**Instructions**

Run the script to create MyGuitarShop database. You will use this database for Assignments 2 and 3.

You will create a single Word document with all your statements and required screenshots.

**Requirements**

1. Create a view named OrderItemProducts that returns columns from the Orders, OrderItems, and Products tables.

 This view should return these columns from the Orders table: OrderID, OrderDate, TaxAmount, and ShipDate.

 This view should return these columns from the OrderItems table: ItemPrice, DiscountAmount, FinalPrice (the discount amount subtracted from the item price), Quantity, and ItemTotal (the calculated total for the item).

 This view should return the ProductName column from the Products table.

create view OrderItemProducts

as

select o.OrderID, convert(date, o.OrderDate) as OrderDate, o.TaxAmount,

convert(date, o.ShipDate) as ShipDate, oi.ItemPrice, oi.DiscountAmount,

SUM(oi.ItemPrice - oi.DiscountAmount) as FinalPrice, oi.Quantity,

SUM((oi.ItemPrice - oi.DiscountAmount) \* Quantity) as ItemTotal,

p.Productname

from Orders o join OrderItems oi on o.OrderID = oi.OrderID

join Products p on oi.ProductID = p.ProductID

group by o.OrderID, o.OrderDate, o.TaxAmount, o.ShipDate, oi.ItemPrice,

oi.DiscountAmount, oi.Quantity, p.ProductName

select \* from OrderItemProducts

2. Create a view named ProductSummary that uses the view you created in point 1. This view should return some summary information about each product.

 Each row should include these columns: ProductName, OrderCount (the number of times the product has been ordered), and OrderTotal (the total sales for the product).

Write a SELECT statement that uses the view that you created in point 2 to get total sales for the five best selling products. Take a screenshot that shows results returned from this query.

-- Create ProductSummary view

create view ProductSummary

as

select ProductName, COUNT(ProductName) as OrderCount, SUM(ItemTotal) as

OrderTotal

from OrderItemProducts

group by ProductName

select \* from ProductSummary

-- Retrieve five best selling products

select top 5 ProductName, OrderCount, OrderTotal

from ProductSummary

order by OrderTotal Desc

3. Suppose an application will often search Products based on ProductName. Create an index that makes such search more efficient. Take a screenshoot that shows your command succeeded.

create index INDX\_Products\_Name

on Products(ProductName)

select ProductName from Products

4. Alter table ORDERS to ensure that both ShipAddressID and BillingAddressID reference table Addresses. Take a screenshoot that shows your command(s) succeeded.

-- ShipAddressID Column from Orders Table will reference AddressID from Addresses Table

alter table Orders

add constraint FK\_Ord\_Ship\_Add foreign key(ShipAddressID) references Addresses(AddressID)

-- BillingAddressID Column from Orders Table will reference AddressID from Addresses Table

alter table Orders

add constraint FK\_Ord\_Bill\_Add foreign key(BillingAddressID) references Addresses(AddressID)