T-SQL Exercises

UDF and TVF Exercise

```
USE TSQL2018;
GO
```

-- Task 1

-- Create a User Defined Function called getEmpName in the HR schema that has an integer input parameter called @empid

-- The function returns the concatenated firstname and lastname values from HR.Employees for the employee whose empid matches the parameter value

```
CREATE FUNCTION HR.getEmpName(@empid INT)
RETURNS TABLE
AS
RETURN
    SELECT CONCAT(firstname, ' ', lastname) AS fullname
    FROM HR.Employees
    WHERE empid = @empid;
GO
```

-- Task 2

-- Write a SELECT statement against the Sales.OrderValues view to retrieve the custid and total sum of the val column when grouped by custid.

-- Filter the results to include orders only for the order year 2016.

```
SELECT custid, SUM(val) AS total
FROM Sales.OrderValues
WHERE YEAR(orderdate) = 2016
```

```
GROUP BY custid;
GO
```

-- Define an inline table-valued function using the following function header and add your previous query after the RETURN clause.

-- Include an integer input parameter called @orderyear

-- modify the query by replacing the '2016' in the WHERE clause with the parameter @orderyear.

```
CREATE FUNCTION dbo.fnGetSalesByCustomer(@orderyear INT)
RETURNS TABLE
AS
RETURN
    SELECT custid, SUM(val) AS totalsalesamount
    FROM Sales.OrderValues
    WHERE YEAR(orderdate) = @orderyear
    GROUP BY custid;
GO
```

-- Highlight the completed code and execute it. This will create an inline table-valued function named dbo.fnGetSalesByCustomer.

-- Write a SELECT statement to retrieve the custid and totalsalesamount columns from the dbo.fnGetSalesByCustomer inline table-valued function.

-- Use the value 2015 for the parameter and execute the query

```
SELECT *
FROM dbo.fnGetSalesByCustomer(2015);
GO
```

-- Change the value to 2016 for the parameter then execute the query again

```
SELECT *
FROM dbo.fnGetSalesByCustomer(2016);
GO
```

-- Bonus: Alter the dbo.fnGetSalesByCustomer function using two input parameters one for the year and one for the custid

```
ALTER FUNCTION dbo.fnGetSalesByCustomer(@orderyear INT, @custid INT)
RETURNS TABLE
AS
RETURN
	SELECT custid, SUM(val) AS totalsalesamount
	FROM Sales.OrderValues
	WHERE YEAR(orderdate) = @orderyear AND custid = @custid
	GROUP BY custid;
GO
```

-- test the new function passing two arguments

```
SELECT *
FROM dbo.fnGetSalesByCustomer(2016, 2);
GO
```