# Instructions N2 Derivco Assessment for Ayanda Langa

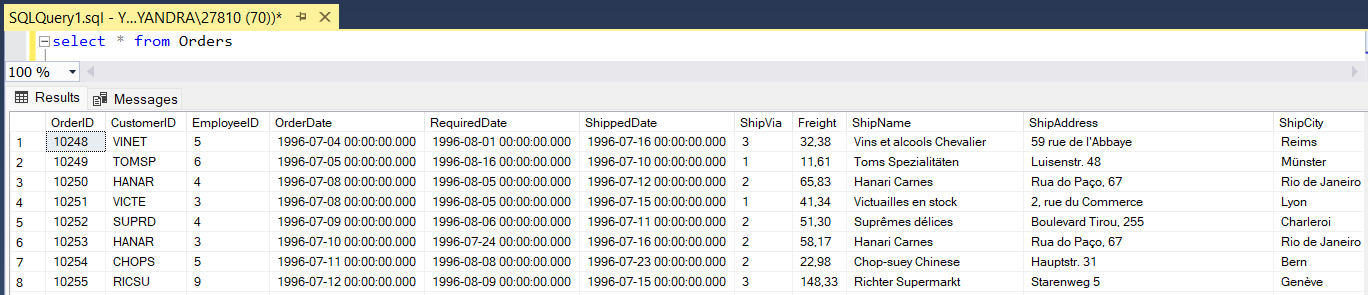
Please review using this repo: <https://github.com/ayahlangagthb/n2-derivco-al-repo>

Installed Applications and frameworks used:

* VS Code for C#
* SQL Server SSMS
* Northwind Sample DB

# Assessment 1

Northwind db snapshot for db.Orders Table



## Procedure:

USE [Northwind]

GO

CREATE PROCEDURE pr\_GetOrderSummary

@StartDate DATETIME,

@EndDate DATETIME,

@EmployeeID INT = NULL,

@CustomerID NVARCHAR(5) = NULL

AS

BEGIN

SELECT

CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS EmployeeFullName,

s.CompanyName AS ShipperCompanyName,

c.CompanyName AS CustomerCompanyName,

COUNT(o.OrderID) AS NumberOfOrders,

CONVERT(DATE, o.OrderDate) AS [Date],

SUM(o.Freight) AS TotalFreightCost,

COUNT(DISTINCT od.ProductID) AS NumberOfDifferentProducts,

SUM(od.UnitPrice \* od.Quantity \* (1 - od.Discount)) AS TotalOrderValue

FROM

Orders o

JOIN Employees e ON o.EmployeeID = e.EmployeeID

JOIN Shippers s ON o.ShipVia = s.ShipperID

JOIN Customers c ON o.CustomerID = c.CustomerID

JOIN [Order Details] od ON o.OrderID = od.OrderID

WHERE

(@EmployeeID IS NULL OR o.EmployeeID = @EmployeeID) AND

(@CustomerID IS NULL OR o.CustomerID = @CustomerID) AND

o.OrderDate BETWEEN @StartDate AND @EndDate

GROUP BY

CONVERT(DATE, o.OrderDate),

e.EmployeeID,

CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName),

s.ShipperID,

s.CompanyName,

c.CustomerID,

c.CompanyName

ORDER BY

[Date],

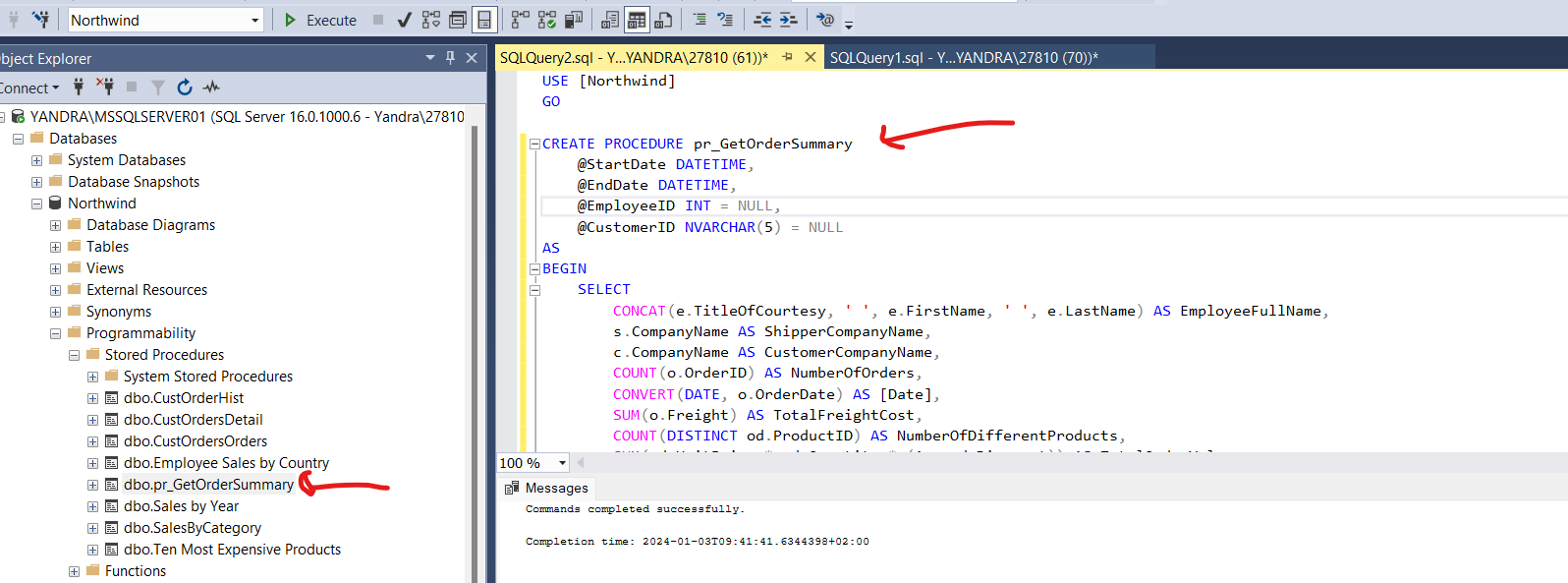
EmployeeFullName,

CustomerCompanyName,

ShipperCompanyName;

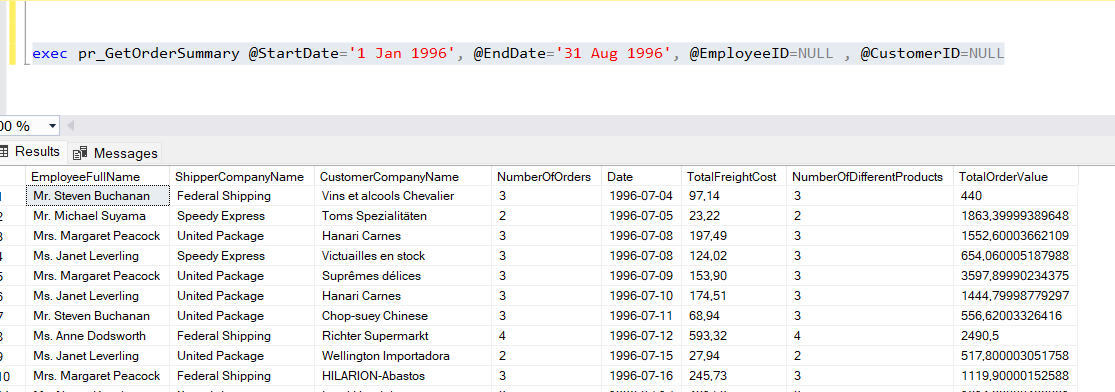
END

## Output snapshot:

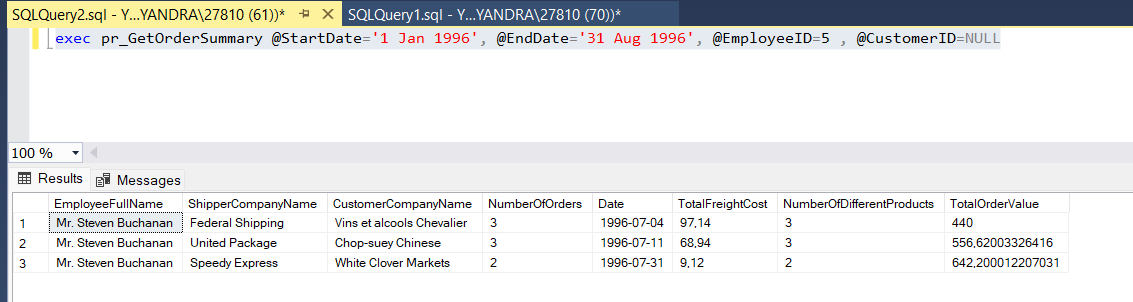


## Test cases

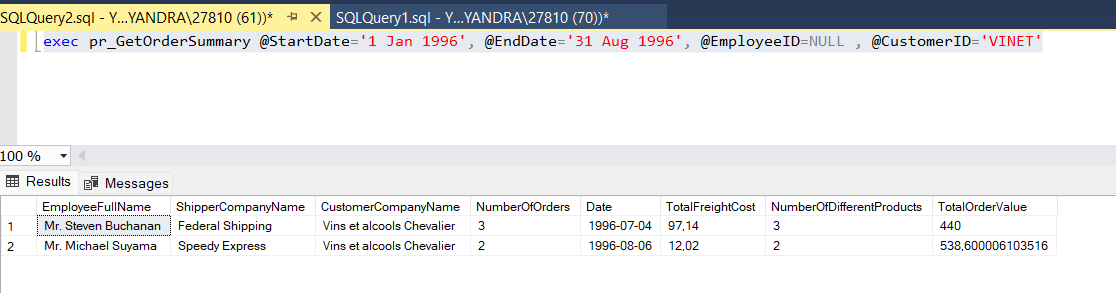
* exec pr\_GetOrderSummary @StartDate='1 Jan 1996', @EndDate='31 Aug 1996', @EmployeeID=NULL , @CustomerID=NULL



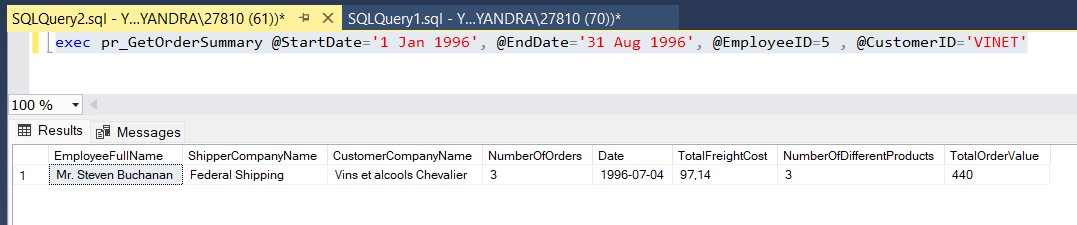
* exec pr\_GetOrderSummary @StartDate='1 Jan 1996', @EndDate='31 Aug 1996', @EmployeeID=5 , @CustomerID=NULL



* exec pr\_GetOrderSummary @StartDate='1 Jan 1996', @EndDate='31 Aug 1996', @EmployeeID=NULL , @CustomerID='VINET'

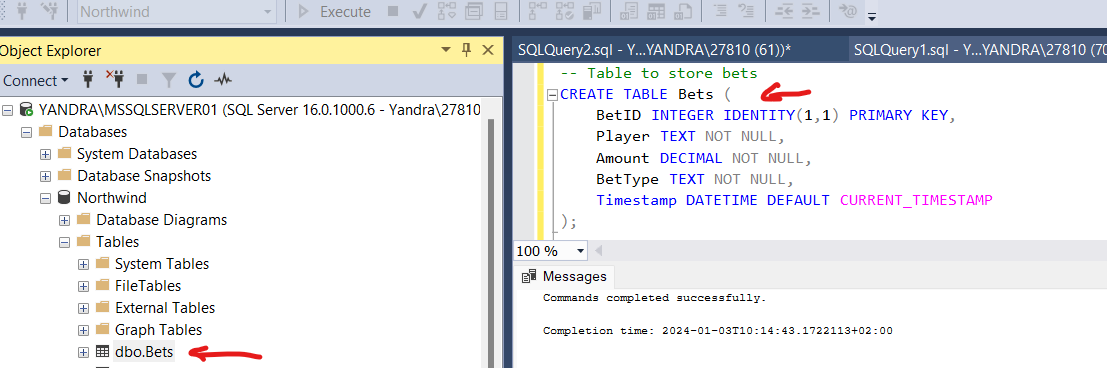


* exec pr\_GetOrderSummary @StartDate='1 Jan 1996', @EndDate='31 Aug 1996', @EmployeeID=5 , @CustomerID='VINET'



# Assessment 2

## Tables required for this:



-- Table to store bets

CREATE TABLE Bets (

BetID INTEGER IDENTITY(1,1) PRIMARY KEY,

Player VARCHAR(255) NOT NULL,

BetNumber INTEGER NOT NULL,

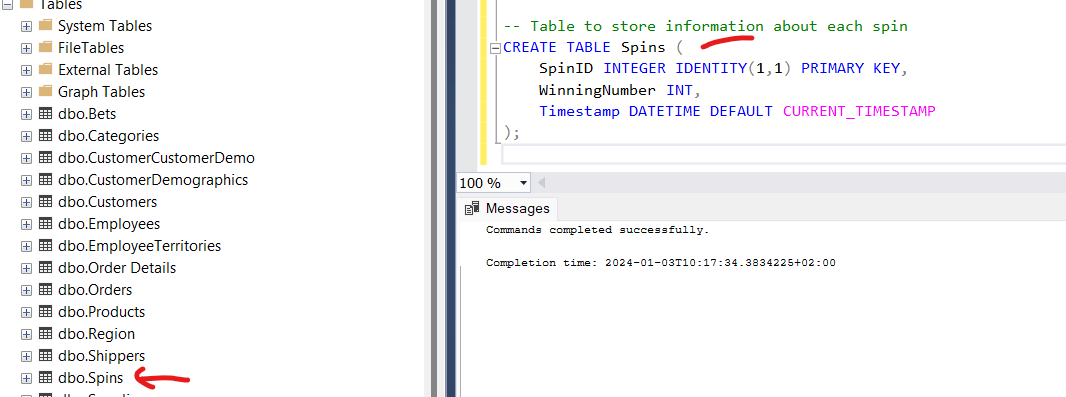
Amount DECIMAL NOT NULL,

BetType VARCHAR(255) NOT NULL,

Payout DECIMAL NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);



-- Table to store information about each spin

CREATE TABLE Spins (

SpinID INTEGER IDENTITY(1,1) PRIMARY KEY,

WinningNumber INT,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);

## Procedures required for this:

USE [Northwind]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[pr\_PlaceBet] Script Date: 03/01/2024 17:41:54 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- Stored Procedure to place a bet

ALTER PROCEDURE [dbo].[pr\_PlaceBet]

@Player TEXT,

@Amount DECIMAL,

@BetNumber INTEGER,

@BetType TEXT

AS

BEGIN

INSERT INTO Bets (Player, Amount, BetType, BetNumber) VALUES (@Player, @Amount, @BetType, @BetNumber);

END;

-- Stored Procedure to perform a roulette spin

CREATE PROCEDURE pr\_SpinRoulette

@WinningNumber INT

AS

BEGIN

INSERT INTO Spins (WinningNumber) VALUES (@WinningNumber);

END;

USE [Northwind]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[pr\_Payout] Script Date: 03/01/2024 16:49:51 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- Stored Procedure to calculate and perform payouts

CREATE PROCEDURE [dbo].[pr\_Payout]

AS

BEGIN

WITH CTE AS (

SELECT TOP 1 WinningNumber

FROM Spins

ORDER BY Timestamp DESC

)

-- Payout logic based on the winning number

UPDATE Bets

SET Payout =

CASE

WHEN BetType = 'Even' AND CTE.WinningNumber % 2 = 0 THEN Amount \* 2 -- Example: Even bets

WHEN BetType = 'Odd' AND CTE.WinningNumber % 2 <> 0 THEN Amount \* 2 -- Example: Odd bets

WHEN BetType = 'Number' AND BetNumber = CTE.WinningNumber THEN Amount \* 36 -- Example: Specific number bets

ELSE 0 -- No payout for other bets

END

FROM CTE

WHERE Payout IS NULL; -- Only update bets that haven't been paid out yet

SELECT TOP 1 Payout

FROM Bets

ORDER BY Timestamp DESC

END

-- Stored Procedure to retrieve previous spins

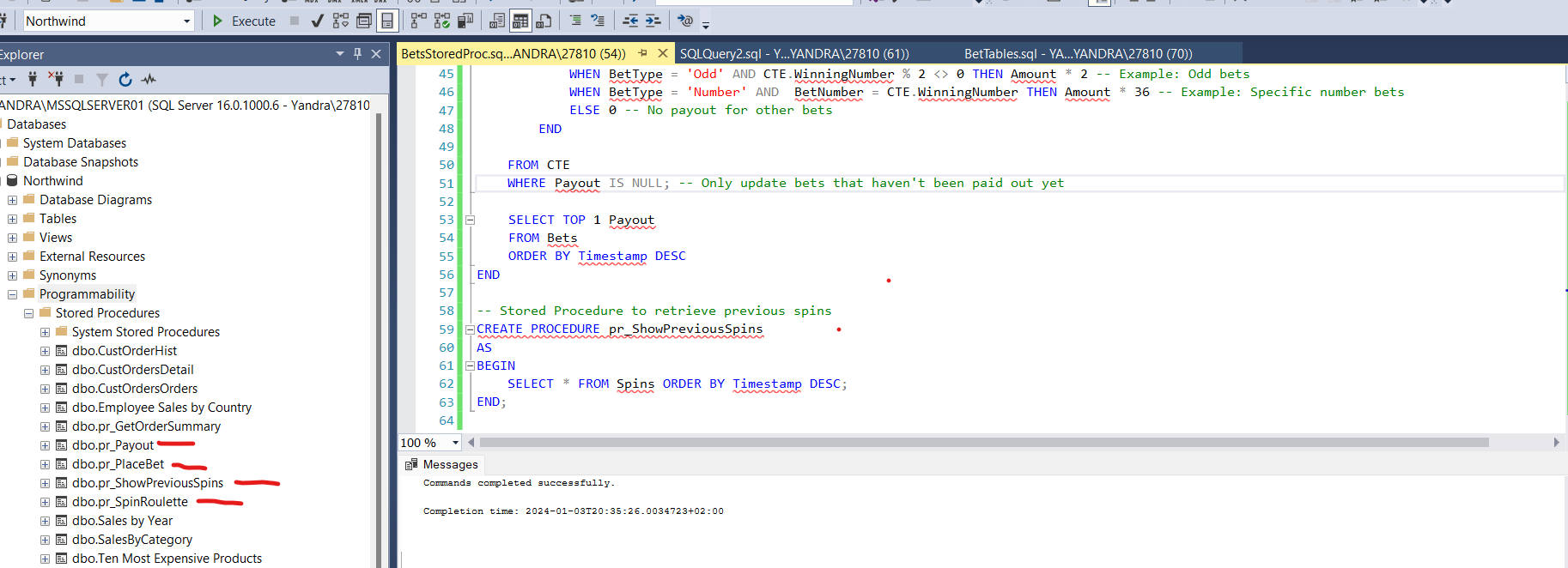
CREATE PROCEDURE pr\_ShowPreviousSpins

AS

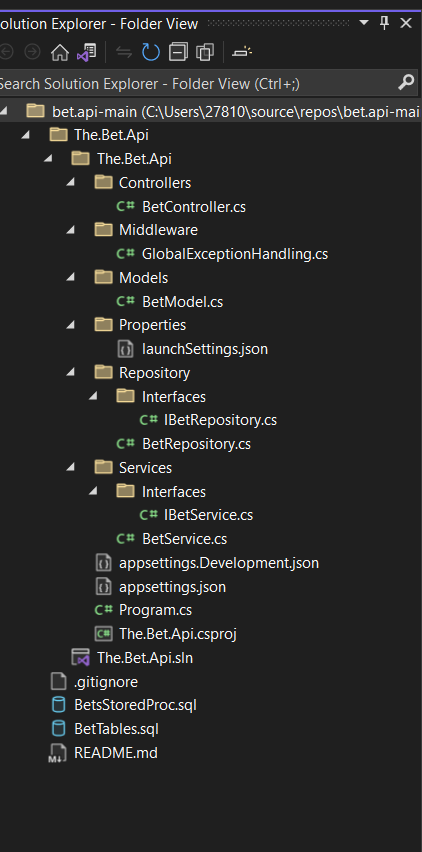
BEGIN

SELECT \* FROM Spins ORDER BY Timestamp DESC;

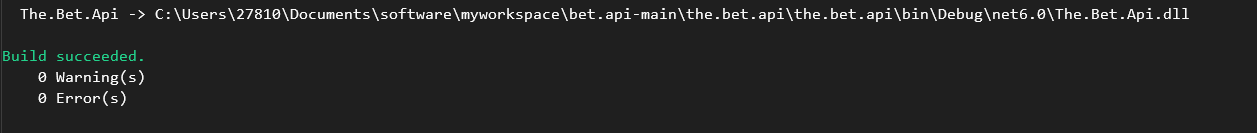
END;



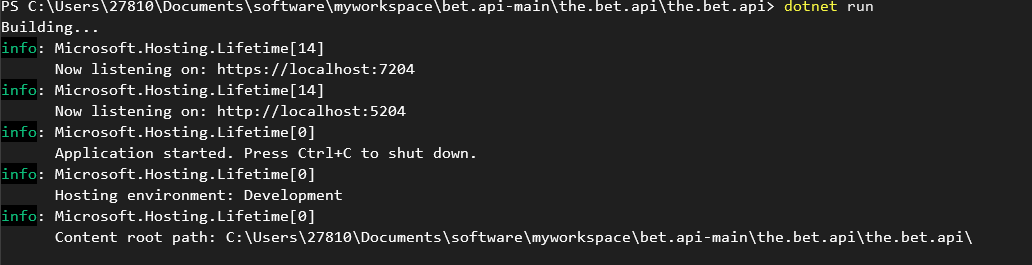
## VS Code MVC Bet.Web API Snapshots:



Build has no errors



Run .core app



Using your local host, run your uri and use this context for api resources

https://localhost:7204/

https://localhost:7204/api/

https://localhost:7204/api/GetAllPreviousBets

https://localhost:7204/api/PlaceBet

https://localhost:7204/api/Spin

<https://localhost:7204/api/GetPayout>



Thanks you. 😊