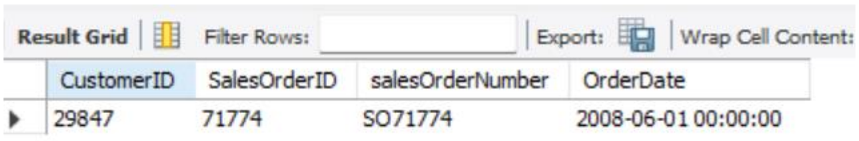
# Stored Procedure in MY SQL

1. creating SP that retrives orders placed by a particular customer within a given date range

```
DELIMITER //

create PROCEDURE retrieve_orders(
   IN Start_Date DATETIME,
   IN End_Date DATETIME,
   IN CusomterId INT
)
BEGIN
   SELECT c.CustomerID, s.SalesOrderID, s.salesOrderNumber, s.OrderDate
   FROM customer c
   INNER JOIN salesorderheader s ON c.CustomerID = s.CustomerID
   WHERE c.CustomerID = CusomterId AND s.OrderDate BETWEEN Start_Date
AND End_Date;
END//

DELIMITER ;

-- Calling the stored procedure
CALL retrieve_orders('2002-01-01 00:00:00', '2020-01-01 00:00:00', 29847);
```

Outputs:

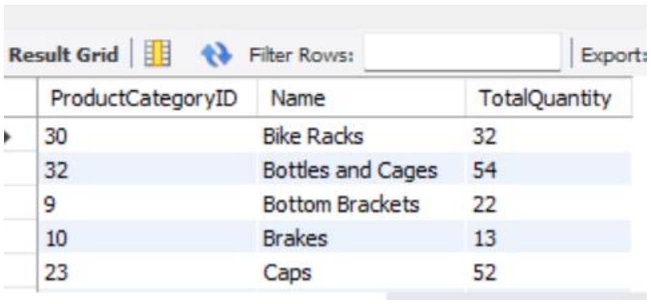


| CustomerID | SalesOrderID | salesOrderNumber | OrderDate |
|---|---|---|---|
| 29847 | 71774 | SO71774 | 2008-06-01 00:00:00 |

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

```
CREATE TABLE TotalSaleForEachProductCategory (
   ProductCategoryID INT,
   Name VARCHAR(30),
   TotalQuantity INT
);

DELIMITER //

CREATE PROCEDURE TotalSales()
BEGIN
   TRUNCATE TABLE TotalSaleForEachProductCategory;

   -- Inserting into the table
   INSERT INTO TotalSaleForEachProductCategory
   SELECT p.ProductCategoryID, pc.Name, SUM(sod.OrderQty) AS 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//

DELIMITER ;

-- Calling the stored procedure
CALL TotalSales();

SELECT * FROM TotalSaleForEachProductCategory;
```

Output:

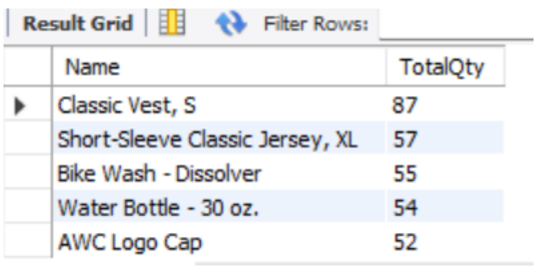


| ProductCategoryID | Name | TotalQuantity |
|---|---|---|
| 30 | Bike Racks | 32 |
| 32 | Bottles and Cages | 54 |
| 9 | Bottom Brackets | 22 |
| 10 | Brakes | 13 |
| 23 | Caps | 52 |

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

```
CREATE TABLE mostSelling (
  Name VARCHAR(40),
  TotalQty INT
);

DELIMITER //

CREATE PROCEDURE TopSelling1()
BEGIN
  TRUNCATE TABLE mostSelling;

  INSERT INTO mostSelling (Name, TotalQty)
  SELECT p.Name, SUM(sod.OrderQty) AS TotalQty
  FROM Product p
  JOIN SalesOrderDetail sod ON p.ProductID = sod.ProductID
  GROUP BY p.Name
  ORDER BY SUM(sod.OrderQty) DESC
  LIMIT 5;
END//

DELIMITER ;

CALL TopSelling1();

SELECT * FROM mostSelling;
```

Output:

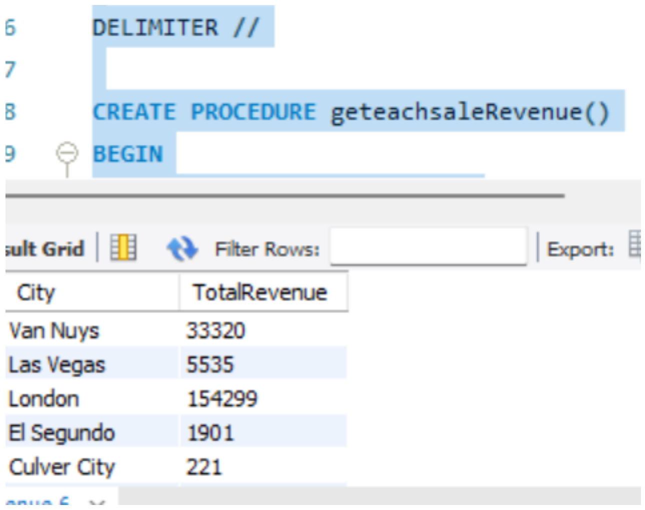


| Name | TotalQty |
|---|---|
| Classic Vest, S | 87 |
| Short-Sleeve Classic Jersey, XL | 57 |
| Bike Wash - Dissolver | 55 |
| Water Bottle - 30 oz. | 54 |
| AWC Logo Cap | 52 |

4. SP that retrieves revenue generated by each sales territory

```
CREATE TABLE Revenue (
   City VARCHAR(30),
   TotalRevenue INT
);

DELIMITER //

CREATE PROCEDURE geteachsaleRevenue()
BEGIN
   TRUNCATE TABLE Revenue;

   INSERT INTO Revenue (City, TotalRevenue)
   SELECT a.City, SUM(sod.OrderQty * (sod.UnitPrice - sod.UnitPriceDiscount)) AS
TotalRevenue
   FROM SalesOrderDetail sod
   JOIN SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID
   JOIN CustomerAddress ca ON ca.CustomerID = soh.CustomerID
   JOIN Address a ON a.AddressID = ca.AddressID
   GROUP BY a.City;
END//

DELIMITER ;

CALL geteachsaleRevenue();

SELECT * FROM Revenue;
```

Output:



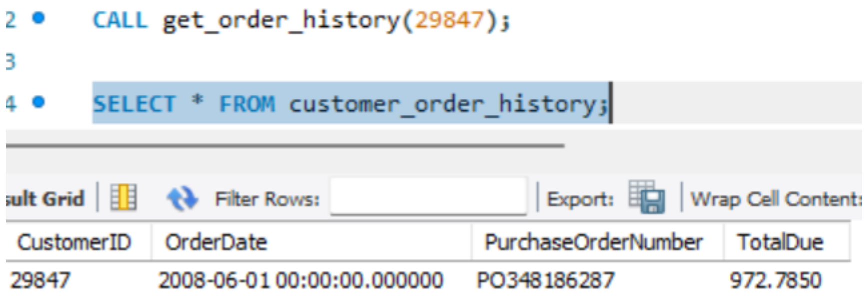
```
DELIMITER //

CREATE PROCEDURE geteachsaleRevenue()
BEGIN
```

| City | TotalRevenue |
|---|---|
| Van Nuys | 33320 |
| Las Vegas | 5535 |
| London | 154299 |
| El Segundo | 1901 |
| Culver City | 221 |

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

```
CREATE TABLE customer_order_history (
   CustomerID INT,
   OrderDate DATETIME,
   PurchaseOrderNumber VARCHAR(25),
   TotalDue DECIMAL(19,4)
);

DELIMITER //

CREATE PROCEDURE get_order_history(IN CustomerID INT)
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 SalesOrderHeader soh
   WHERE soh.CustomerID = CustomerID;
END//

DELIMITER ;

CALL get_order_history(29847);

SELECT * FROM customer_order_history;
```

Output:



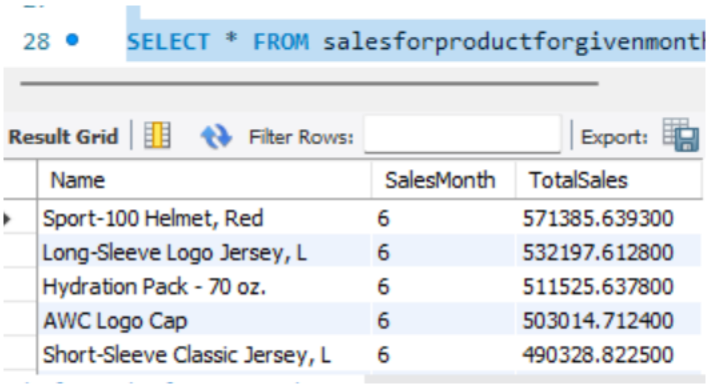
```
CALL get_order_history(29847);

SELECT * FROM customer_order_history;
```

| CustomerID | OrderDate | PurchaseOrderNumber | TotalDue |
|---|---|---|---|
| 29847 | 2008-06-01 00:00:00.000000 | PO348186287 | 972.7850 |

5. SP that retrieves total sales for each product by month

```
CREATE TABLE salesforproductforgivenmonth (
   Name VARCHAR(50),
   SalesMonth INT,
   TotalSales DECIMAL(38, 6)
);

DELIMITER //

CREATE PROCEDURE total_sales_of_products_for_given_month()
BEGIN


   TRUNCATE TABLE salesforproductforgivenmonth;

   INSERT INTO salesforproductforgivenmonth (Name, SalesMonth, TotalSales)
   SELECT p.Name, MONTH(soh.OrderDate) AS SalesMonth, SUM(soh.SubTotal) AS TotalSales
   FROM Product p
   JOIN SalesOrderDetail sod ON p.ProductID = sod.ProductID
   JOIN SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
   GROUP BY p.Name, MONTH(soh.OrderDate)
   ORDER BY 3 DESC;
END//

DELIMITER ;

CALL total_sales_of_products_for_given_month();

SELECT * FROM salesforproductforgivenmonth;
```

Output:



```
SELECT * FROM salesforproductforgivenmontl
```

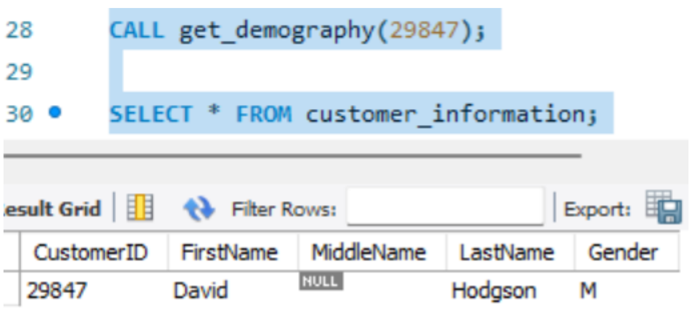
Result Grid | Filter Rows: | Export:

| Name | SalesMonth | TotalSales |
|---|---|---|
| Sport-100 Helmet, Red | 6 | 571385.639300 |
| Long-Sleeve Logo Jersey, L | 6 | 532197.612800 |
| Hydration Pack - 70 oz. | 6 | 511525.637800 |
| AWC Logo Cap | 6 | 503014.712400 |
| Short-Sleeve Classic Jersey, L | 6 | 490328.822500 |

6. SP that retrieves customer demographics such as age, gender, and income

```
CREATE TABLE customer_information (
   CustomerID INT,
   FirstName VARCHAR(50),
   MiddleName VARCHAR(50),
   LastName VARCHAR(50),
   Gender VARCHAR(10)
);

DELIMITER //

CREATE PROCEDURE get_demography(IN CID INT)
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 AS Gender
   FROM Customer
   WHERE CustomerID = CID;
END//

DELIMITER ;

CALL get_demography(29847);

SELECT * FROM customer_information;
```

Output:



```
CALL get_demography(29847);

SELECT * FROM customer_information;
```

| CustomerID | FirstName | MiddleName | LastName | Gender |
|---|---|---|---|---|
| 29847 | David | NULL | Hodgson | M |