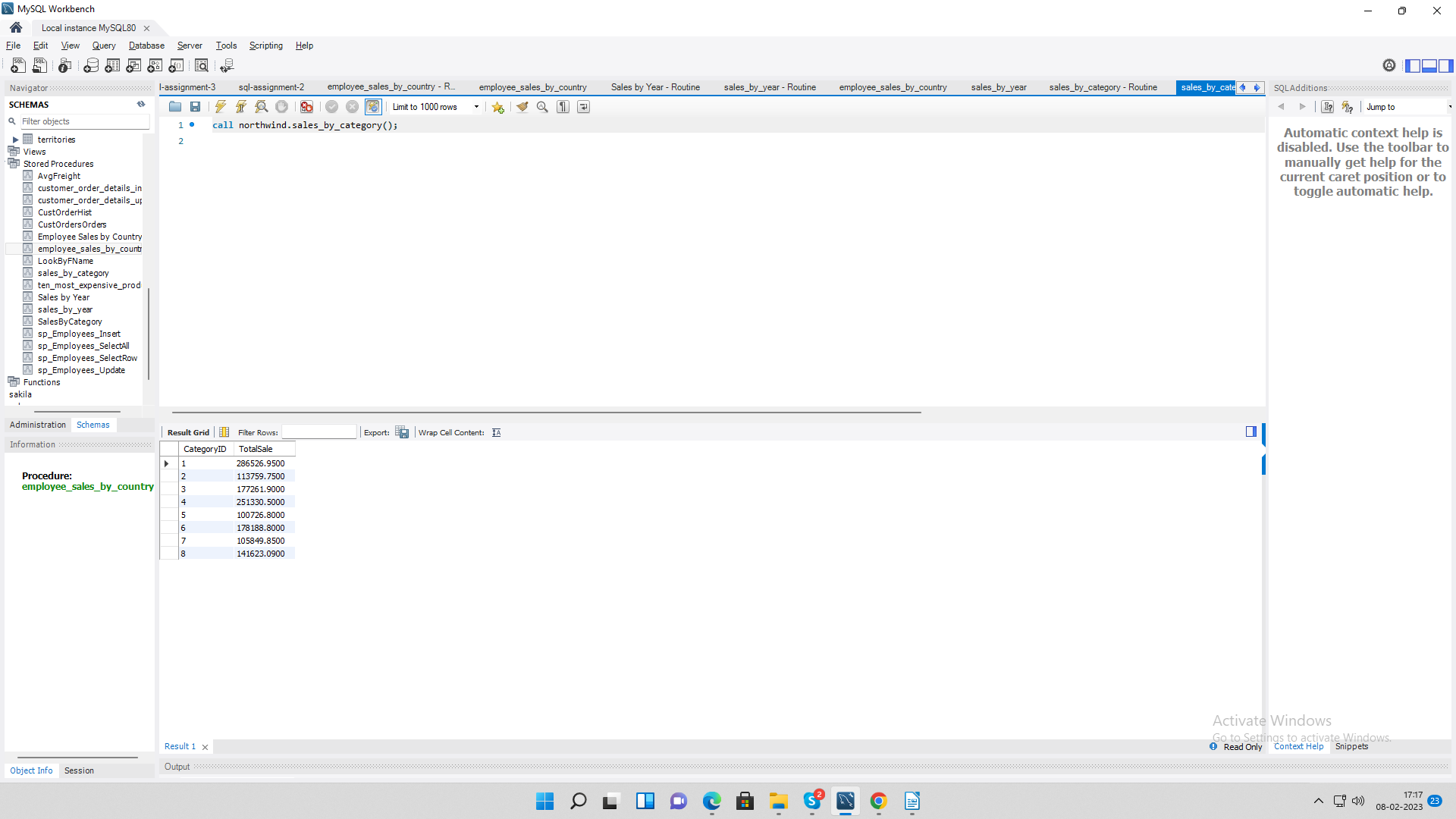


**4. Write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category**

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `sales\_by\_category`()  
BEGIN  
SELECT p.CategoryID,   
SUM(od.UnitPrice \* od.Quantity) AS TotalSale   
FROM products p INNER JOIN `order details` od ON p.ProductID=od.ProductID GROUP BY (p.CategoryID);  
END

**Output:**

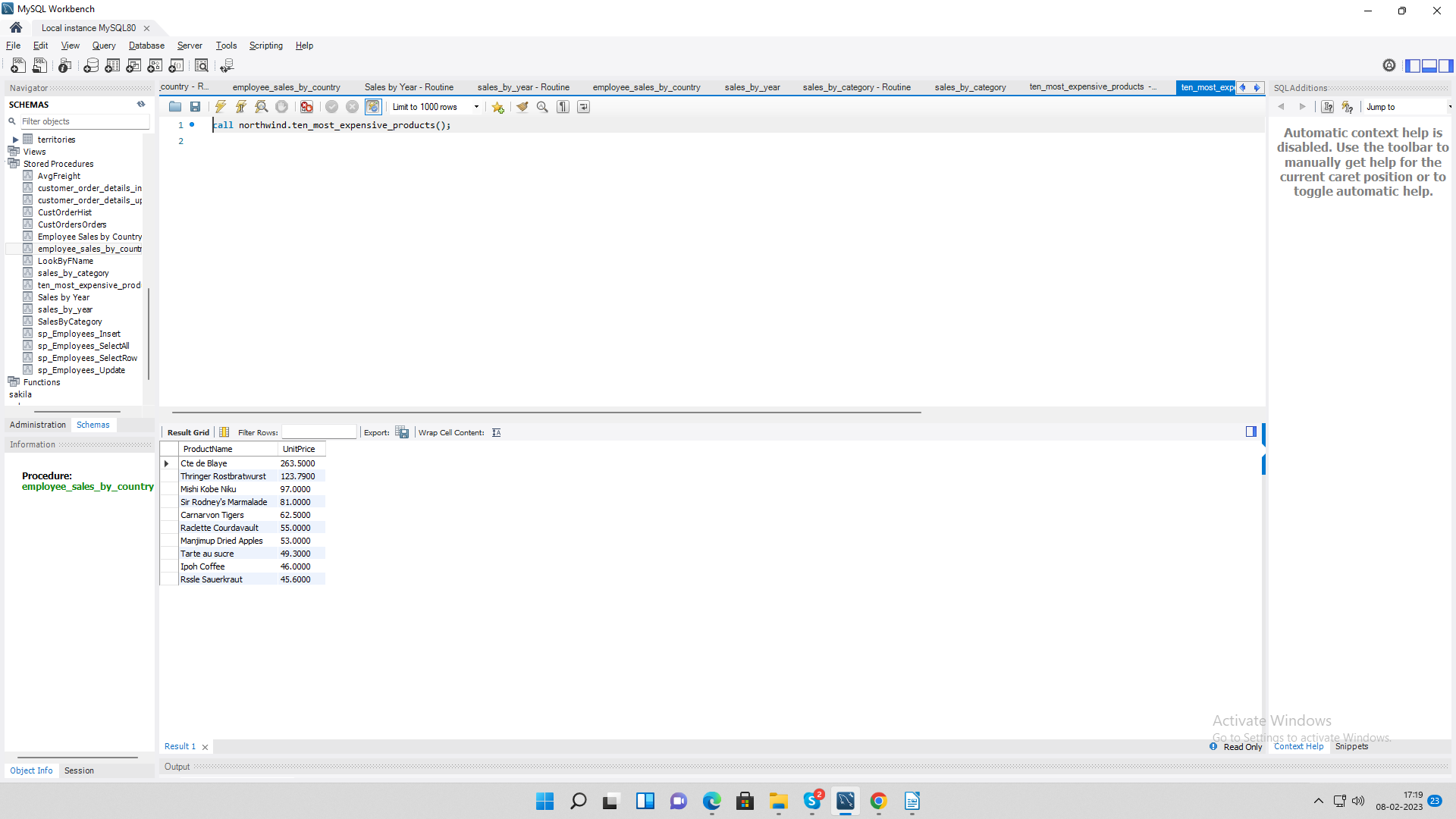


**5. Write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products**

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `ten\_most\_expensive\_products`()  
BEGIN  
SELECT ProductName, UnitPrice FROM Products ORDER BY UnitPrice DESC LIMIT 10;  
END

**Output:**

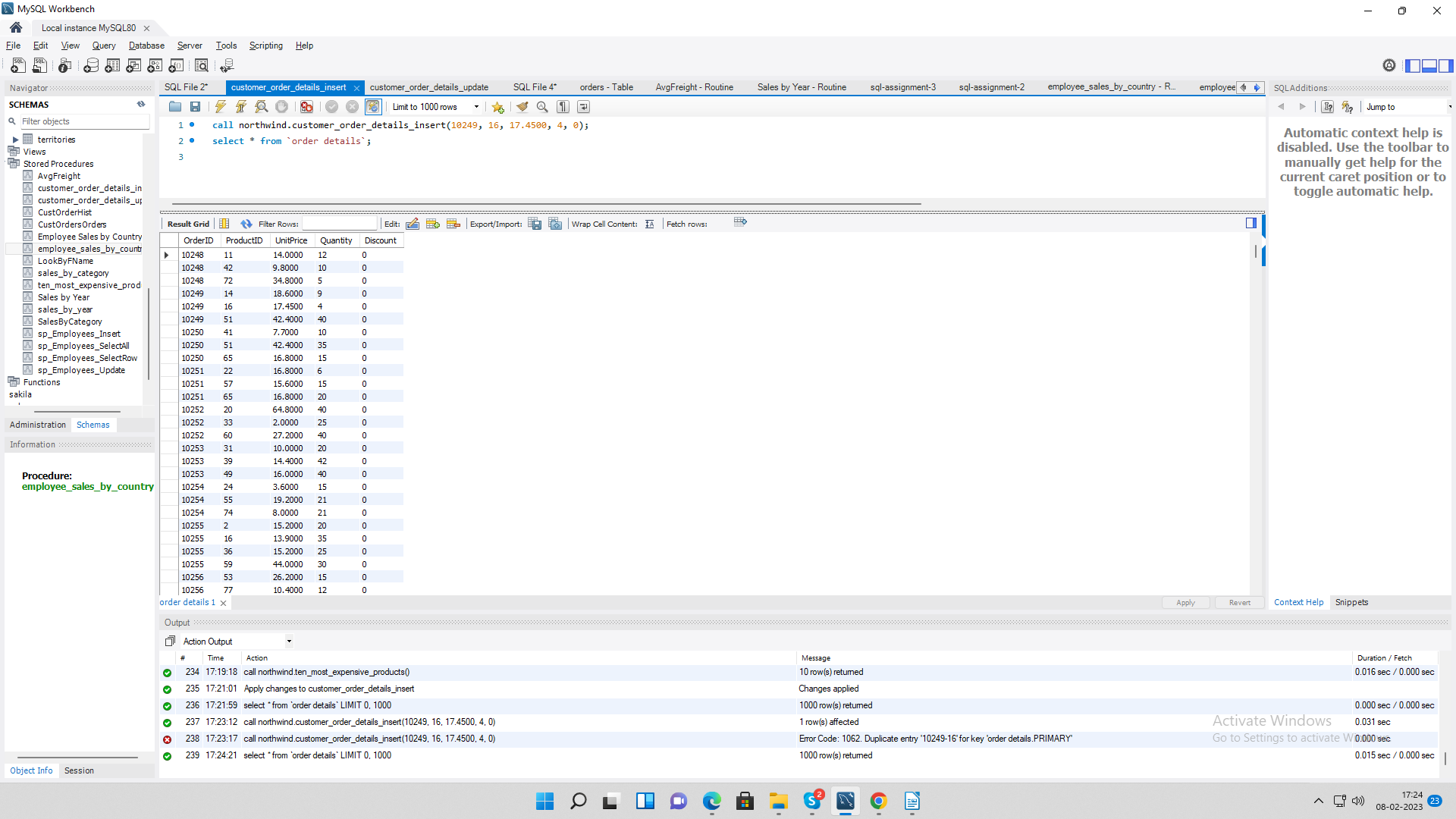


**6. Write a SQL query to Create Stored procedure in the Northwind database to insert**

**Customer Order Details**

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `customer\_order\_details\_insert`(  
In AtOrderID INT(20),  
In AtProductID INT(20),  
In AtUnitPrice DECIMAL(10,4),  
In AtQuantity SMALLINT,  
In AtDiscount DOUBLE(8,0)  
)  
BEGIN  
INSERT INTO `order details` VALUES(AtOrderID,AtProductID,AtUnitPrice,AtQuantity,AtDiscount);  
END

**Output:**

**7. Write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details**

**Query:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `customer\_order\_details\_update`(  
In AtOrderID INT(20),  
In AtProductID INT(20),  
In AtUnitPrice DECIMAL(10,4),  
In AtQuantity SMALLINT,  
In AtDiscount DOUBLE(8,0)  
)  
BEGIN  
UPDATE `order details`  
 SET  
 ProductID = AtProductID,  
 UnitPrice = AtUnitPrice,  
 Quantity = AtQuantity,  
 Discount = AtDiscount  
 WHERE  
 OrderID = AtOrderID;  
END

**Output:**

