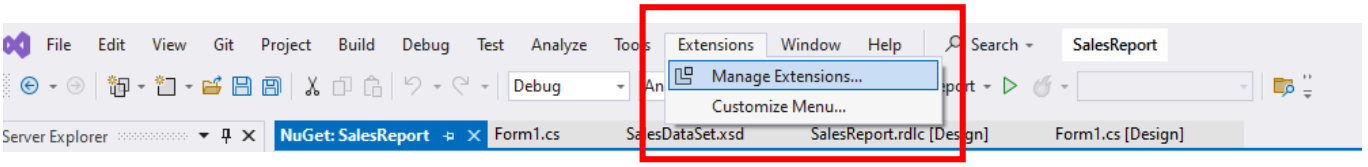
Practical Guide 10

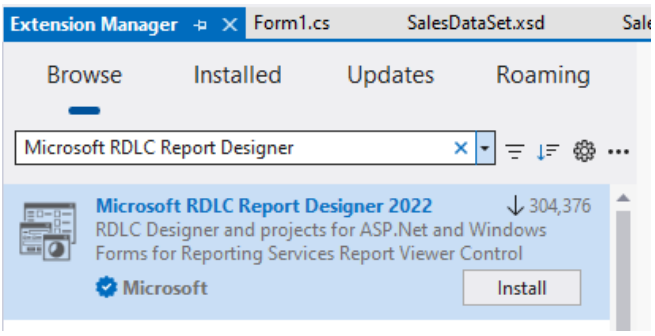
RDLC Reports with Chart

Step 1: Install RDLC Report Designer Extension

1. Open **Visual Studio 2022**
2. Go to **Extensions → Manage Extensions**



3. Search for `Microsoft RDLC Report Designer`



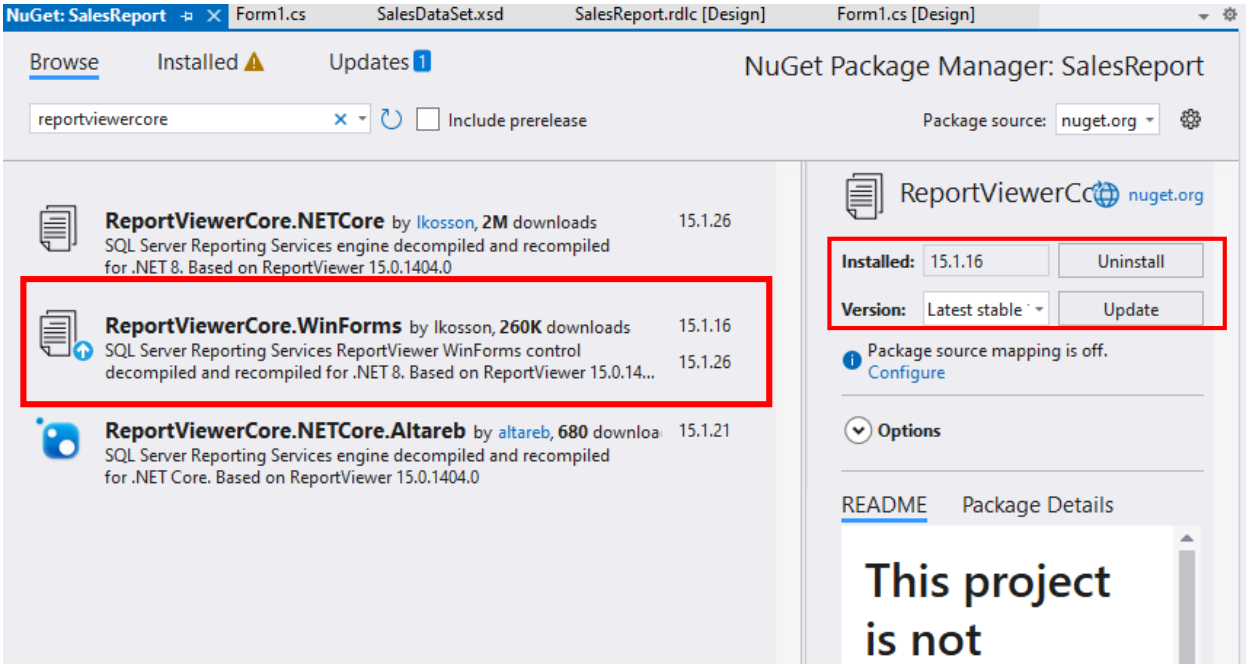
4. Click **Install**
5. Close **Visual Studio**
6. The **Visual Studio Installer** will automatically launch and install the extension (It downloads a file like `Microsoft.RdlcDesigner.vsix` in the background)
7. Once installation completes, **reopen Visual Studio**

Step 2: Create New Windows Forms Project

1. Go to **File → New → Project**
2. Select **Windows Forms App** (or **.NET Core/NET 7/8** based on your requirement)
3. Name the project: `SalesReportApp`
4. Click **Create**

Step 3: Install NuGet Packages

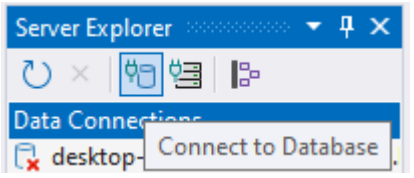
1. Right-click on the **project → Manage NuGet Packages**
2. Go to the **Browse** tab in **NuGet Package Manager** and install the following packages:
    - **reportviewercore.windows**
        - Use older version like **15.1.16** if you're using **.NET 7**
        - Use the **latest version** if you're using **.NET 8**



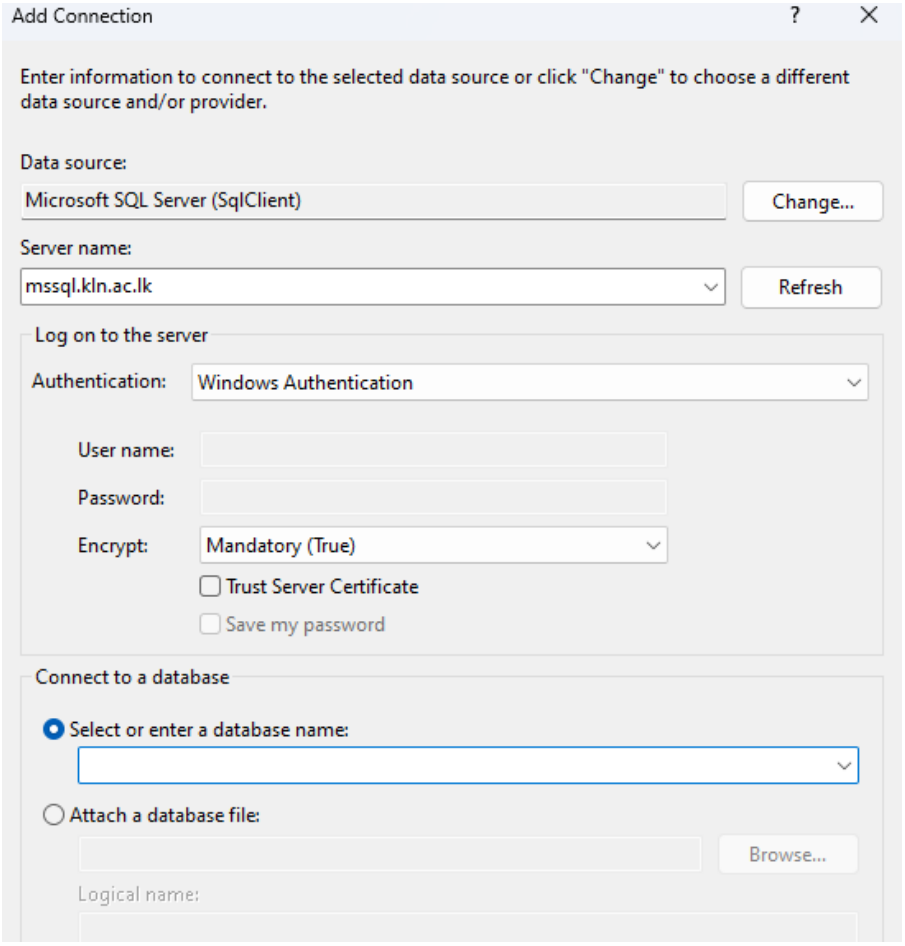
- **System.Data.SqlClient**
  - Install the **latest stable version**
  - This is required for accessing SQL databases from your application

Step 4: Connect Database via Server Explorer

1. Go to **View → Server Explorer**
2. Click on **Connect to Database.**



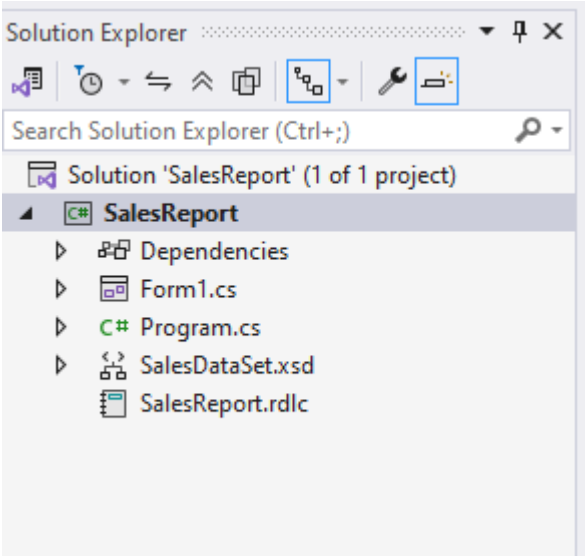
3. Choose **Microsoft SQL Server**
4. Enter server name and select your database



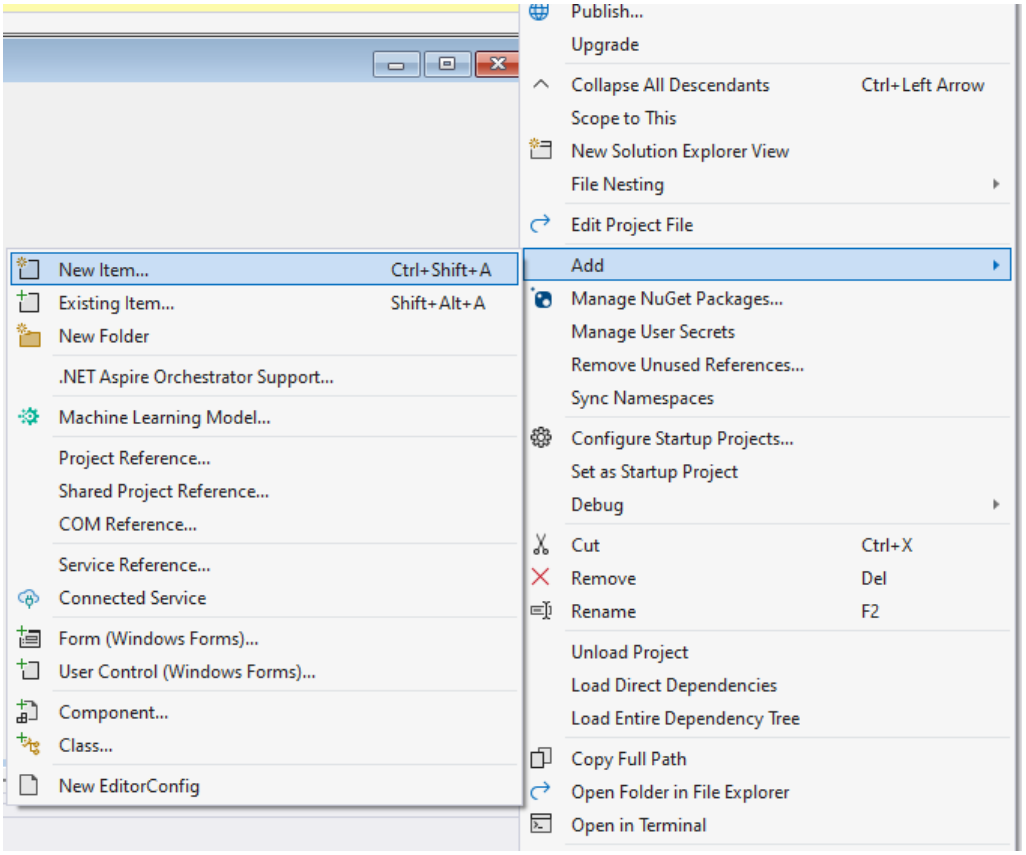
5. Click **OK**
6. Now your DB is connected to Visual Studio.

Step 5: Add New RDLC Report

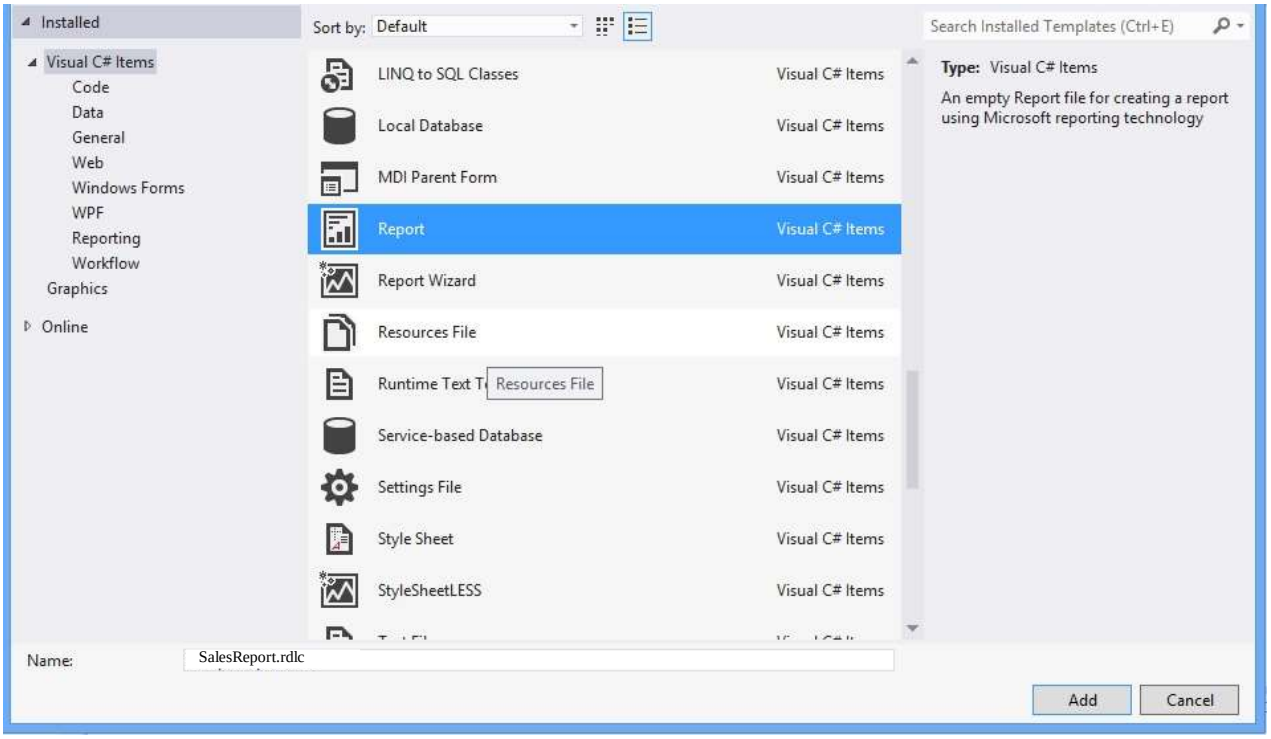
1. Right-click **project name**



2. **project name → Add → New Item**



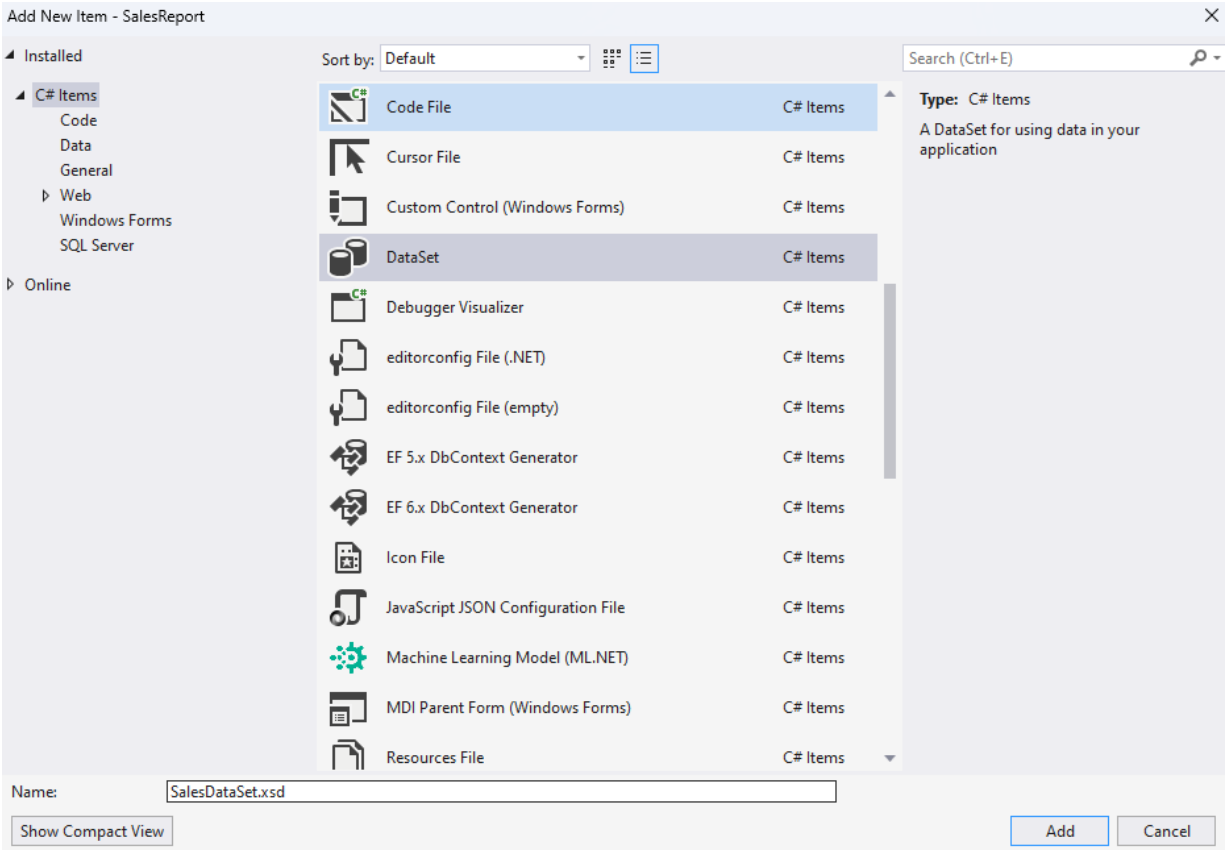
3. Choose **Report**



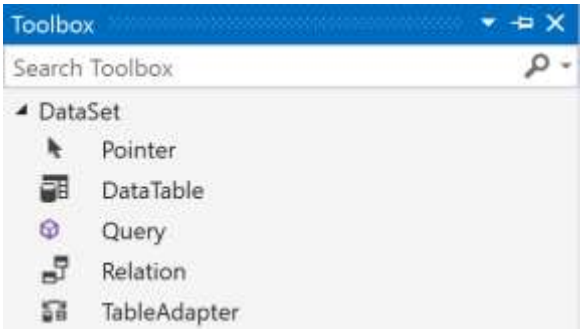
4. Name it: `SalesReport.rdlc` (make sure it ends with `.rdlc`)
5. Click **Add**
6. The RDLC report designer will open.

Step 6: Add Dataset to Project

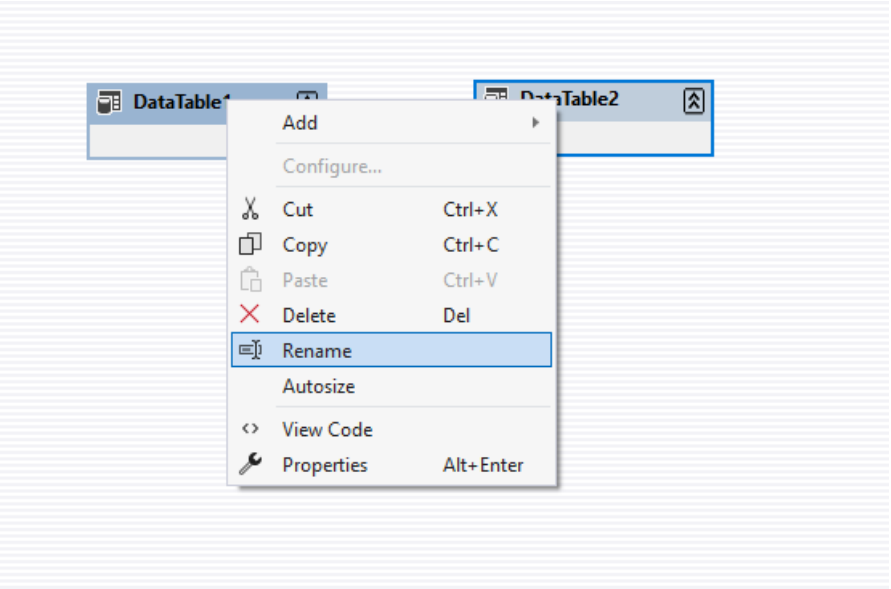
1. Right-click **project name → Add → New Item**
2. Select **DataSet** and name it: `SalesDataSet.xsd`



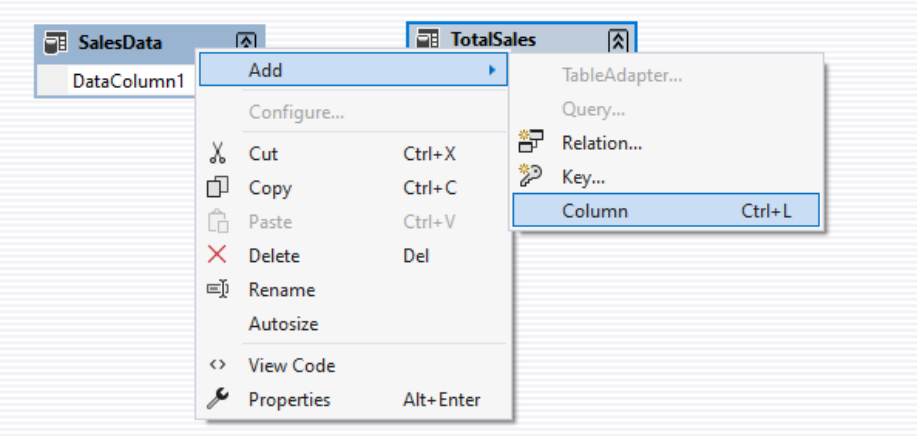
3. Click **Add**
4. In the dataset designer that opens, **drag and drop** one or more **DataTables.**



5. **Right-click** on the DataTable and rename them to `SalesData` and `TotalSales` as needed

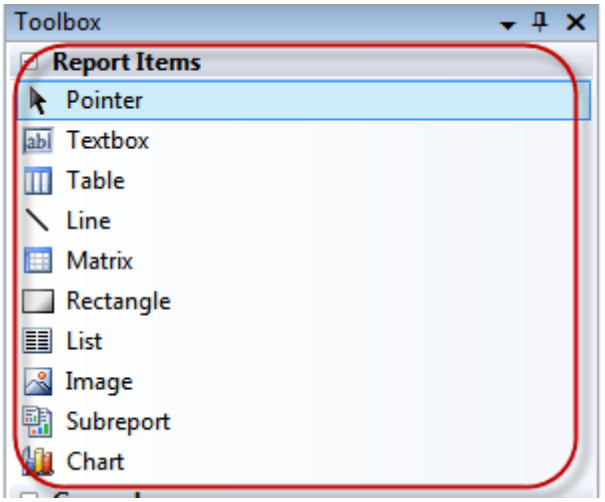


6. **Right-click** on the DataTable → Select **Add** → **Column**:
    1. For `SalesData`, add:
        1. `ProductName` *(string)*
        2. `Quantity` *(int or double)*
    2. For `TotalSales`, add:
        1. `ProductName` *(string)*
        2. `Total` *(int or double)*

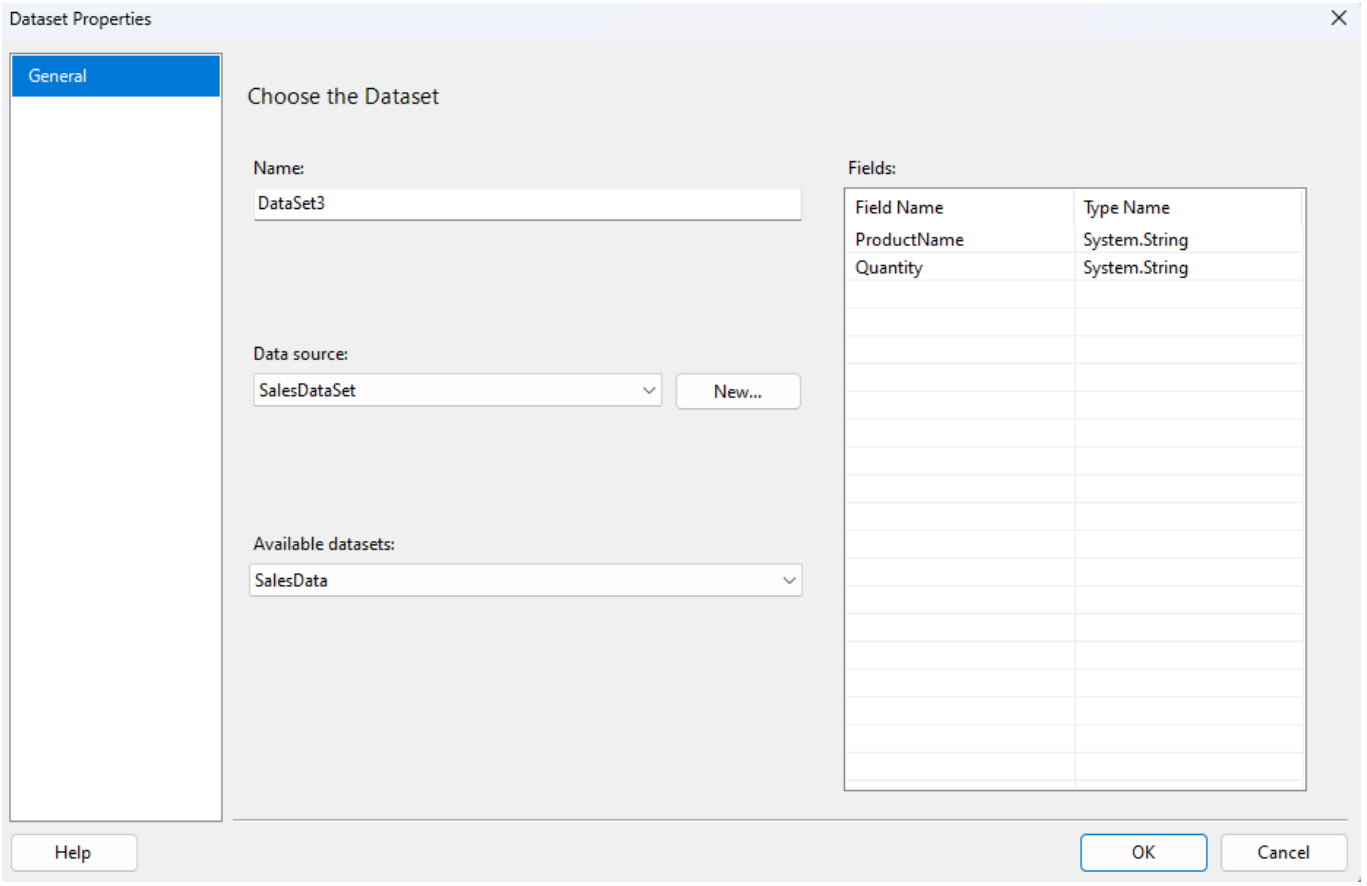


Step 7: Add Chart

1. Go back to the **SalesReport.rdlc** report designer
2. In the report Design tab, from **Toolbox** add a chart that you like.



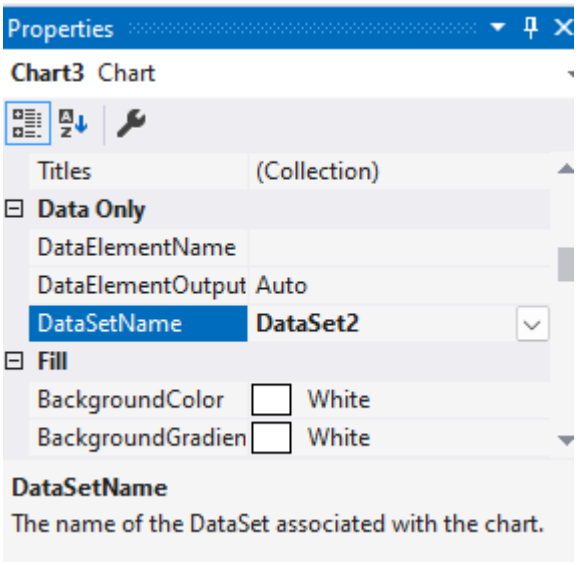
3. The **Dataset Properties** window will pop up
4. Set a **Dataset Name** (e.g., `SalesData`)
5. Under **Data source**, choose the previously created dataset (e.g., from `SalesDataSet.xsd`)
6. Under **Available datasets**, select the DataTable you want to use (e.g., `SalesData` or `TotalSales`)



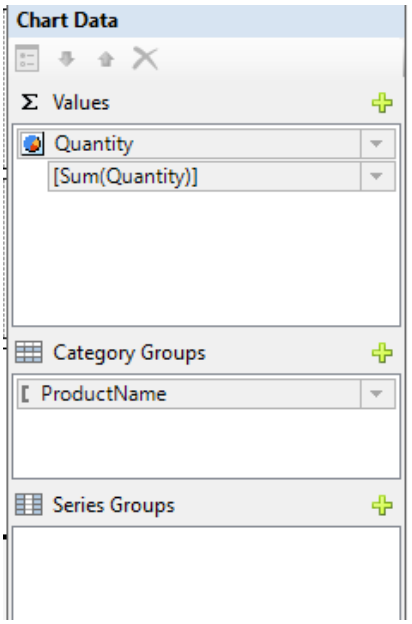
7. Click **OK**

Step 8: Assign dataset and configure values in the chart

1. Select the chart in the report designer.
2. Go to the **Properties** window.
3. Find the **DatasetName** property and select the dataset that corresponds to the data you want to show in the chart.



4. Click **OK.**
5. With the chart still selected, open the **Chart Data** panel.
6. In the **Values** section, click **Add (+)** and choose the field you want to display (e.g., `Quantity`).
7. You can apply an aggregate function such as **Sum**, **Count**, **Average**, etc.,
8. In the Category Groups section click Add (+) and select the column(s) from the dataset to use as categories (e.g., ProductName).



Step 9: Show Report on Windows Form and Populate Dataset

1. Make sure you have created the database and necessary tables using **SQL Server Management Studio (SSMS)**
2. In your **Form.cs** file, add the following namespaces at the top.

```
using System.Data.SqlClient;
using Microsoft.Reporting.WinForms;
```

3. Add the following code inside your form's class

```
string connStr = "Server=localhost;Database=SalesDB1;Trusted_Connection=True;";
public Form1()
{
    InitializeComponent();
}
private List<SalesData> GetSalesData()
{
    var list = new List<SalesData>();


    using (var conn = new SqlConnection(connStr))
    {
        conn.Open();
        string query = "SELECT ProductName, Quantity FROM Sales";

        using (var cmd = new SqlCommand(query, conn))
        using (var reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                list.Add(new SalesData
                {
                    ProductName = reader.GetString(0),
                    Quantity = reader.GetInt32(1)
                });
            }
        }
    }
    return list;
}
private List<TotalSalesData> GetTotalSalesData()
{
    var list = new List<TotalSalesData>();

    using (var conn = new SqlConnection(connStr))
    {
        conn.Open();
        string query = "SELECT ProductName, Price*Quantity FROM Sales";

        using (var cmd = new SqlCommand(query, conn))
        using (var reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                list.Add(new TotalSalesData
                {
                    ProductName = reader.GetString(0),
                    Total = reader.GetDecimal(1)
```

```
                        });
                    }
                }
            }
            return list;
        }

        private void Form1_Load(object sender, EventArgs e)
        {

            var reportViewer = new ReportViewer
            {
                Dock = DockStyle.Fill,
                ProcessingMode = ProcessingMode.Local
            };
            this.Controls.Add(reportViewer);

            reportViewer.LocalReport.ReportPath = "D:\\Documents\\Documents\\Visual
Programming\\Class Examples\\SalesReport\\SalesReport\\SalesReport.rdlc";
            reportViewer.LocalReport.DataSources.Clear();


            reportViewer.LocalReport.DataSources.Add(new ReportDataSource("DataSet2",
GetSalesData()));
            reportViewer.LocalReport.DataSources.Add(new ReportDataSource("DataSet1",
GetTotalSalesData()));

            reportViewer.RefreshReport();
        }
    }
    public class SalesData
    {
        public string ProductName { get; set; }
        public int Quantity { get; set; }
    }
    public class TotalSalesData
    {
        public string ProductName { get; set; }
        public decimal Total { get; set; }
    }
}
```

Note

Method: GetSalesData()

This method connects to the database and retrieves `ProductName` and `Quantity` from the `Sales` table.

The result is stored as a list of `SalesData` objects to bind to the report.

Method: GetTotalSalesData()

This method retrieves total sales (Price × Quantity) for each product from the same `Sales` table. It's stored as a list of `TotalSalesData` objects for use in a second chart or table in the report.

Form Load: Display Report

When the form opens:

- A `ReportViewer` control is added to the form.
- It loads the `.rdlc` report file from the given path.
- It clears previous data sources and adds two:
- `DataSet2` shows quantity data from `GetSalesData()`
- `DataSet1` shows total sales from `GetTotalSalesData()`
- Finally, it refreshes the report to show data.

Data Classes

```
public class SalesData
{
    public string ProductName { get; set; }
    public int Quantity { get; set; }
}

public class TotalSalesData
{
    public string ProductName { get; set; }
    public decimal Total { get; set; }
}
```

These define the structure of the data you'll use in your report:

- `SalesData`: for basic sales (name and quantity)
- `TotalSalesData`: for total value (name and total amount)