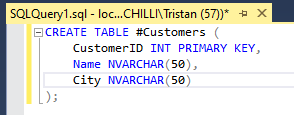
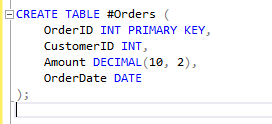
**CREATION**

I’ve created a temporary table called Customers and Orders. Please note that orders have a foreign key ‘CustomerId’. It can’t directly be declared as a foreign key as it is a Temporary table.





**INSERT**

Inserting into the tables with the following statements:

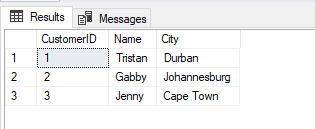
INSERT INTO #Customers VALUES (1, 'Tristan', 'Durban'), (2, 'Gabby', 'Johannesburg'), (3, 'Jenny', 'Cape Town');

INSERT INTO #Orders VALUES (101, 1, 250.00, '2024-02-01'), (102, 2, 150.00, '2024-02-05'), (103, 1, 450.00, '2024-02-10'), (104, 3, 300.00, '2024-02-12');

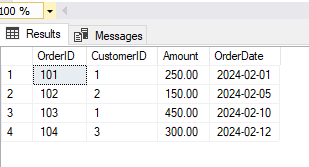
**SELECT**

Selecting everything from the Customers and Orders table with the following statements to ensure our records were added successfully:

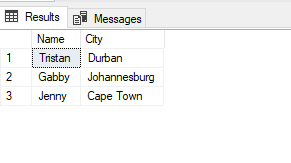
SELECT \* FROM #Customers



SELECT \* FROM #Orders



The \* Selects all the columns in the table. When working with large amounts of data this can sometimes become difficult to read. We can then select specific columns with the following statement:

SELECT [Name], City FROM #Customers  
  


**JOINS**

We have a few joins in SQL some of them are:  
  
**INNER JOIN:**

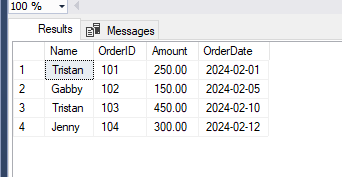
Returns only the rows where there is a match in both tables.

For example: “Give me only customers who have placed orders.”

SELECT c.Name, o.OrderId, o.Amount, o.OrderDate

FROM #Customers c

JOIN #Orders o ON c.CustomerId = o.CustomerId;



**LEFT JOIN (LEFT OUTER JOIN):**

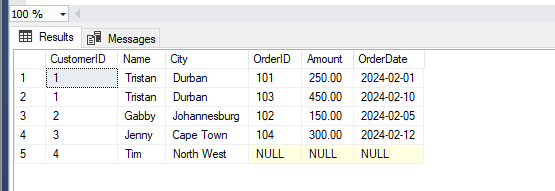
Returns all rows from the left table and matched rows from the right table.

For example: *"*Give me all customers, even those with no orders."

SELECT c.CustomerId, c.Name, c.City, o.OrderId, o.Amount, o.OrderDate

FROM #Customers c

LEFT JOIN #Orders o ON c.CustomerId = o.CustomerId;



There is a customer with no order displayed.

**RIGHT JOIN (RIGHT OUTER JOIN):**

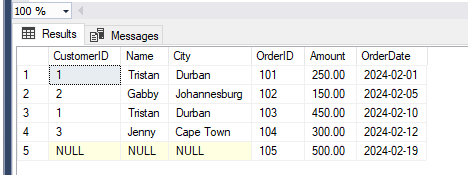
Returns all rows from the right table and matched rows from the left table.

For example: "Give me all orders, even if there’s no matching customer."

SELECT c.CustomerId, c.Name, c.City, o.OrderId, o.Amount, o.OrderDate

FROM #Customers c

RIGHT JOIN #Orders o ON c.CustomerId = o.CustomerId;



There is an order with no customer associated with it, and we cant see Tim from the above table.