**Problem 1:** Display the bike names and list prices for bikes costing $5,000 or more. Order the results in descending order based on the list price.

USE BikeStores

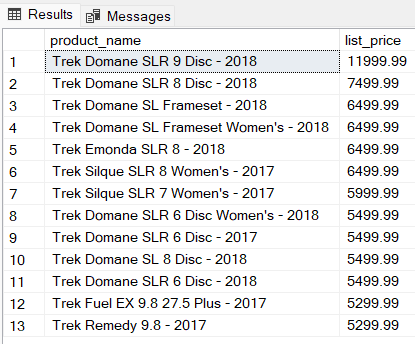
GO

SELECT product\_name, list\_price

FROM Production.Products

WHERE list\_price >= 5000

ORDER BY list\_price DESC;



**Problem 2:** Present a list of the bikes made by a company called “Haro” in alphabetical order.

USE BikeStores

GO

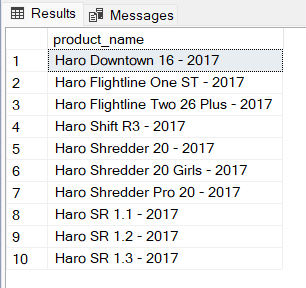
SELECT product\_name

FROM Production.Products INNER JOIN Production.Brands

ON Production.Products.brand\_id = Production.Brands.brand\_id

WHERE Production.Brands.brand\_name = 'Haro'

ORDER BY product\_name;



**Problem 3**: Select customer records for customers residing in the **14450** zip code.

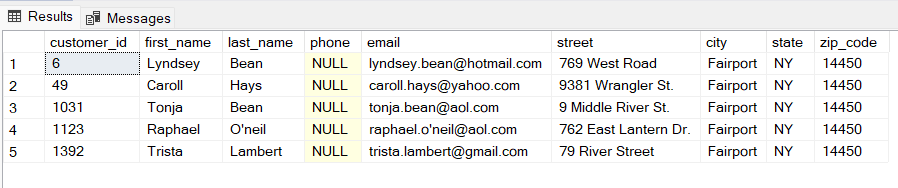
USE BikeStores

GO

SELECT \*

FROM Sales.Customers

WHERE zip\_code = '14450'



**Problem 4**: List the first and last names of staff members whose manager is Mireya Copeland.

USE BikeStores

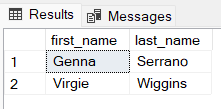
GO

SELECT Sales.Staffs.first\_name, Sales.Staffs.last\_name

FROM Sales.Staffs INNER JOIN Sales.Staffs M

ON Sales.Staffs.manager\_id = M.staff\_id

WHERE M.first\_name = 'Mireya' AND M.last\_name = 'Copeland'



**Problem 5**: The company is now selling bikes from a company called **GT Bicycles**. Enter a new record into the database to reflect this change.

USE BikeStores

GO

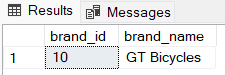
INSERT INTO Production.Brands (brand\_name)

VALUES ('GT Bicycles');

SELECT \*

FROM Production.brands

WHERE brand\_name = 'GT Bicycles'



**Problem 6**: Update the amount of stock for the bike with product\_id = 198 at the store with store\_id = 2. The new stock amount should be 20.

USE BikeStores

GO

UPDATE Production.stocks

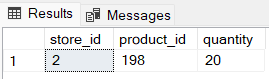
SET quantity = 20

WHERE product\_id = 198 AND store\_id= 2;

SELECT \*

FROM Production.stocks

WHERE product\_id = 198 AND store\_id= 2;



**Problem 7**: Show bike names and the total sales amounts for transactions involving bikes costing $10,000 or more. The total sales amount is the product of the bike order item list price, quantity sold, and sale discount.

USE BikeStores

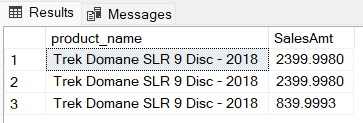
GO

SELECT P.product\_name, OI.list\_price\*quantity\*discount AS SalesAmt

FROM Sales.Order\_Items OI INNER JOIN Production.Products P

ON OI.product\_id = P.product\_id

WHERE OI.list\_price >= 10000;



**Problem 8**: Provide a distinct list of the states in alphabetical order where the company’s customers reside.

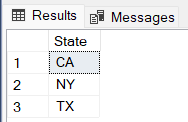
USE BikeStores

GO

SELECT DISTINCT State

FROM Sales.Customers

ORDER BY State



**Problem 9**: Display the full names of all staff personnel in First\_Name, “ ” , Last\_Name format. Use the built-in **CONCAT()** function for this query.

USE BikeStores

GO

SELECT First\_Name + ' ' + Last\_Name AS 'Staff Name'

FROM Sales.Staffs;

SELECT CONCAT (First\_Name, ' ', Last\_Name) AS 'Staff Name'

FROM Sales.Staffs;



**Problem 10**: Show the bikes that have zero stock at the Texas (TX) store.

USE BikeStores

GO

SELECT product\_name, state, quantity

FROM Production.Products P INNER JOIN Production.Stocks S

ON P.product\_id = S.product\_id

INNER JOIN Sales.Stores ST

ON S.store\_id = ST.store\_id

WHERE ST.state = 'TX' AND S.quantity = 0

ORDER BY product\_name;

