1. Create an SP that retrieves orders placed by a particular customer within a given date range.

```
use AdventureWorksLT2019;

select * from SalesLT.SalesOrderHeader;

-- creating SP that retrives orders placed by a particular customer within a given data range
--input: customerid, startdate, enddate

Go

CREATE PROCEDURE retrieve_orders
@Start_Date DATETIME,
@End_Date DATETIME,
@CusomterId int
AS

BEGIN

SELECT c.CustomerID, s.SalesOrderId, s.salesOrderNumber, s.OrderDate
FROM SalesLT.Customer c
inner join SalesLT.SalesOrderHeader s
ON c.CustomerID = s.CustomerID
WHERE c.CustomerID = @CusomterId AND s.OrderDate BETWEEN @Start_Date
AND @End_Date;

END;

--calling the SP
Execute retrieve_orders @Start_Date='2002-01-01 00:00:00', @End_Date= '2020-01-01 00:00:00', @CusomterId = 29847;
```

Output:

	CustomerID	SalesOrderId	salesOrderNumber	OrderDate
1	29847	71774	SO71774	2008-06-01 00:00:00.000

2. Create an SP that retrieves total sales for each product category.

```
--output: total sales

--creating table for input
create table TotalSaleForEachProductCategory(
ProductCategoryID int,
Name nvarchar(30),
TotalQuantity int
);

 GO

create procedure TotalSales
as

begin

truncate table TotalSaleForEachProductCategory;

--inserting into the table
insert into TotalSaleForEachProductCategory

select p.ProductCategoryID, pc.Name, sum(sod.OrderQty)
'TotalQuantity'
from SalesLT.Product p
join SalesLT.ProductCategory pc on
p.ProductCategoryID=pc.ProductCategoryID
join SalesLT.SalesOrderDetail sod on sod.ProductID=p.ProductID
group by p.ProductCategoryID, pc.Name;

end

--calling SP
exec TotalSales;

select * from TotalSaleForEachProductCategory;
```

Output:

Results | Messages

	ProductCategoryID	Name	TotalQuantity
1	5	Mountain Bikes	209
2	6	Road Bikes	222
3	7	Touring Bikes	252
4	8	Handlebars	27
5	9	Bottom Brackets	22
6	10	Brakes	13
7	11	Chains	8
8	12	Cranksets	22
9	13	Derailleurs	21
10	16	Mountain Frames	128
11	17	Pedals	84

3. Create an SP that retrieves the top 5 selling products in terms of quantity sold.

```
--o/p: top 5 most selling product in terms of quantity

--creating table
create table mostSelling(
Name nvarchar(40),
TotalQty int,
);

 Go

Alter procedure TopSelling
as
begin
truncate table mostSelling;
insert into mostSelling (Name, TotalQty)
        select top 5 p.Name, sum(sod.OrderQty) 'TotalQty'
        from SalesLT.Product p
        join SalesLT.SalesOrderDetail sod on p.ProductID=sod.ProductID
        group by p.Name
        order by sum(sod.OrderQty) desc;

end


--calling Sp
exec TopSelling;

select * from mostSelling;
```

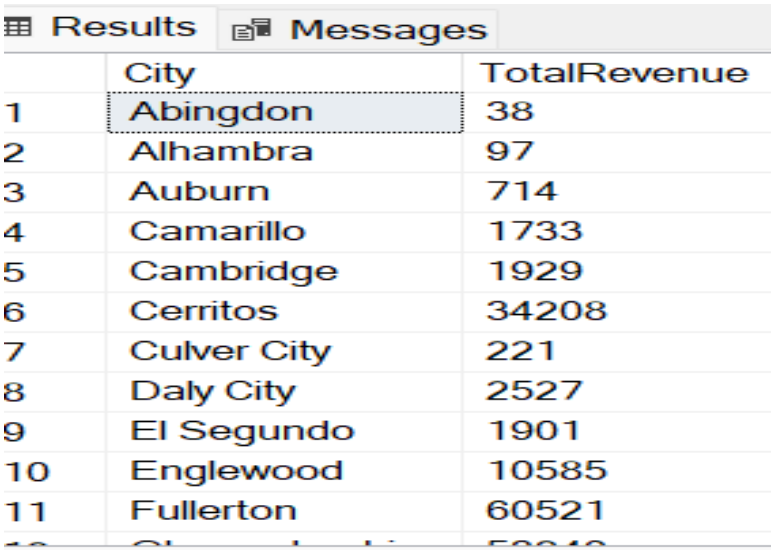
Output:

	Name	TotalQty
1	Classic Vest, S	87
2	Short-Sleeve Classic Jersey, XL	57
3	Bike Wash - Dissolver	55
4	Water Bottle - 30 oz.	54
5	AWC Logo Cap	52

4. Create an SP that retrieves revenue generated by each sales territory.

```
--creating table

create table Revenue(
City nvarchar(30),
TotalRevenue int
);

--Sp
--O/P: Revenue generated by each sales territory
Go

create procedure geteachsaleRevenue
as
begin
truncate table Revenue;
insert into Revenue (City, TotalRevenue)

        select a.city, SUM(sod.OrderQty * (sod.UnitPrice-
        sod.UnitPriceDiscount)) as TotalRevenue

        from SalesLT.SalesOrderDetail sod
        join SalesLT.SalesOrderHeader soh on

        soh.SalesOrderID=sod.SalesOrderID

        join SalesLT.CustomerAddress ca on

        ca.CustomerID=soh.CustomerID

        join SalesLT.Address a on a.AddressID = ca.AddressID
        group by a.City;

end

--calling sp
go
exec geteachsaleRevenue;

--displaying the result
select * from Revenue;
```

Output:



Results Messages

	City	TotalRevenue
1	Abingdon	38
2	Alhambra	97
3	Auburn	714
4	Camarillo	1733
5	Cambridge	1929
6	Cerritos	34208
7	Culver City	221
8	Daly City	2527
9	El Segundo	1901
10	Englewood	10585
11	Fullerton	60521

5. Create an SP that retrieves a customer's order history, including order date, order number, and total amount spent.

```
--creating table

create table customer_order_history(
CustomerID int,
OrderDate datetime,
PurchaseOrderNumber nvarchar(25),
TotalDue money
);

--i/p: customerID
--o/p: customer_order_history
Go

create procedure get_order_history
@CustomerID int
as
begin
        truncate table customer_order_history;
        insert into customer_order_history
        (CustomerID, OrderDate,
        PurchaseOrderNumber, TotalDue)

        select soh.CustomerID, soh.OrderDate, soh.PurchaseOrderNumber,
        soh.TotalDue
        from SalesLT.SalesOrderHeader soh
        where soh.CustomerID = @CustomerID;

end

--calling SP
go
exec get_order_history
@CustomerID=29847

--displaying stored result in the table
select * from customer_order_history;
```

Output:

Results | Messages

	CustomerID	OrderDate	PurchaseOrderNumber	TotalDue
1	29847	2008-06-01 00:00:00.000	PO348186287	972.785

6. Create an SP that retrieves total sales for each product by month.

```
--create Table
create table salesforproductforgivenmonth(
Name nvarchar(50),
SalesMonth int,
TotalSales numeric(38, 6)
);


--o/p: total sales
Go

Alter PROCEDURE total_sales_of_products_for_given_month


AS
BEGIN

SET NOCOUNT ON;

truncate table salesforproductforgivenmonth
insert into salesforproductforgivenmonth
(Name, SalesMonth,
TotalSales)

        select p.Name, month(soh.OrderDate) 'SalesMonth',
        sum(soh.SubTotal) 'TotalSales'
        from SalesLT.Product p
        join SalesLT.SalesOrderDetail sod on p.ProductID=sod.ProductID
        join SalesLT.SalesOrderHeader soh on
        sod.SalesOrderID=soh.SalesOrderID
        group by p.Name, MONTH(soh.OrderDate)
        --order by SalesMonth, p.Name;
        order by 3 desc

END

--calling SP
go
exec total_sales_of_products_for_given_month ;

--displaying results
select * from salesforproductforgivenmonth;
```

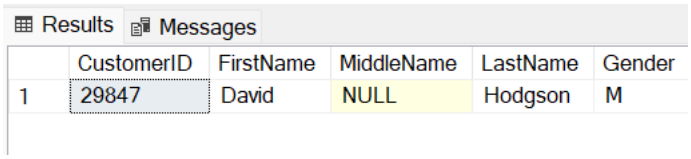
Output:

Name	SalesMonth	TotalSales
AWC Logo Cap	6	503014.712400
Bike Wash - Dissolver	6	486930.656600
Chain	6	217705.438800
Classic Vest, M	6	473106.948300
Classic Vest, S	6	489570.756300
Front Brakes	6	273574.825000
Front Derailleur	6	258092.963800
Half-Finger Gloves, L	6	260166.806900
Half-Finger Gloves, M	6	473106.948300
Half-Finger Gloves, S	6	270449.947600
Hitch Rack - 4-Bike	6	476638.127200

7. Create an SP that retrieves customer demographics such as age, gender, and income.

```
--creating table

create table customer_information(
CustomerID int,
FirstName nvarchar(50),
MiddleName nvarchar(50),
LastName nvarchar(50),
Gender nvarchar(10)
);

--i/p: customer Id
--o/p: customer information
Go

create procedure get_demography
@CID int
as
begin

truncate table customer_information;
insert into customer_information
(CustomerID, FirstName, MiddleName,
LastName, Gender)

select CustomerID, FirstName, MiddleName, LastName,
case when Title = 'Mr.' then 'M'
when Title = 'Ms.' then 'F'

else 'Unknown'
end 'Gender'
from SalesLT.Customer
where CustomerID=@CID;
end

go

--calling sp
exec get_demography
@cid=29847

--displaying results
select * from customer_information;
```

Output:



Results Messages

	CustomerID	FirstName	MiddleName	LastName	Gender
1	29847	David	NULL	Hodgson	M