**SSRS Drillthrough & Subreport Tutorial – AdventureWorks2014 (VS 2022 Report Server Project)**

*Step-by-step hands-on guide with copy‑paste SQL and design notes*

# Business Scenario

Your retail leadership team needs a paginated reporting flow to explore sales in stages: a high-level summary by territory, a territory-level order detail with embedded top products, and an invoice-style view for a specific order.

1. Reports we will build:

* Executive Sales by Territory (main report): totals and unique customers by Country/Region → Territory. Optional Start/End Date filters.
* Territory Details (drillthrough): list orders for a selected territory and date range; embed a Top 5 Products subreport.
* Invoice (drillthrough): printable order header and line items for a selected SalesOrderID.

Why drillthrough? Jump to a focused report with its own layout/parameters. Why subreport? Embed contextual data (e.g., Top 5 Products) inside a parent report section.

# Prerequisites

• Visual Studio 2022 with SQL Server Reporting Services extensions installed.

• AdventureWorks2014 restored on a reachable SQL Server instance.

• A Report Server you can deploy to (URL like http://<server>/ReportServer).

• A new Report Server Project in VS 2022 (File → New → Project → Report Server Project).

# 1) Create a Shared Data Source

Solution Explorer → Shared Data Sources → Add New Data Source

Name: dsAdventureWorks2014

Type: Microsoft SQL Server

Connection string:

Data Source=YOUR\_SQL\_SERVER;Initial Catalog=AdventureWorks2014

Set appropriate credentials, Test Connection, OK.

# 2) Main Report: SalesByTerritory.rdl

## 2.1 Add Parameters

• StartDate (Date/Time) – Allow null value.

• EndDate (Date/Time) – Allow null value.

Leaving them null shows all data.

## 2.2 Dataset: dsSalesByTerritory

SELECT  
 t.CountryRegionCode,  
 t.Name AS Territory,  
 t.TerritoryID,  
 SUM(soh.SubTotal) AS SalesAmount,  
 COUNT(DISTINCT soh.SalesOrderID) AS Orders,  
 COUNT(DISTINCT soh.CustomerID) AS Customers  
FROM Sales.SalesOrderHeader AS soh  
JOIN Sales.SalesTerritory AS t ON soh.TerritoryID = t.TerritoryID  
WHERE  
 (@StartDate IS NULL OR soh.OrderDate >= @StartDate) AND  
 (@EndDate IS NULL OR soh.OrderDate < DATEADD(DAY, 1, @EndDate))  
GROUP BY t.CountryRegionCode, t.Name, t.TerritoryID  
ORDER BY SalesAmount DESC;

Note: (@Param IS NULL OR …) returns all rows if the parameter is blank. EndDate uses < DATEADD(DAY,1,EndDate) to be inclusive.

## 2.3 Design the Tablix

• Insert a Table with columns: CountryRegionCode, Territory, SalesAmount, Orders, Customers.

• Format SalesAmount as currency.

• Add a row group by CountryRegionCode (optional child group by Territory).

• Add group totals if desired.

## 2.4 Add Drillthrough on Territory

• Select the Territory textbox → Textbox Properties → Action → Go to report (TerritoryDetail.rdl).

• Pass parameters:

– TerritoryID = Fields!TerritoryID.Value

– StartDate = Parameters!StartDate.Value

– EndDate = Parameters!EndDate.Value

# 3) Drillthrough Report: TerritoryDetail.rdl

## 3.1 Parameters

• TerritoryID (Integer) – Required

• StartDate (Date/Time) – Allow null

• EndDate (Date/Time) – Allow null

## 3.2 Dataset: dsTerritoryOrders

SELECT  
 soh.SalesOrderID,  
 soh.OrderDate,  
 soh.SubTotal,  
 soh.TaxAmt,  
 soh.Freight,  
 soh.TotalDue,  
 c.CustomerID,  
 pcu.FirstName + ' ' + pcu.LastName AS CustomerName,  
 sp.BusinessEntityID AS SalesPersonID,  
 psp.FirstName + ' ' + psp.LastName AS SalesPersonName  
FROM Sales.SalesOrderHeader AS soh  
JOIN Sales.Customer AS c ON soh.CustomerID = c.CustomerID  
LEFT JOIN Person.Person AS pcu ON c.PersonID = pcu.BusinessEntityID  
LEFT JOIN Sales.SalesPerson AS sp ON soh.SalesPersonID = sp.BusinessEntityID  
LEFT JOIN Person.Person AS psp ON sp.BusinessEntityID = psp.BusinessEntityID  
WHERE  
 soh.TerritoryID = @TerritoryID AND  
 (@StartDate IS NULL OR soh.OrderDate >= @StartDate) AND  
 (@EndDate IS NULL OR soh.OrderDate < DATEADD(DAY,1,@EndDate))  
ORDER BY soh.OrderDate DESC;

## 3.3 Layout

• Title shows TerritoryID (or lookup name if you add a small helper dataset).

• Add a Table: OrderDate, SalesOrderID, CustomerName, SalesPersonName, SubTotal, TaxAmt, Freight, TotalDue.

• Add drillthrough on SalesOrderID to Invoice.rdl, passing SalesOrderID.

## 3.4 Embed Top 5 Products Subreport

• Insert a Subreport control in a header/rectangle above the orders table.

• Report: TopProductsSub.rdl.

• Pass parameters: TerritoryID, StartDate, EndDate from parent parameters.

# 4) Subreport: TopProductsSub.rdl

## 4.1 Parameters

• TerritoryID (Integer) – Required

• StartDate (Date/Time) – Allow null

• EndDate (Date/Time) – Allow null

## 4.2 Dataset: dsTopProducts

SELECT TOP 5  
 pc.Name AS Category,  
 p.Name AS ProductName,  
 SUM(sod.LineTotal) AS Revenue,  
 SUM(sod.OrderQty) AS Units  
FROM Sales.SalesOrderHeader AS soh  
JOIN Sales.SalesOrderDetail AS sod ON soh.SalesOrderID = sod.SalesOrderID  
JOIN Production.Product AS p ON sod.ProductID = p.ProductID  
LEFT JOIN Production.ProductSubcategory AS psc ON p.ProductSubcategoryID = psc.ProductSubcategoryID  
LEFT JOIN Production.ProductCategory AS pc ON psc.ProductCategoryID = pc.ProductCategoryID  
WHERE  
 soh.TerritoryID = @TerritoryID AND  
 (@StartDate IS NULL OR soh.OrderDate >= @StartDate) AND  
 (@EndDate IS NULL OR soh.OrderDate < DATEADD(DAY,1,@EndDate))  
GROUP BY pc.Name, p.Name  
ORDER BY Revenue DESC;

## 4.3 Layout

• Compact tablix with Category, ProductName, Units, Revenue (currency).

# 5) Drillthrough Target: Invoice.rdl

## 5.1 Parameter

• SalesOrderID (Integer) – Required

## 5.2 Datasets

Header (dsInvoiceHeader):

SELECT  
 soh.SalesOrderID,  
 soh.OrderDate,  
 soh.DueDate,  
 soh.ShipDate,  
 soh.SubTotal,  
 soh.TaxAmt,  
 soh.Freight,  
 soh.TotalDue,  
 c.CustomerID,  
 pcu.FirstName + ' ' + pcu.LastName AS CustomerName,  
 a.AddressLine1, a.City, a.PostalCode,  
 sp.BusinessEntityID AS SalesPersonID,  
 psp.FirstName + ' ' + psp.LastName AS SalesPersonName,  
 t.Name AS Territory  
FROM Sales.SalesOrderHeader AS soh  
JOIN Sales.Customer AS c ON soh.CustomerID = c.CustomerID  
LEFT JOIN Person.Person AS pcu ON c.PersonID = pcu.BusinessEntityID  
LEFT JOIN Person.Person AS psp ON soh.SalesPersonID = psp.BusinessEntityID  
LEFT JOIN Sales.SalesTerritory AS t ON soh.TerritoryID = t.TerritoryID  
LEFT JOIN Person.Address AS a ON soh.ShipToAddressID = a.AddressID  
WHERE soh.SalesOrderID = @SalesOrderID;

Lines (dsInvoiceLines):

SELECT  
 sod.SalesOrderID,  
 sod.SalesOrderDetailID,  
 p.Name AS ProductName,  
 sod.OrderQty,  
 sod.UnitPrice,  
 sod.UnitPriceDiscount,  
 sod.LineTotal  
FROM Sales.SalesOrderDetail AS sod  
JOIN Production.Product AS p ON sod.ProductID = p.ProductID  
WHERE sod.SalesOrderID = @SalesOrderID  
ORDER BY sod.SalesOrderDetailID;

## 5.3 Layout

• Header block with customer, salesperson, dates, territory, and address.

• Lines table with ProductName, OrderQty, UnitPrice, Discount, LineTotal.

• Totals area showing SubTotal, Tax, Freight, and TotalDue from header dataset.

# 6) Wire Up the Drillthroughs

• In SalesByTerritory.rdl → Territory textbox → Action → Go to report = TerritoryDetail.rdl; pass TerritoryID, StartDate, EndDate.

• In TerritoryDetail.rdl → SalesOrderID textbox → Action → Go to report = Invoice.rdl; pass SalesOrderID.

# 7) "Blank Means All" – Reliable Pattern

• Use WHERE (@Param IS NULL OR Column >= @Param) style in all queries.

• Mark StartDate/EndDate parameters as Allow null so previewing with blanks returns all data.

• Use < DATEADD(DAY,1,@EndDate) to make EndDate inclusive regardless of time components.

# 8) Preview & Deploy

• Preview each report; test with and without Start/End dates.

• Project Properties → TargetServerURL = your Report Server URL (e.g., http://YOUR\_RS\_SERVER/ReportServer).

• Build and Deploy; verify the drillthrough path in the web portal.

# Tips & Variations

• Add conditional formatting (e.g., highlight high freight).

• Add a sparkline or mini chart via a monthly trend dataset.

• Consider shared datasets/parameters for reuse and consistency.

• Use page headers/footers with useful metadata (run date, user).