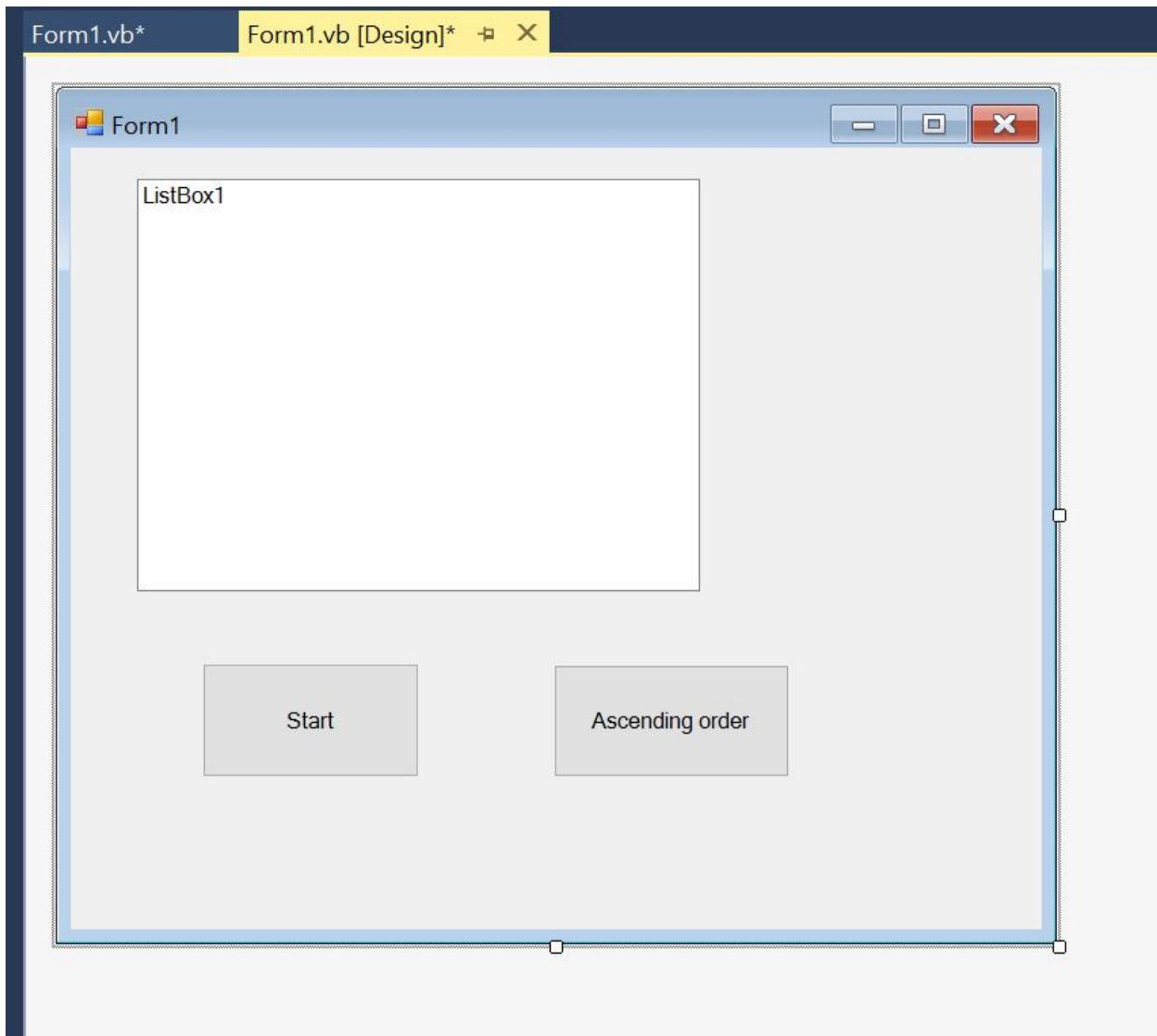


Rhituraj Bhalerao

Design view of the Form view: The list box will display the 100 integers in an array after inputting it. Ascending order button will sort the number in ascending order.



Code view:

```
Form1.vb*  X  Form1.vb [Design]*  
WindowsApp1  
VB Start  
1 reference  
1  Public Class Form1  
2   Dim numbers(99) As Integer  
3  
4   0 references  
5   Private Sub Start_Click(sender As Object, e As EventArgs) Handles Start.Click  
6     ListBox1.Items.Clear()  
7     For i As Integer = 0 To 99  
8       numbers(i) = CInt(InputBox("Enter integer", "Enter values"))  
9       ListBox1.Items.Add(numbers(i))  
10      Next  
11    End Sub  
12  
12  0 references  
13  Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
14    ListBox1.Items.Clear()  
15    Array.Sort(numbers)  
16    displayNumbers()  
17  End Sub  
18  
18  1 reference  
19  Private Sub displayNumbers()  
20    For Each item As Integer In numbers  
21      ListBox1.Items.Add(item)  
22    Next  
23  End Sub  
24  
25  
26  End Class  
27
```

Code view:

The screenshot shows the Visual Studio IDE in Code view, displaying the Default.aspx.vb file. The window title bar includes tabs for App_Web_ca5bh0vv.4.vb, Default3.aspx, Default2.aspx, Default.aspx.vb*, and Default.aspx*. The code editor shows the following VB.NET code:

```
2 references
2 1_Partial Class _Default
3     Inherits System.Web.UI.Page
4     Dim a As Double
5 
6     Private Sub method()
7         Response.Redirect("Default.aspx")
8     End Sub
9 
10    Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
11        'method()
12        If ListBox1.SelectedValue = 1 Then
13            ListBox1.Items.Clear()
14            a = TextBox1.Text * 0.88
15            'MsgBox(a)
16            ListBox1.Items.Add(a)
17        ElseIf ListBox2.SelectedValue = 2 Then
18            ListBox1.Items.Clear()
19            a = TextBox1.Text * 113.14
20            ListBox1.Items.Add(a)
21        End If
22    End Sub
23
24
25
26
27 End Class
28
```

The code implements a partial class named `_Default` that inherits from `System.Web.UI.Page`. It contains a private sub `method()` that redirects to `Default.aspx`. The `Button1_Click` event handler performs calculations based on the selected value in `ListBox1` or `ListBox2` and adds the result to `ListBox1`.

Design view of form:

The screenshot shows the design view of a web form titled "Default.aspx". The top navigation bar includes links for "App_Web_ca5bh0vv.4.vb", "Default3.aspx", "Default2.aspx", "Default.aspx.vb", and the active tab "Default.aspx". The form itself has a "body" section containing several controls:

- A text input field labeled "Enter the Conversion value".
- A dropdown menu labeled "Select a conversion type" with options "USD to Euro" and "USD to Yen".
- A text input field labeled "Answer" with the value "Unbound".
- A button labeled "Convert".

Web form view:

The screenshot shows the web form as it appears in a browser window. The address bar shows the URL "http://localhost:1039/Default.aspx". The page content is identical to the design view:

- A text input field labeled "Enter the Conversion value" containing the value "2".
- A dropdown menu labeled "Select a conversion type" with the option "USD to Euro" highlighted.
- A text input field labeled "Answer" containing the value "1.76".
- A button labeled "Convert".



Enter the Conversion value

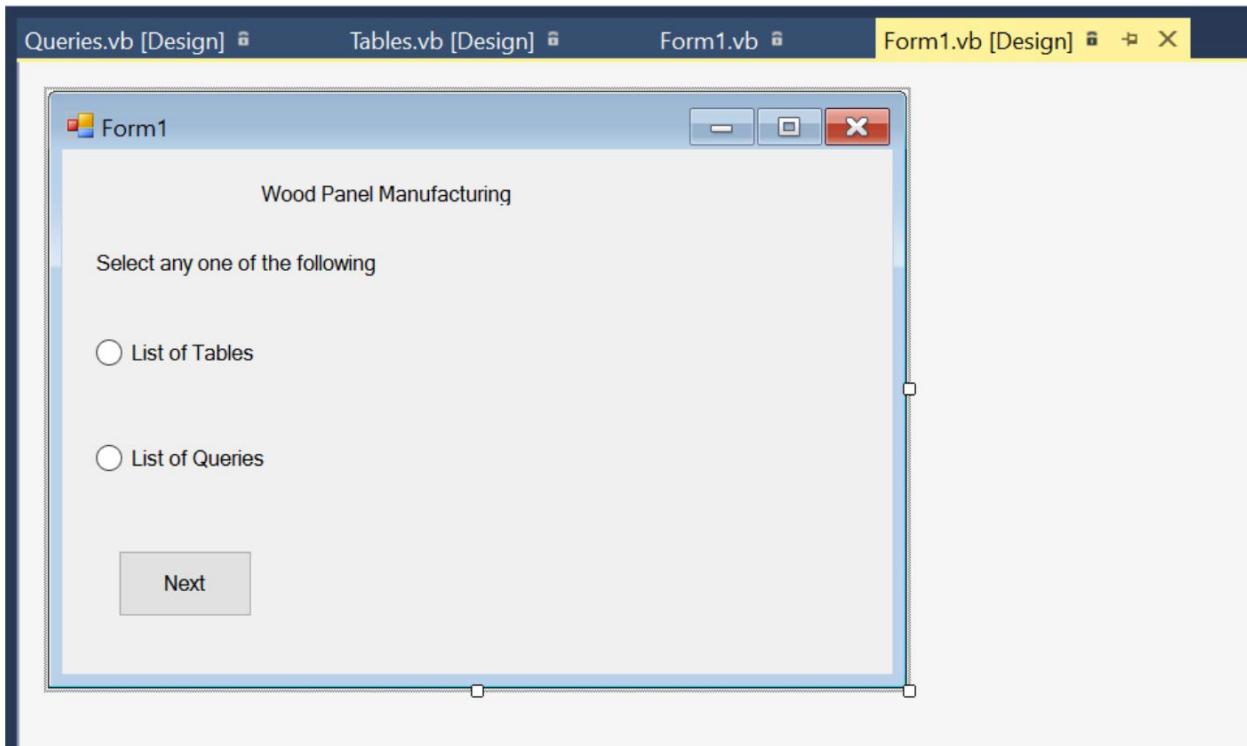
Select a conversion type

- [USD to Euro](#)
- [USD to Yen](#)

Answer

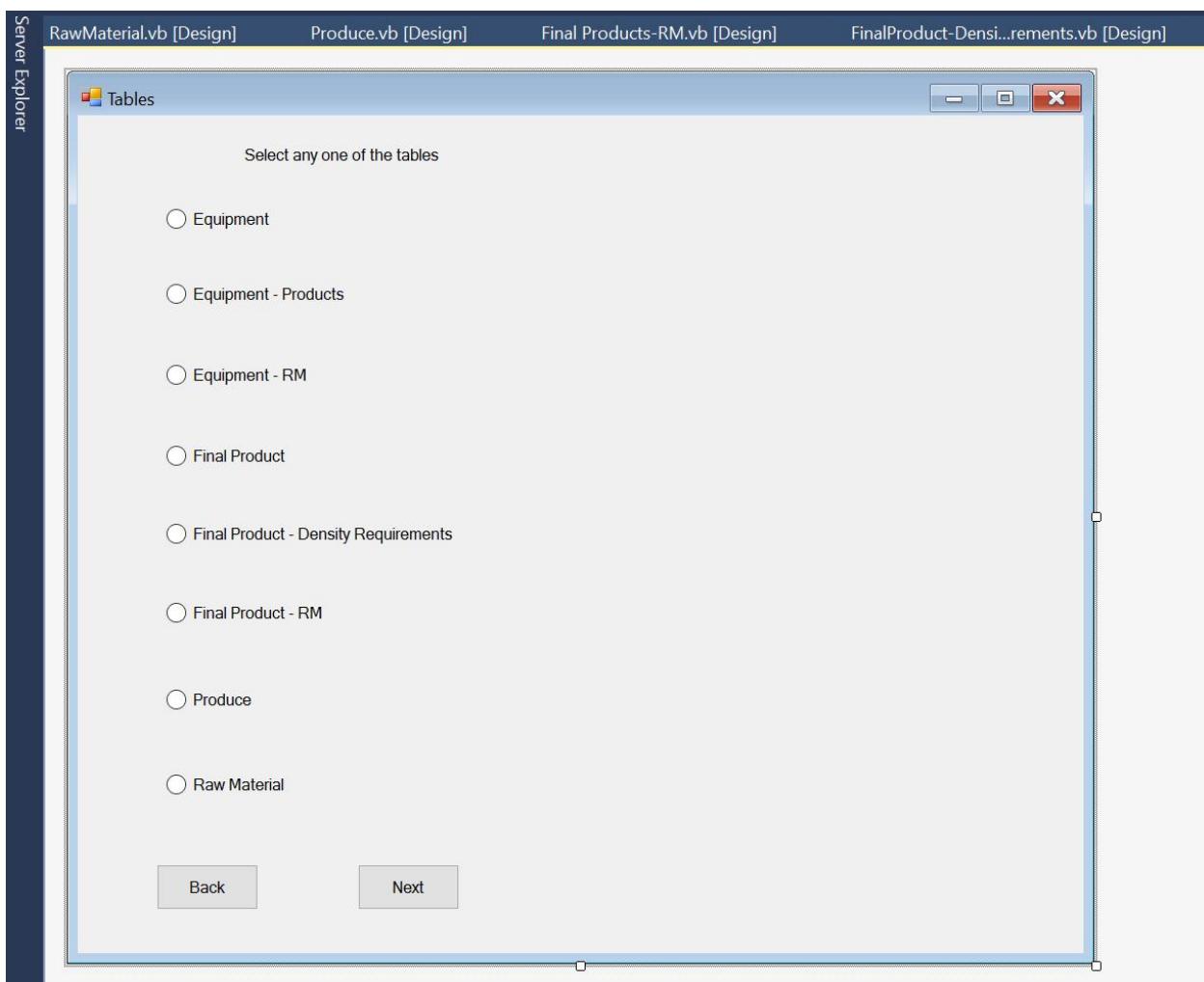
Wood Panel Manufacturing

Windows forms:



```
1 reference
1  Public Class Form1
2
3
4  Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
5    If RadioButton1.Checked Then
6      Me.Hide()
7      Tables.Show()
8    ElseIf RadioButton2.Checked Then
9      Me.Hide()
10     Queries.Show()
11    End If
12  End Sub
13
14 End Class
```

Tables form:

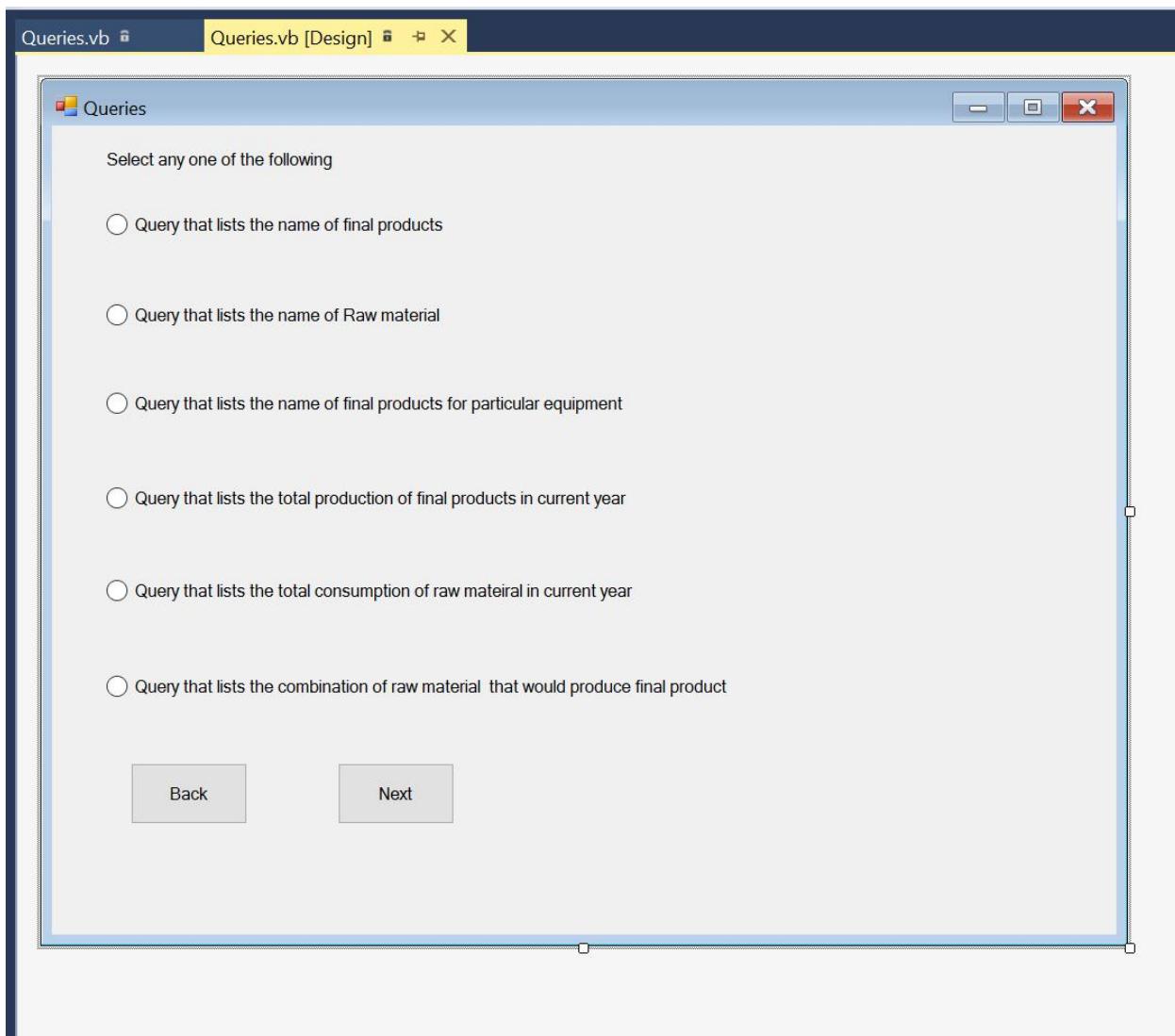


RawMaterial.vb [Design] Produce.vb [Design] Final Products-RM.vb [Design] FinalProduct-Densi...rements.vb [Design]

VB WindowsApp2

```
1  Public Class Tables
2      ' References:
3      Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
4          Me.Hide()
5          Form1.Show()
6      End Sub
7
8      ' References:
9      Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
10         If RadioButton1.Checked Then
11             Me.Hide()
12             Equipment.Show()
13         ElseIf RadioButton2.Checked Then
14             Me.Hide()
15             Equipments_Products.Show()
16         ElseIf RadioButton3.Checked Then
17             Me.Hide()
18             Equipment_RM.Show()
19         ElseIf RadioButton4.Checked Then
20             Me.Hide()
21             FinalProduct.Show()
22         ElseIf RadioButton5.Checked Then
23             Me.Hide()
24             FinalProduct_DensityRequirements.Show()
25         ElseIf RadioButton6.Checked Then
26             Me.Hide()
27             Final_Products_RM.Show()
28         ElseIf RadioButton7.Checked Then
29             Me.Hide()
30             Produce.Show()
31         ElseIf RadioButton8.Checked Then
32             Me.Hide()
33             RawMaterial.Show()
34
35     End If
End Sub
End Class
```

Queries Form:



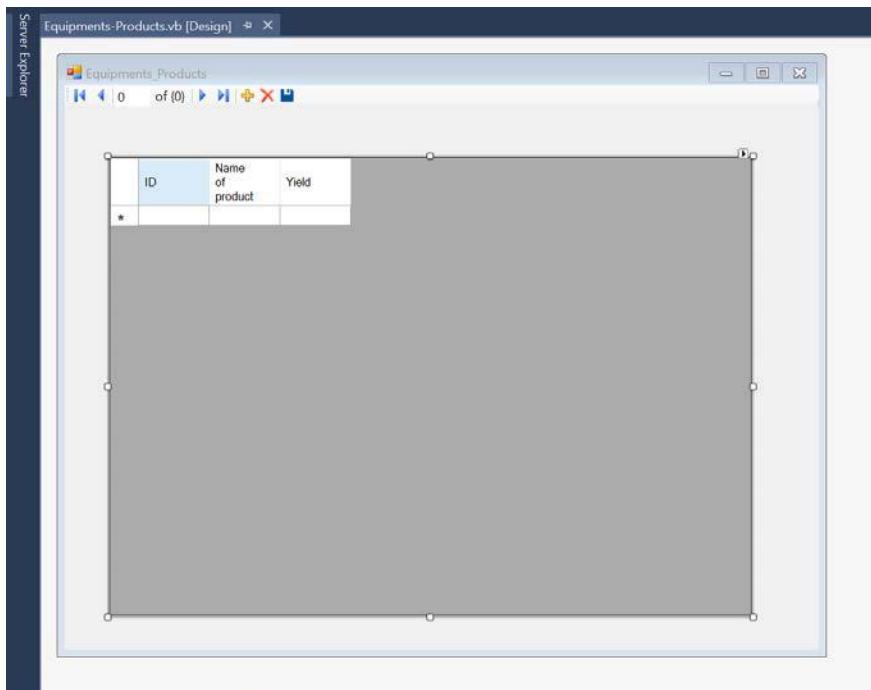
Queries.vb* WindowsApp2* Query1.vb [Design] Queries.vb [Design]*

VB WindowsApp2

```
1 reference
1 Public Class Queries
2     0 references
3         Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
4             Me.Hide()
5             Form1.Show()
6             End Sub
7
8             0 references
9                 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
10                    If RadioButton1.Checked Then
11                        Me.Hide()
12                        Query1.Show()
13                    ElseIf RadioButton2.Checked Then
14                        Me.Hide()
15                        Query2.Show()
16                    ElseIf RadioButton3.Checked Then
17                        Me.Hide()
18                        Query3.Show()
19                    ElseIf RadioButton4.Checked Then
20                        Me.Hide()
21                        Query4.Show()
22                    ElseIf RadioButton5.Checked Then
23                        Me.Hide()
24                        Query5.Show()
25                    ElseIf RadioButton6.Checked Then
26                        Me.Hide()
27                        Query6.Show()
28
29                End If
30            End Sub
31
32        End Class
```

Design and execution view of the Tables:

Equipment-Products:



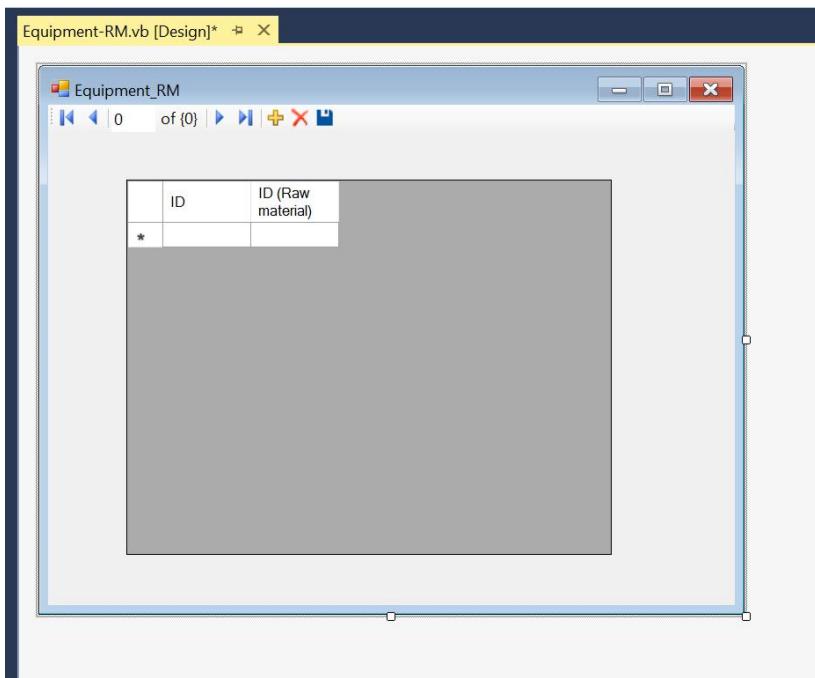
Execution view:

The screenshot shows the 'Equipments_Products' table in execution mode. The data is as follows:

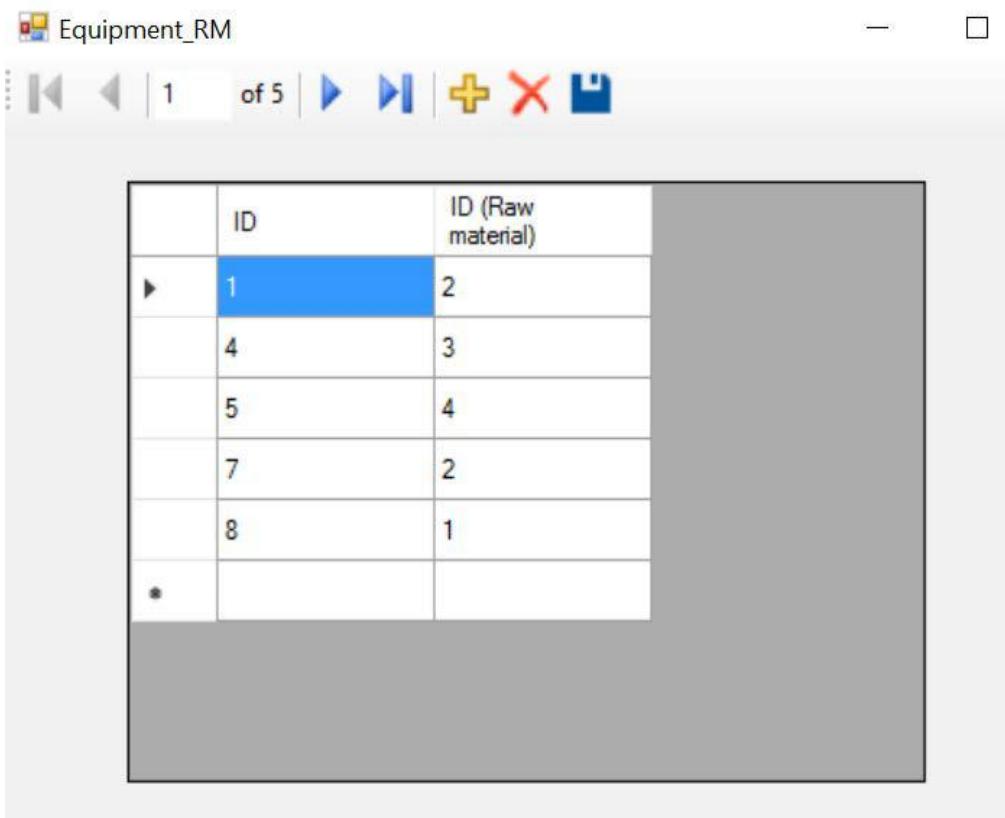
	ID	Name of product	Yield
▶	1	Ply	15
	4	Wood panel outsi...	20
	5	Sturdy	50
	7	Sturdy	25
	8	Door	20
*			

Equipment-RM

Design view:

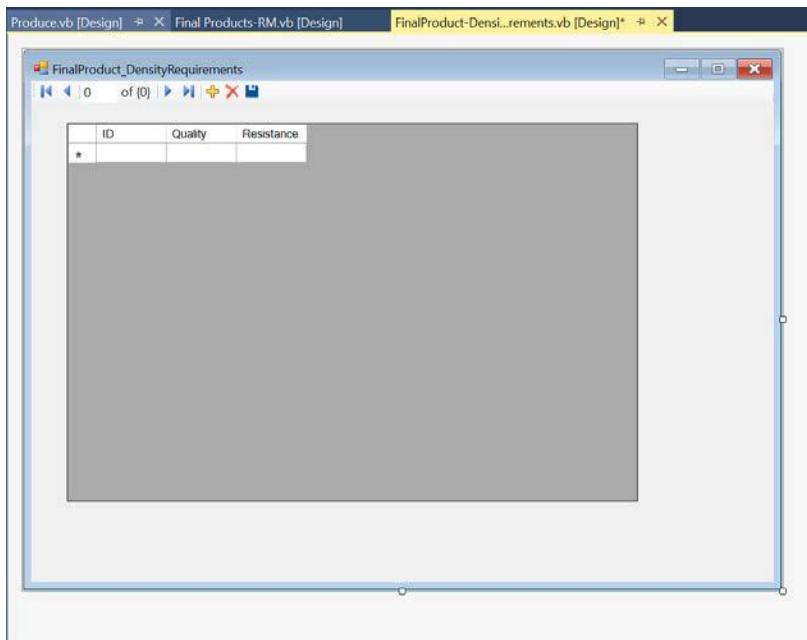


Execution view:



Final Product- Density Requirements

Design view:



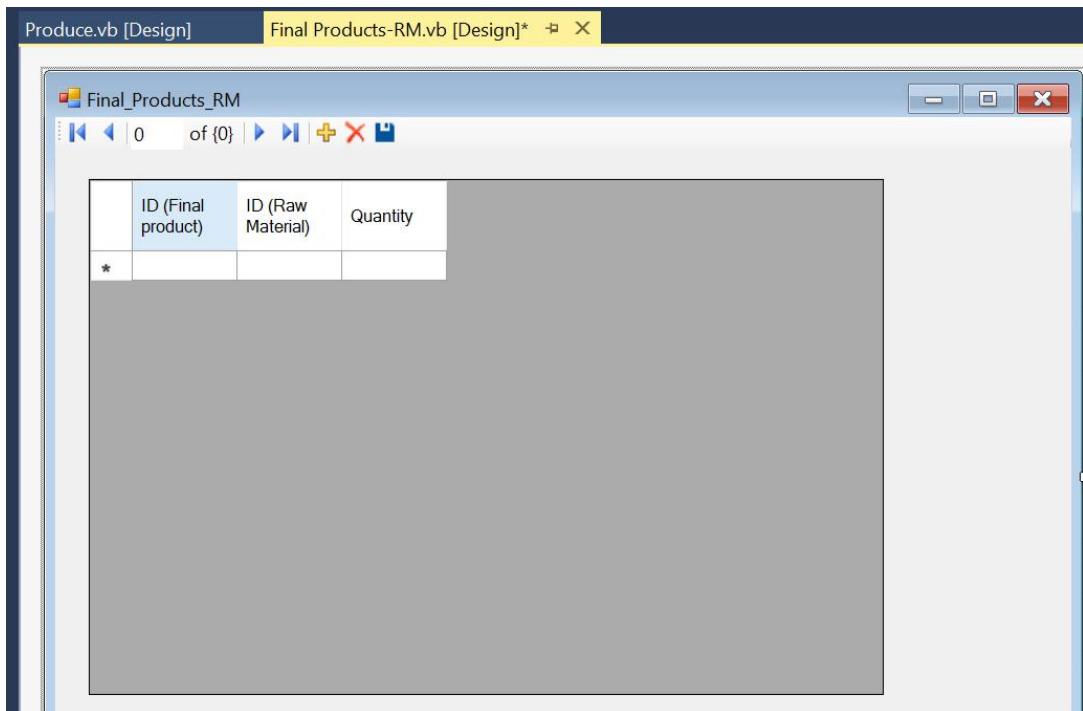
Execution view:

The screenshot shows the same application window in execution mode. The table now contains four rows of data:

	ID	Quality	Resistance
▶	1	Q1	R1
	2	Q2	R2
	3	Q3	R3
	4	Q4	R4
*			

Final Products –RM

Design view:

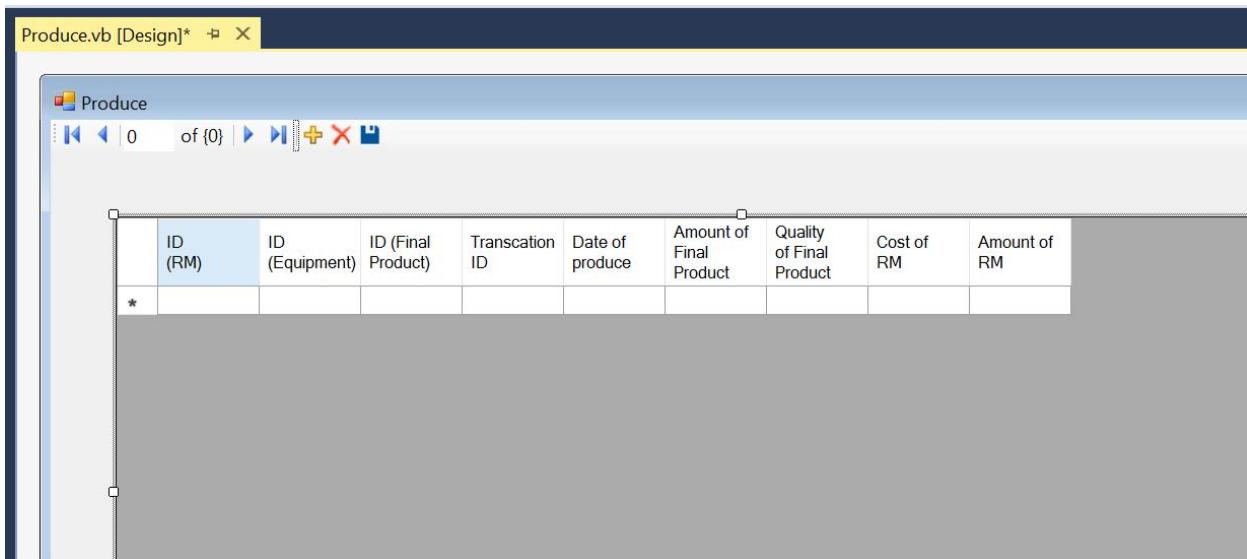


Execution view:

	ID (Final product)	ID (Raw Material)	Quantity
▶	1	2	60
	2	2	40
	2	3	50
	3	1	50
	4	4	20
*			

Produce

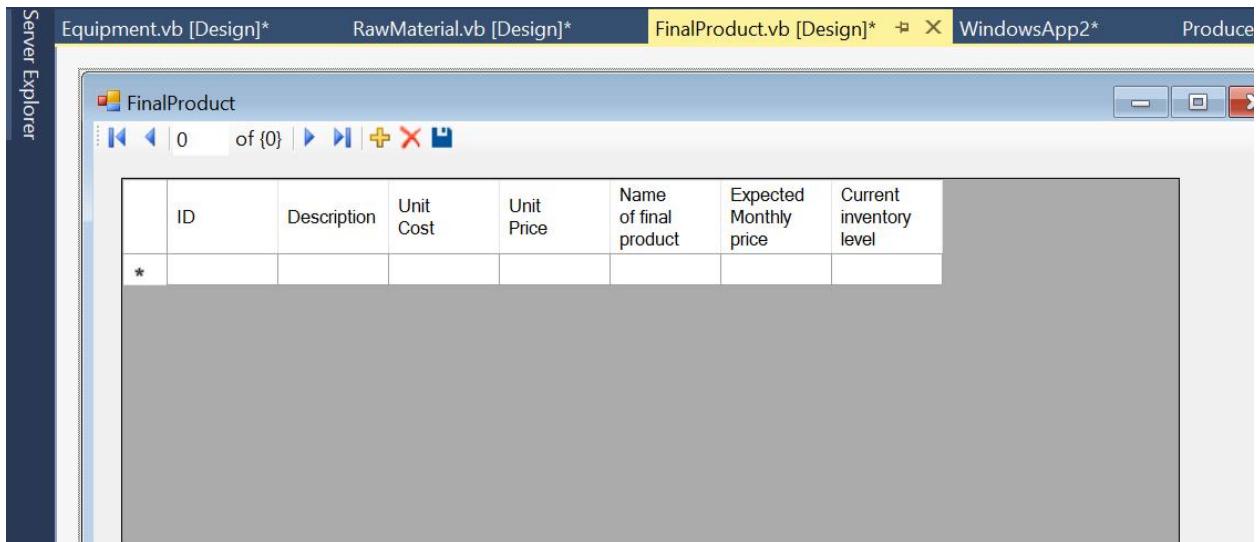
Design view:



Execution view:

Final Product

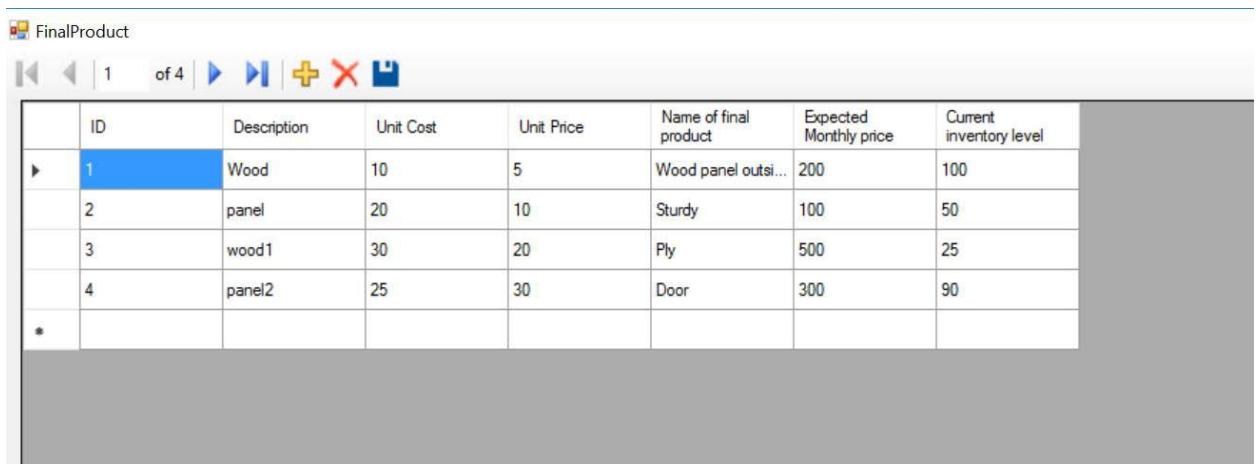
Design view:



The screenshot shows the Windows App Designer interface for the 'FinalProduct.vb [Design]' tab. At the top, there are tabs for 'Equipment.vb [Design]*', 'RawMaterial.vb [Design]*', 'FinalProduct.vb [Design]*' (which is selected and highlighted in yellow), 'WindowsApp2*', and 'Produc...'. On the left, a vertical 'Server Explorer' pane is visible. The main area displays a grid titled 'FinalProduct' with columns: ID, Description, Unit Cost, Unit Price, Name of final product, Expected Monthly price, and Current inventory level. A single row is present with an asterisk (*) in the ID column.

	ID	Description	Unit Cost	Unit Price	Name of final product	Expected Monthly price	Current inventory level
*							

Execution view:



The screenshot shows the running application window titled 'FinalProduct'. The grid displays four rows of data. The first row is highlighted with a blue background, indicating it is currently selected. The columns are labeled: ID, Description, Unit Cost, Unit Price, Name of final product, Expected Monthly price, and Current inventory level.

	ID	Description	Unit Cost	Unit Price	Name of final product	Expected Monthly price	Current inventory level
▶	1	Wood panel	10	5	Wood panel outside	200	100
	2	panel	20	10	Sturdy	100	50
	3	wood1	30	20	Ply	500	25
	4	panel2	25	30	Door	300	90
*							

Raw Material

Design view:

The screenshot shows the 'RawMaterial.vb [Design]' tab selected in the top navigation bar. On the left, there is a 'Server Explorer' pane. The main area contains a Windows Form titled 'RawMaterial'. The form has a title bar with the application name and a toolbar with standard icons. Below the toolbar is a DataGridView control. The grid has columns labeled 'ID', 'Name of Raw Material', 'Description', 'Unit Cost', 'Expected Monthly consumption', and 'Inventory level'. A single row is visible, starting with an asterisk (*) in the first column.

	ID	Name of Raw Material	Description	Unit Cost	Expected Monthly consumption	Inventory level
*						

Execution view:

The screenshot shows the 'RawMaterial' window running. The title bar indicates it is now active. The form contains a DataGridView with the same structure as the design view. The data is populated with five rows, each representing a raw material. Row 1 (ID 1) is highlighted in blue, indicating it is currently selected. The other rows contain data for 'flakes', 'strands', and 'shreds'.

	ID	Name of Raw Material	Description	Unit Cost	Expected Monthly consumption	Inventory level
▶	1	chips	10mm dimension	100	50	20
	2	flakes	20mm dimension	200	60	10
	3	strands	30mm dimension	300	50	40
	4	shreds	40mm dimension	150	10	50
*						

Equipment

Design view:

The screenshot shows a Windows application window titled "Equipment". The title bar also includes tabs for "RawMaterial.vb [Design]*", "FinalProduct.vb [Design]", and "WindowsApp2*". The main area displays a grid with columns labeled "ID", "Name of Equipment", "Location", and "Capacity". A single row is present with an asterisk (*) in the ID column and empty fields for Name, Location, and Capacity. Navigation buttons at the top left include back, forward, and search functions.

	ID	Name of Equipment	Location	Capacity
*				

Execution view:

The screenshot shows the same "Equipment" application window in execution mode. The grid now contains 10 rows of data, each representing a piece of equipment with a unique ID and specific details. The first row (ID 1) is highlighted in blue, indicating it is currently selected or active.

	ID	Name of Equipment	Location	Capacity
▶	1	wood cutter	L1	5
	2	CNC	L2	6
	3	Chain saw	L3	5
	4	File	L4	6
	5	Presser	L5	9
	6	Holder	L6	8
	7	Engraver	L7	6
	8	Cold Press	L8	5
	9	Veneer Peeling	L9	7
*	10	Sand paper	L10	9

Design and execution view of the Queries:

Query1:

Create a query that prompts for the name of a raw material and lists the name of the final products that can be produced using this raw material. The list should also contain the quantity of raw material needed to produce one unit of final product.

Query Builder and SQL view:

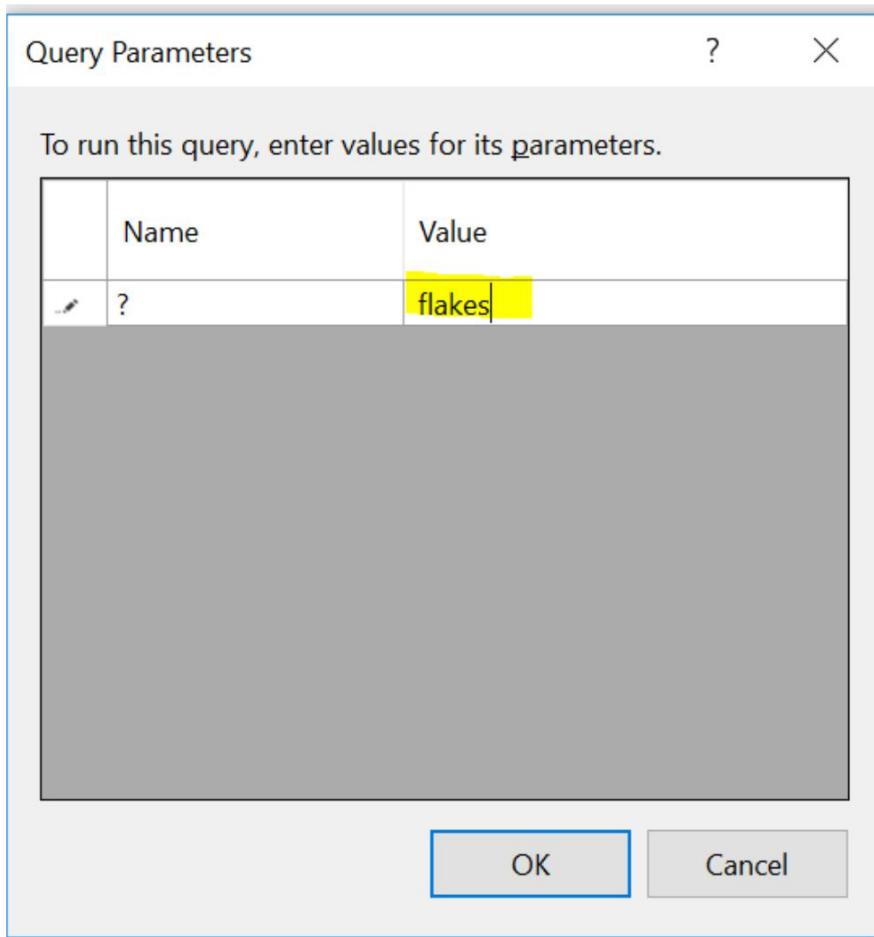
Query Builder

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of Raw Material]	Raw Material	Raw Material	<input checked="" type="checkbox"/>			= ?			
[Name of final product]	Final Product	Final Product	<input checked="" type="checkbox"/>						
Quantity	Final product - RM	Final product - RM	<input checked="" type="checkbox"/>						
			<input checked="" type="checkbox"/>						
			<input checked="" type="checkbox"/>						
			<input checked="" type="checkbox"/>						
			<input checked="" type="checkbox"/>						

```
SELECT [Raw Material].[Name of Raw Material], [Final Product].[Name of final product], [Final product - RM].Quantity
FROM ((([Final Product] INNER JOIN
        [Final product - RM] ON [Final Product].ID = [Final product - RM].[ID (Final product)]) INNER JOIN
        Produce ON [Final Product].ID = Produce.[ID (Final Product)]) INNER JOIN
        [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID)
WHERE ([Raw Material].[Name of Raw Material] <= ?)
```

Execute Query OK Cancel

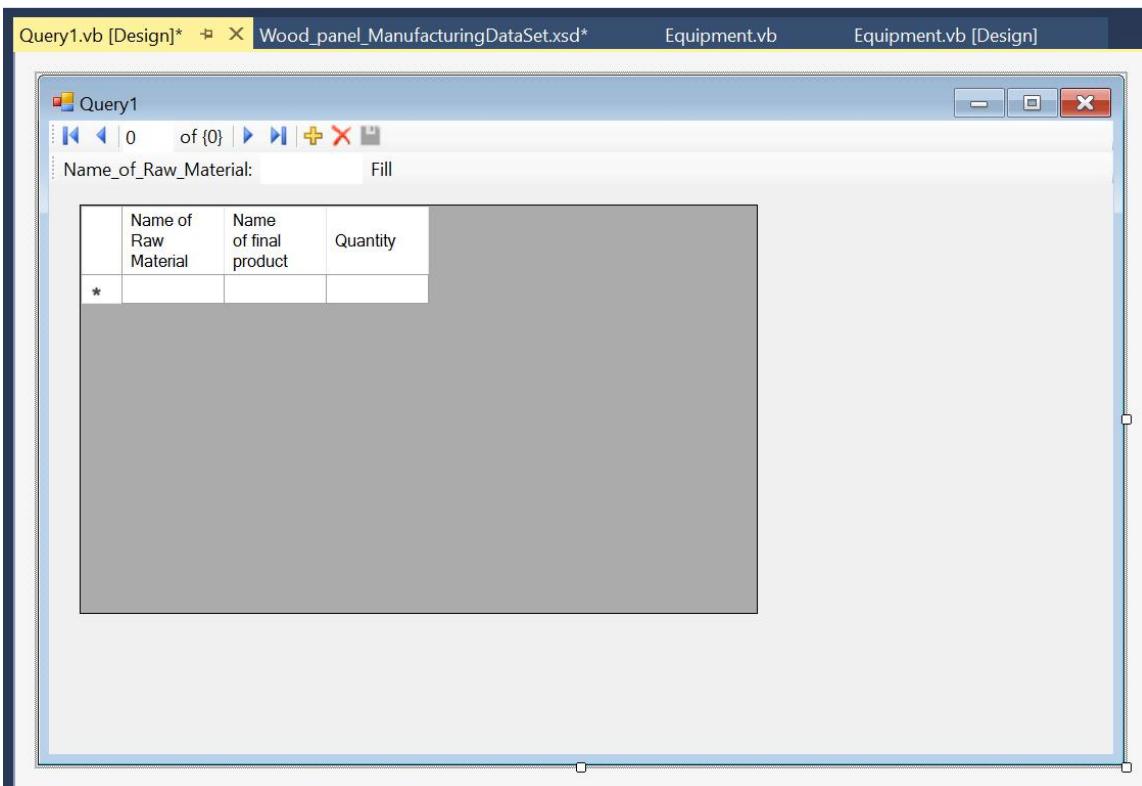
Execution view:



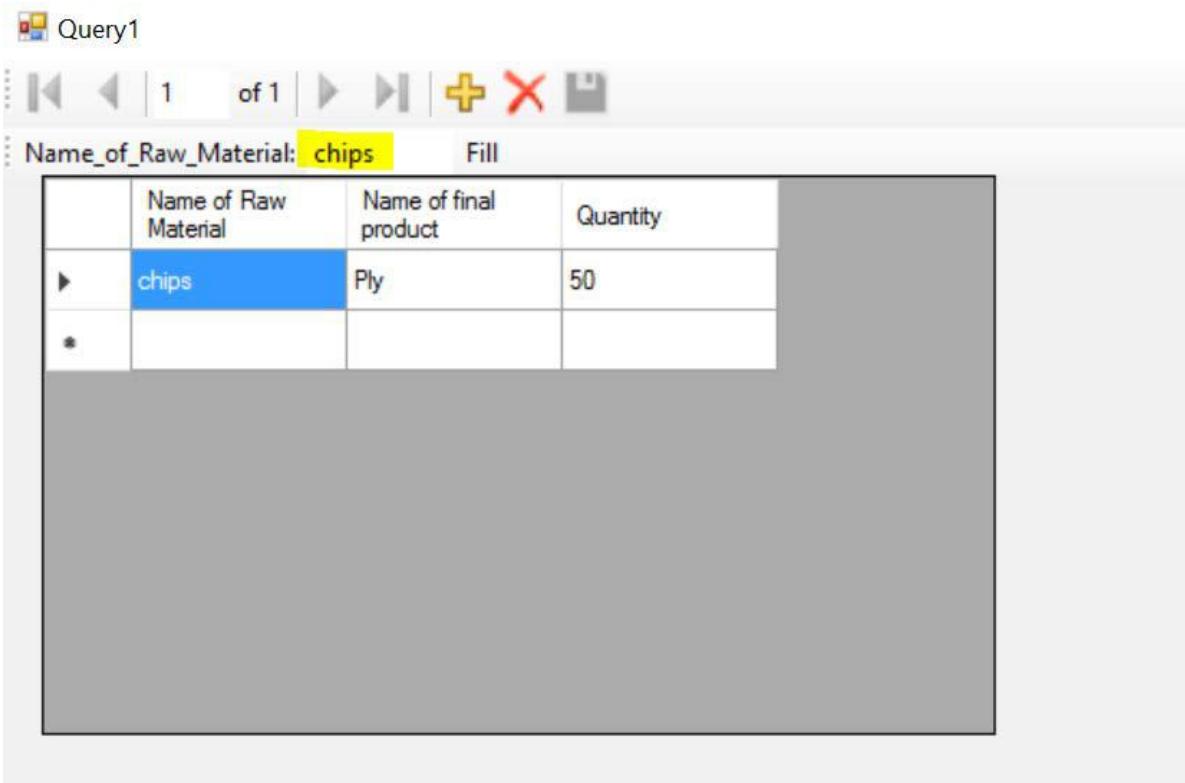
	Name of...	Name of...	Quantity
▶	chips	Ply	50
	flakes	Wood pa...	60
	flakes	Sturdy	40
	flakes	Sturdy	50

◀ ◀ | 1 of 4 ▶ ▶ ⌂ ⌂

Form design view of Query 1:



Execution view of form of Query 1:



Query 2:

Create a query that prompts for the name of a final product and presents a list of the raw materials that can be used. Present the quantity needed of each raw material to produce one unit of the final product.

Query Builder and SQL View:

Query Builder

The Query Builder interface shows the following setup:

- Raw Material:** Columns include * (All Columns), ID, Name of Raw Material (selected), Description, Unit Cost, Expected Monthly consumption, and Inventory level.
- Produce:** Columns include * (All Columns), ID (RM), ID (Equipment), ID (Final Product), Transaction ID, Date of produce, Amount of Final Product, Quality of Final Product, Cost of RM, and Amount of RM.
- Final Product - RM:** Columns include * (All Columns), ID, Description, Unit Cost, Unit Price, Name of final product (selected), Expected Monthly price, and Current inventory level.

SQL View:

Column	Alias	Table	Output	Sort type	Sort Order	Filter	Or...	Or...	Or...
[Name of final product]	Final Product					= ?			
[Name of Raw Material]	Raw Material								
Quantity	Final product - RM								

```
SELECT [Final Product].[Name of final product], [Raw Material].[Name of Raw Material], [Final product - RM].Quantity
FROM   ([Final Product] INNER JOIN
        [Final product - RM] ON [Final Product].ID = [Final product - RM].ID (Final Product)) INNER JOIN
        Produce ON [Final Product].ID = Produce.ID (Final Product)) INNER JOIN
        [Raw Material] ON Produce.ID (RM) = [Raw Material].ID
WHERE  [Final Product].[Name of final product] <= ?
```

Buttons at the bottom: Execute Query, OK, Cancel.

Execution view:

Query Parameters

?

X

To run this query, enter values for its parameters.

	Name	Value
?	?	Sturdy

OK

Cancel

	Name of...	Name of...	Quantity
►	Sturdy	flakes	40
	Sturdy	flakes	50
	Sturdy	strands	40
	Sturdy	strands	50

Form design view of Query 2:

Query2

0 of {0} | + X

Name_of_final_product: Fill

	Name of final product	Name of Raw Material	Quantity
*			

Execution view of form of Query 2:

Query2

1 of 6 | + X

Name_of_final_product: Sturdy Fill

	Name of final product	Name of Raw Material	Quantity
▶	Sturdy	flakes	40
	Sturdy	flakes	50
	Sturdy	strands	40
	Sturdy	strands	50

Query 3:

Create a query that prompts for the name of an equipment and presents a list of the final products that can be produced using this particular equipment.

Query Builder and SQL view:

Query Builder

Column	Alias	Table	Output Type	Sort Order	Filter	Or...	Or...	Or...
[Name of Equipment]		Equipment	<input checked="" type="checkbox"/>		= ?			
[Name of product]		Equipment - Products	<input checked="" type="checkbox"/>					

```
SELECT Equipment.[Name of Equipment], [Equipment - Products].[Name of product]
FROM (((Equipment INNER JOIN
[Equipment - Products] ON Equipment.ID = [Equipment - Products].ID) INNER JOIN
Produce ON Equipment.ID = Produce.[ID (Equipment)]) INNER JOIN
[Final Product] ON Produce.[ID (Final Product)] = [Final Product].ID)
WHERE (Equipment.[Name of Equipment] <= ?)
```

Execute Query OK Cancel

Execution view:

Query Parameters ? ×

To run this query, enter values for its parameters.

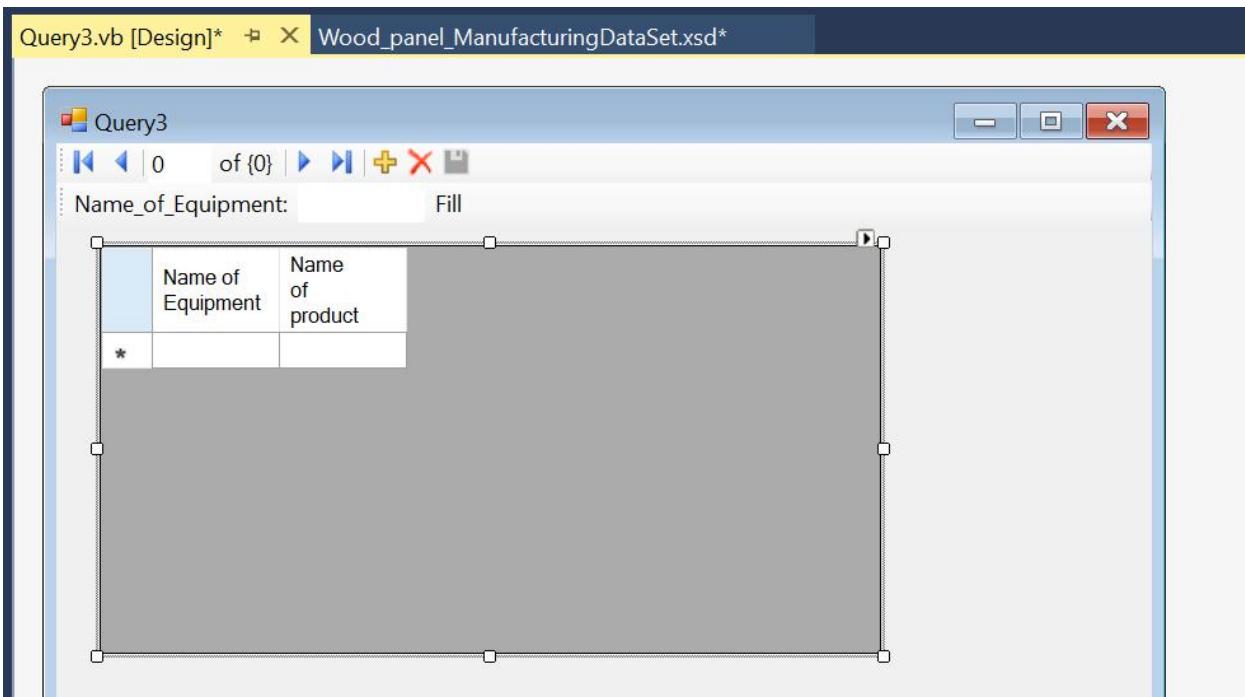
	Name	Value
	?	wood cutter

OK

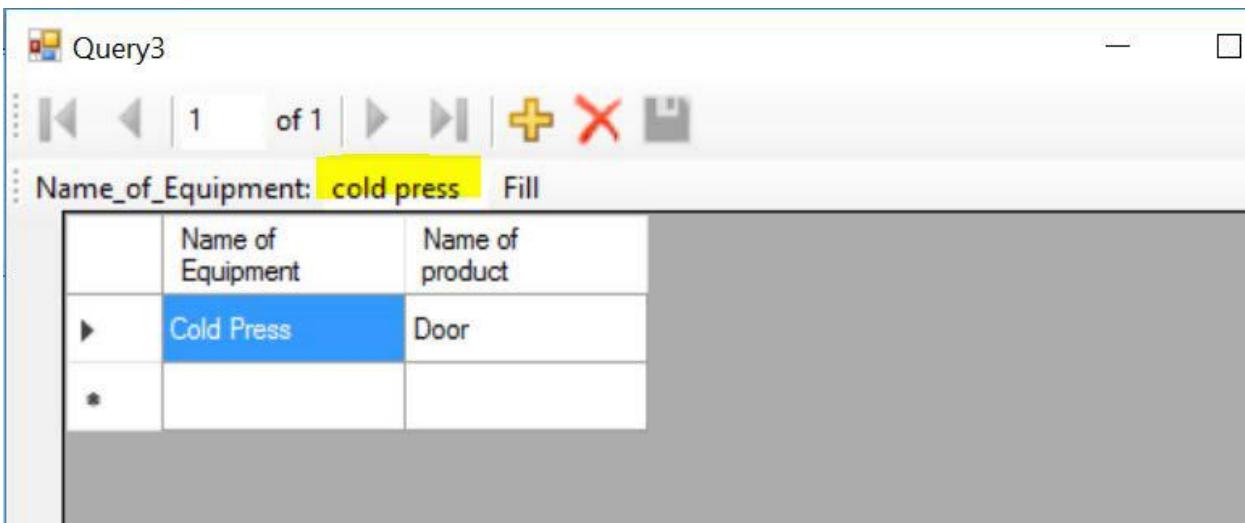
Cancel

	Name of Equipment	Name of product
	wood cutter	Ply

Form design view of Query 3:



Execution view of form of Query 3:

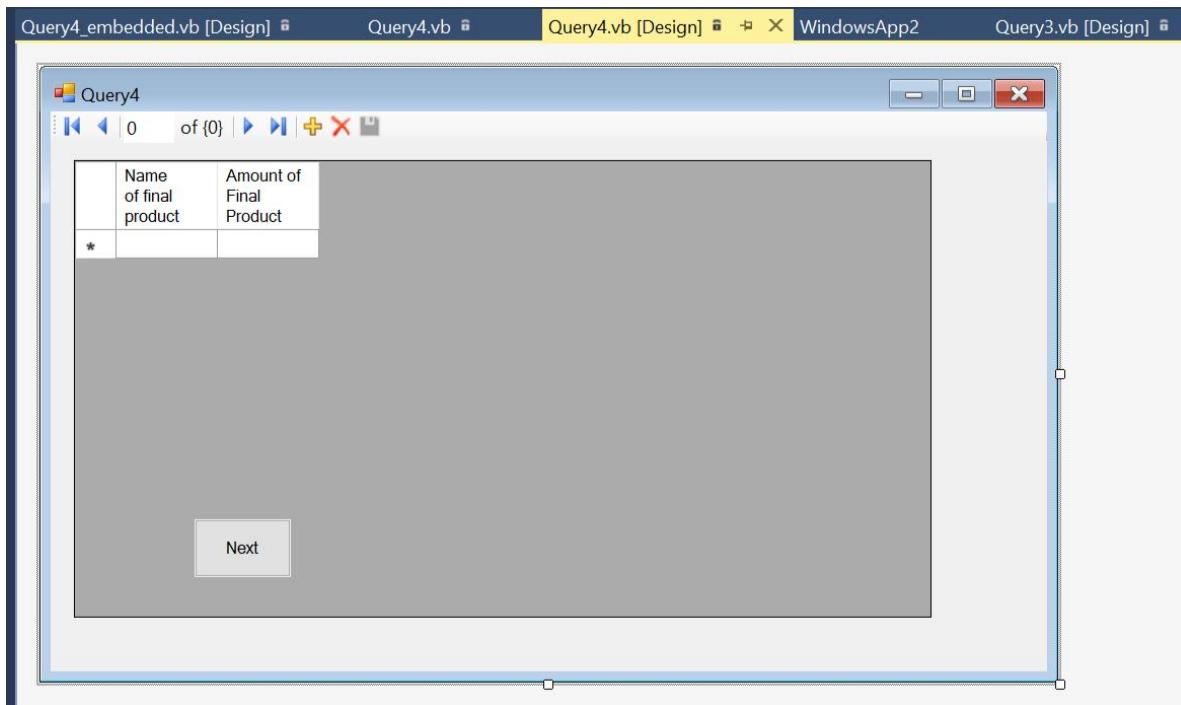


Query 4:

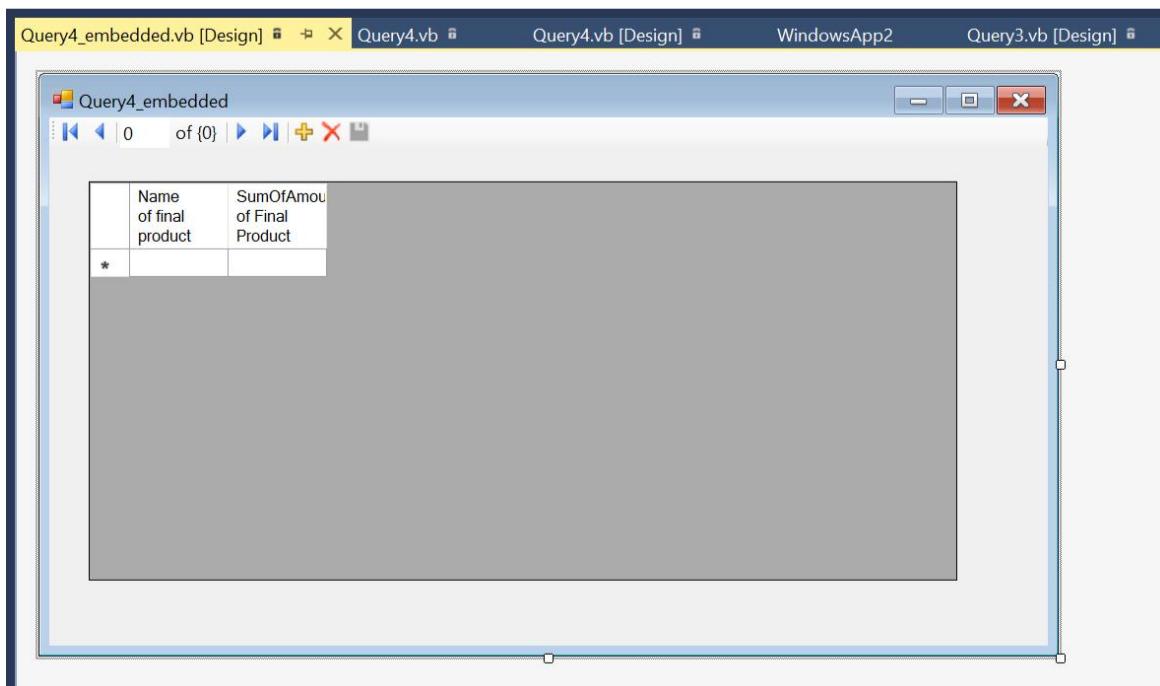
Create a query that presents for each final product the total production in the current year.

Since this is an embedded query, we will use 2 forms for Queries.

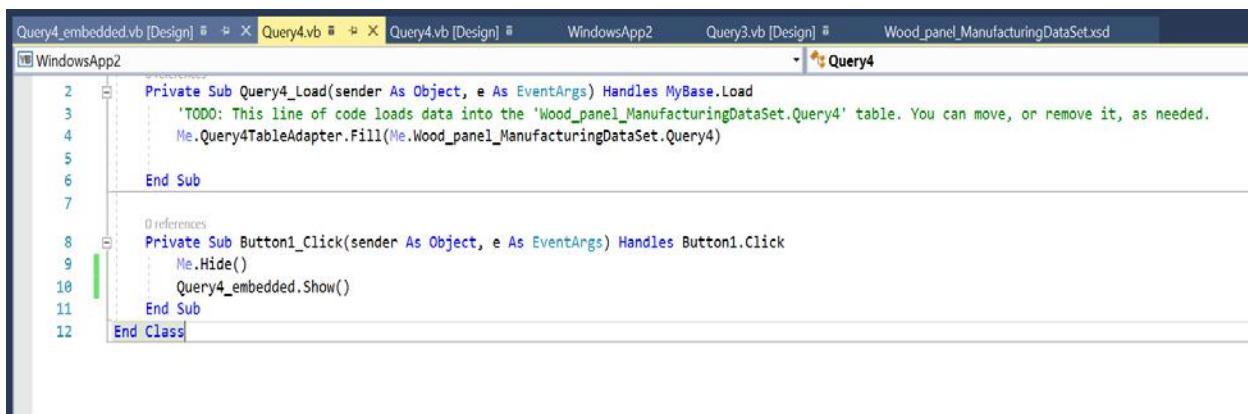
Form design view of Query 4:



Form design view of Query 4-embedded:



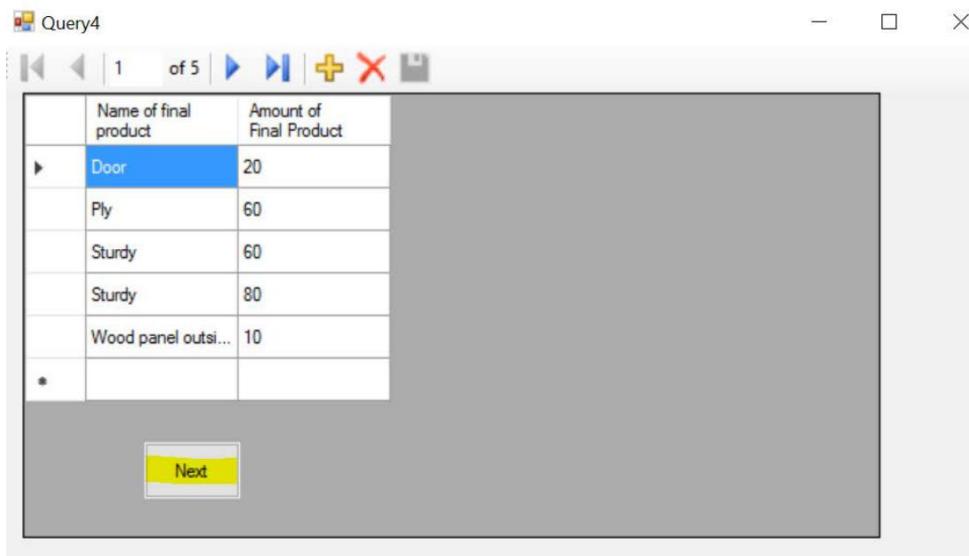
Code view:



The screenshot shows the Visual Studio IDE with multiple tabs at the top: Query4_embedded.vb [Design], Query4.vb [Design], Query4.vb [Design], WindowsApp2, Query3.vb [Design], and Wood_panel_ManufacturingDataSet.xsd. The active tab is Query4.vb [Design]. The code editor contains the following VB.NET code:

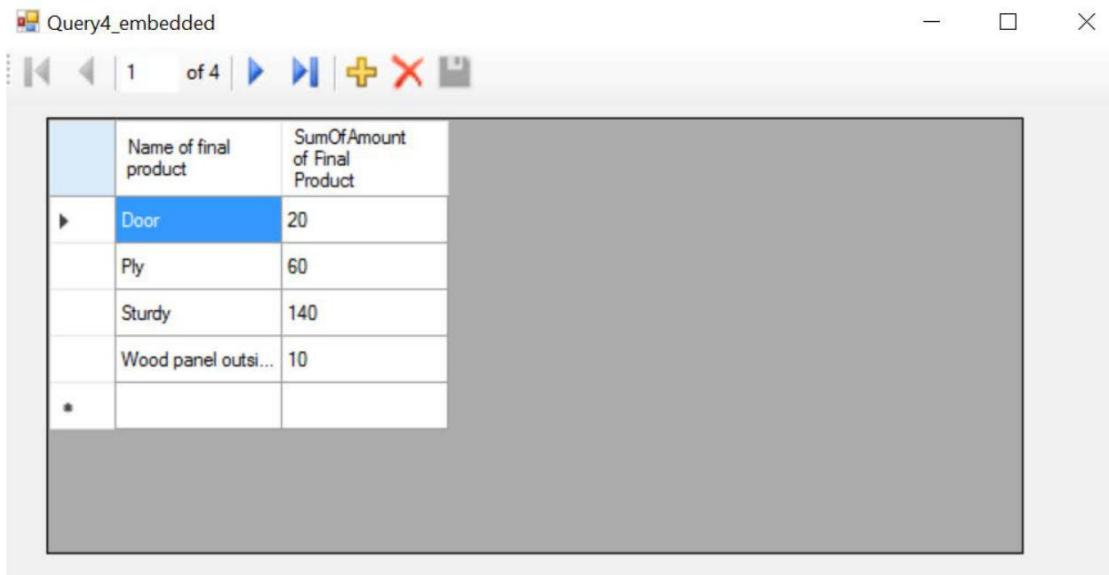
```
2 Private Sub Query4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3     'TODO: This line of code loads data into the 'Wood_panel_ManufacturingDataSet.Query4' table. You can move, or remove it, as needed.
4     Me.Query4TableAdapter.Fill(Me.Wood_panel_ManufacturingDataSet.Query4)
5
6     End Sub
7
8     0 references
9     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
10        Me.Hide()
11        Query4_embedded.Show()
12    End Sub
13
14    End Class
```

Execution view:



The screenshot shows the execution view for Query4. The window title is "Query4". The interface includes navigation buttons (Back, Forward, First, Last, New, Delete, Save), a status bar showing "1 of 5", and a "Next" button at the bottom. A table displays the following data:

	Name of final product	Amount of Final Product
▶	Door	20
	Ply	60
	Sturdy	60
	Sturdy	80
	Wood panel outsi...	10
*		



The screenshot shows the execution view for Query4_embedded. The window title is "Query4_embedded". The interface includes navigation buttons (Back, Forward, First, Last, New, Delete, Save), a status bar showing "1 of 4", and a table displaying the following data:

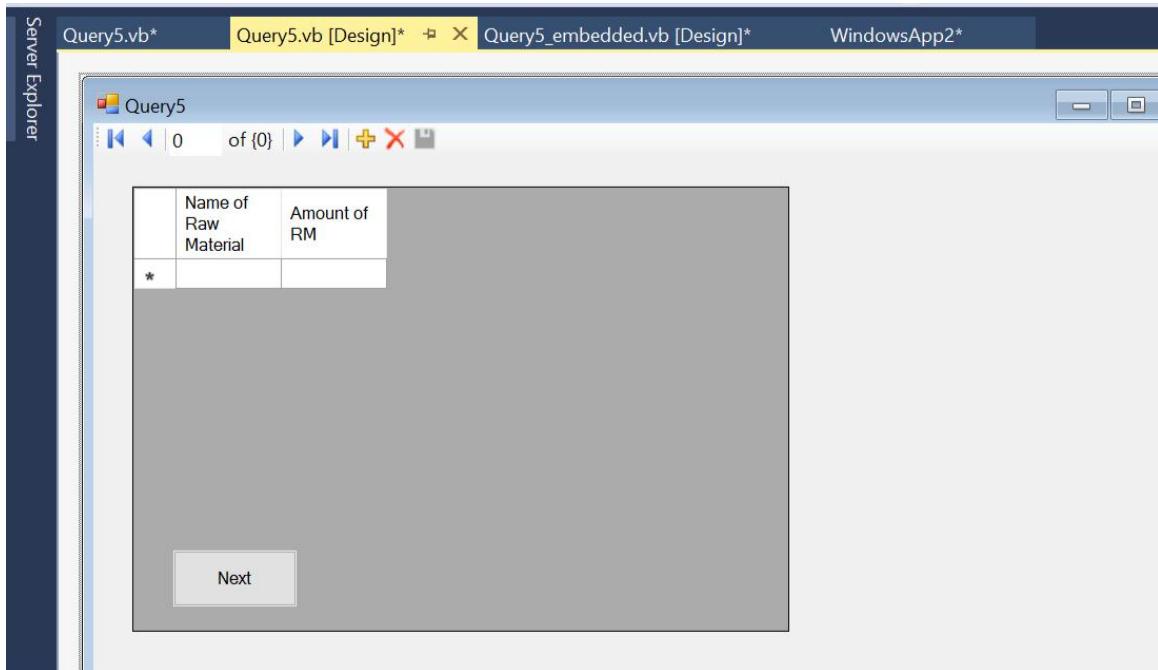
	Name of final product	SumOfAmount of Final Product
▶	Door	20
	Ply	60
	Sturdy	140
	Wood panel outsi...	10
*		

Query 5:

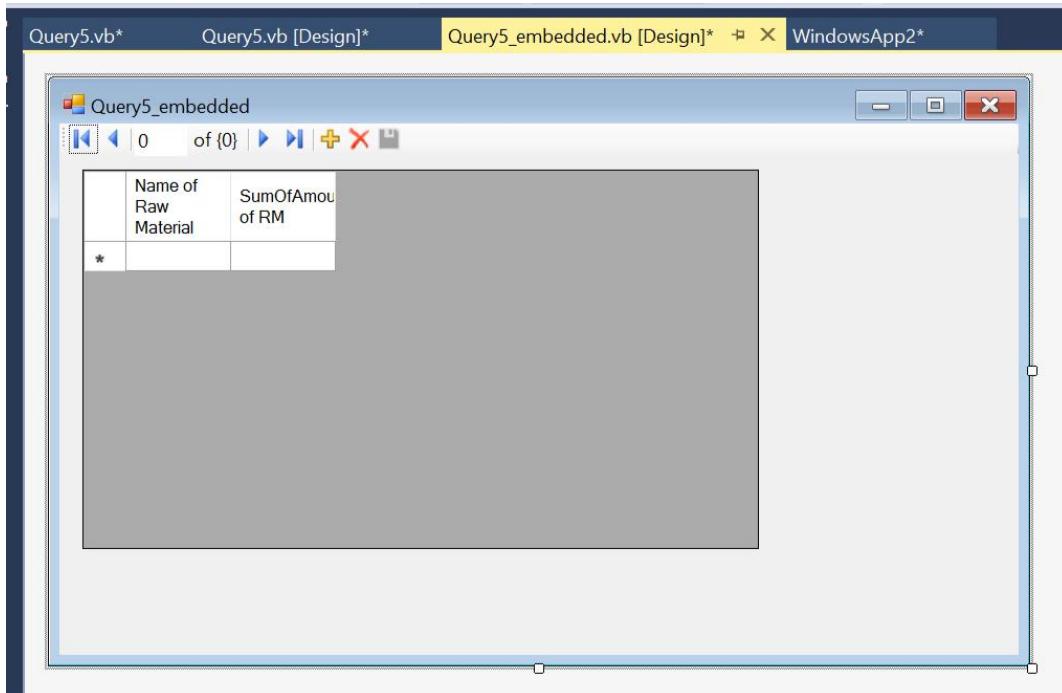
Create a query that presents for each raw material the total consumption in the current year.

Since this is an embedded query, we will use 2 forms for Queries.

Form design view of Query 5:



Form design view of Query 5-embedded:



Code view:

The screenshot shows the Visual Studio IDE with the Query5.vb file open. The code is as follows:

```
2 Private Sub Query5_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3     'TODO: This line of code loads data into the 'Wood_panel_ManufacturingDataSet.Query5' table. You can move, or remove it, as needed.
4     Me.Query5TableAdapter.Fill(Me.Wood_panel_ManufacturingDataSet.Query5)
5
6 End Sub
7
8 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
9     Me.Hide()
10    Query5_embedded.Show()
11 End Sub
12 End Class
```

Execution view:

The screenshot shows the execution view for Query5. It displays a table with two columns: "Name of Raw Material" and "Amount of RM". The data is as follows:

	Name of Raw Material	Amount of RM
▶	chips	10
	flakes	20
	flakes	40
	shreds	20
	strands	40
*		

At the bottom of the window is a "Next" button.

The screenshot shows the execution view for Query5_embedded. It displays a table with three columns: an empty header cell, "Name of Raw Material", and "SumOfAmount of RM". The data is as follows:

	Name of Raw Material	SumOfAmount of RM
▶	chips	10
	flakes	60
	shreds	20
	strands	40
*		

Query 6

Create a query that prompts for the name of a final product, a quality level, and a resistance level and returns the combination (type and quantity) of the raw materials that would produce the required final product.

Query Builder and SQL:

Query Builder

The Query Builder interface shows the following setup:

- Final Product - Density requirements:** Contains columns: * (All Columns), ID, Quality, Resistance.
- Final Product:** Contains columns: Unit Cost, Unit Price, Name of final product, Expected Monthly price, Current inventory level.
- Produce:** Contains columns: * (All Columns), ID (RM), ID (Equipment), ID (Final Product), Transaction ID.
- Raw Material:** Contains columns: * (All Columns), ID, Name of Raw Material, Description, Unit Cost.

Query Grid (Column Alias Table Output Sort Type Sort Order Filter Or... Or... Or...):

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
{Name of final product}		Final Product				= ?			
Quality		Final Product - Density require...				= ?			
Resistance		Final Product - Density require...				= ?			
[ID] [Raw Material]		Final product - RM							
Quantity		Final product - RM							

SQL Generated:

```
SELECT [Final Product].[Name of final product], [Final Product - Density requirements].Quality, [Final Product - Density requirements].Resistance, [Final product - RM].[ID (Raw Material)], [Final product - RM].Quantity
FROM ((([Final Product] INNER JOIN
    [Final Product - Density requirements] ON [Final Product].ID = [Final Product - Density requirements].ID) INNER JOIN
    [Final product - RM] ON [Final Product].ID = [Final product - RM].[ID (Final product)]) INNER JOIN
    Produce ON [Final Product].ID = Produce.[ID (Final Product)]) INNER JOIN
    [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID
WHERE ([Final Product].[Name of final product] <= ?) AND ([Final Product - Density requirements].Quality <= ?) AND ([Final Product - Density requirements].Resistance <= ?)
```

Execute Query OK Cancel

Execution view:

Query Parameters ? X

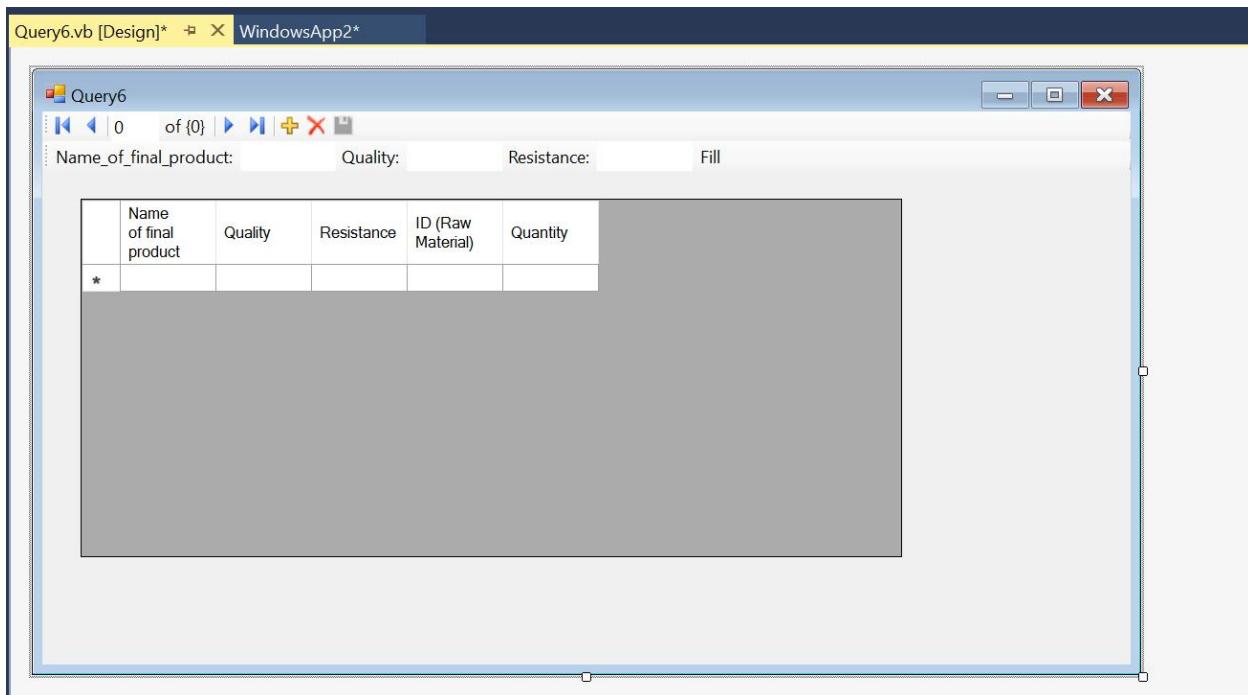
To run this query, enter values for its parameters.

	Name	Value
?	Ply	
?	Q3	
?	R3	

OK Cancel

	Name of final product	Quality	Resistance	ID (Raw Material)	Quantity
▶	Ply	Q3	R3	1	50

Form Design view of Query 6:



Execution view of Query 6:

	Name_of_final_product	Quality	Resistance	Fill
▶	Ply	Q3	R3	
*				

WEB FORM

Design view of form:

The screenshot shows the design view of a web page. The title bar says "Default.aspx". The page content area contains the following text and controls:

```
body|  
Wood Panel Manufacturing  
Select any one of the following:  
○ List of Tables  
○ List of Queries  
Next
```

Code view of form:

The screenshot shows the code view of the Default.aspx page. The title bar says "Default.aspx". The code editor displays the following VB.NET code:

```
2  Partial Class _Default  
3      Inherits System.Web.UI.Page  
4  
5      Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
6          If RadioButton1.Checked Then  
7              Response.Redirect("Tables.aspx")  
8          ElseIf RadioButton2.Checked Then  
9              Response.Redirect("Queries.aspx")  
10         End If  
11     End Sub  
12  
13 End Class
```

Output:

Main form:



Wood Panel Manufacturing

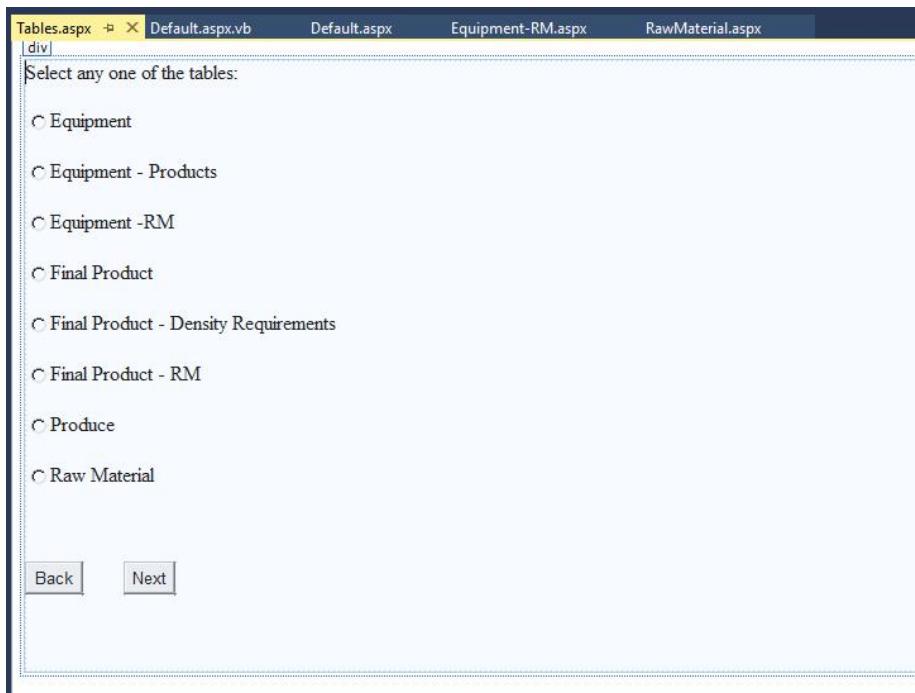
Select any one of the following:

List of Tables

List of Queries

[Next](#)

Design view of the Table form:



Code view of the Table form:

```
Tables.aspx.vb  X  Tables.aspx  Default.aspx.vb  Default.aspx  Equipment-RM.aspx  RawMaterial.aspx
12_Tables.aspx  * Button1

2  Partial Class Tables
3      Inherits System.Web.UI.Page
4
5      Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
6          Response.Redirect("Default.aspx")
7      End Sub
8
9      Protected Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
10         If RadioButton2.Checked Then
11             Response.Redirect("Equipment.aspx")
12         ElseIf RadioButton3.Checked Then
13             Response.Redirect("Equipment-Products.aspx")
14         ElseIf RadioButton4.Checked Then
15             Response.Redirect("Equipment-RM.aspx")
16         ElseIf RadioButton5.Checked Then
17             Response.Redirect("FinalProduct.aspx")
18         ElseIf RadioButton6.Checked Then
19             Response.Redirect("FinalProduct-DensityRequirements.aspx")
20         ElseIf RadioButton7.Checked Then
21             Response.Redirect("FinalProduct-RM.aspx")
22         ElseIf RadioButton8.Checked Then
23             Response.Redirect("Produce.aspx")
24         ElseIf RadioButton9.Checked Then
25             Response.Redirect("RawMaterial.aspx")
26     End If
27 End Sub
28 End Class
29
```

Tables form:



Select any one of the tables:

- Equipment
- Equipment - Products
- Equipment -RM
- Final Product
- Final Product - Density Requirements
- Final Product - RM
- Produce
- Raw Material

[Back](#)

[Next](#)

Design view of the Queries form:

Select any one of the following queries

- C Query that lists the name of final products
- C Query that lists the name of Raw Material
- C Query that lists the name of the final products for a particular equipment
- C Query that lists the total production of final products in current year
- C Query that lists the total consumption of Raw Material in current year
- C Query that lists the combination of Raw Material that would produce the final product.

Back Next

Code view of the Queries form:

```
2 references
2   Partial Class Queries
3     Inherits System.Web.UI.Page
4
5     0 references
6     Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7       Response.Redirect("Default.aspx")
8     End Sub
9
10    0 references
11    Protected Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
12      If RadioButton1.Checked Then
13        Response.Redirect("Query1.aspx")
14      ElseIf RadioButton2.Checked Then
15        Response.Redirect("Query2.aspx")
16      ElseIf RadioButton3.Checked Then
17        Response.Redirect("Query3.aspx")
18      ElseIf RadioButton4.Checked Then
19        Response.Redirect("Query4.aspx")
20      ElseIf RadioButton5.Checked Then
21        Response.Redirect("Query5.aspx")
22      ElseIf RadioButton6.Checked Then
23        Response.Redirect("Query6.aspx")
24      End If
25    End Sub
26  End Class
```

Queries form:



Select any one of the following queries

- Query that lists the name of final products
- Query that lists the name of Raw Material
- Query that lists the name of the final products for a particular equipment
- Query that lists the total production of final products in current year
- Query that lists the total consumption of Raw Material in current year
- Query that lists the combination of Raw Material that would produce the final product.

[Back](#)

[Next](#)

Tables:

1. Equipment table

Design view:

	ID	Name of Equipment	Location	Capacity
Edit	0	abc	abc	0
Delete	1	abc	abc	1
Select	2	abc	abc	2
Edit	3	abc	abc	3
Delete	4	abc	abc	4
Select	5	abc	abc	5
Edit	6	abc	abc	6
Delete	7	abc	abc	7
Select	8	abc	abc	8
Edit	9	abc	abc	9

1 2

AccessDataSource - AccessDataSource1

Output view:

	ID	Name of Equipment	Location	Capacity
Edit	1	wood cutter	L1	5
Delete	2	CNC	L2	6
Select	3	Chain saw	L3	5
Edit	4	File	L4	6
Delete	5	Presser	L5	9
Select	6	Holder	L6	8
Edit	7	Engraver	L7	6
Delete	8	Cold Press	L8	5
Select	9	Veneer Peeling	L9	7
Edit	10	Sand paper	L10	9

2. Equipment – Products

Design view:

	<u>ID</u>	<u>Name of product</u>	<u>Yield</u>
Edit Delete Select	0	abc	0
Edit Delete Select	1	abc	1
Edit Delete Select	2	abc	2
Edit Delete Select	3	abc	3
Edit Delete Select	4	abc	4
Edit Delete Select	5	abc	5
Edit Delete Select	6	abc	6
Edit Delete Select	7	abc	7
Edit Delete Select	8	abc	8
Edit Delete Select	9	abc	9

Output:

	<u>ID</u>	<u>Name of product</u>	<u>Yield</u>
Edit Delete Select	1	Ply	15
Edit Delete Select	4	Wood panel outside	20
Edit Delete Select	5	Sturdy	50
Edit Delete Select	7	Sturdy	25
Edit Delete Select	8	Door	20

3. Equipment –

RM Design view:

	ID	column1
Delete	0	0
Select	1	1
Delete	2	2
Select	3	3
Delete	4	4
Select	5	5
Delete	6	6
Select	7	7
Delete	8	8
Select	9	9

AccessDataSource - AccessDataSource1

Output:

	ID	column1
Delete	1	2
Select	4	3
Delete	5	4
Select	7	2
Delete	8	1

4. Final Product

Design view:

The screenshot shows the ASP.NET design view for a page named 'FinalProduct.aspx'. The page title bar includes links for 'Equipment.aspx', 'Equipment-Products.aspx', 'Equipment-RM.aspx', 'Default.aspx.vb', and 'Default.aspx'. The main content area contains a table with the following columns: ID, Description, Unit Cost, Unit Price, Name of final product, Expected Monthly price, and Current inventory level. The table has 10 rows, each with an 'Edit' and 'Delete' link in the first column. The data in the table is as follows:

	ID	Description	Unit Cost	Unit Price	Name of final product	Expected Monthly price	Current inventory level
Edit	0	abc	0	0	abc	0	0
Edit	1	abc	0.1	0.1	abc	0.1	1
Edit	2	abc	0.2	0.2	abc	0.2	2
Edit	3	abc	0.3	0.3	abc	0.3	3
Edit	4	abc	0.4	0.4	abc	0.4	4
Edit	5	abc	0.5	0.5	abc	0.5	5
Edit	6	abc	0.6	0.6	abc	0.6	6
Edit	7	abc	0.7	0.7	abc	0.7	7
Edit	8	abc	0.8	0.8	abc	0.8	8
Edit	9	abc	0.9	0.9	abc	0.9	9

Below the table, there is a footer section labeled 'AccessDataSource - AccessDataSource1'.

Output:

The screenshot shows the browser output for the 'FinalProduct.aspx' page. The address bar shows the URL 'http://localhost:5927/FinalProduct.aspx'. The page displays a table with the same structure and data as the design view, showing four rows of final products:

	ID	Description	Unit Cost	Unit Price	Name of final product	Expected Monthly price	Current inventory level
Edit	1	Wood	10	5	Wood panel outside	200	100
Edit	2	panel	20	10	Sturdy	100	50
Edit	3	wood1	30	20	Ply	500	25
Edit	4	panel2	25	30	Door	300	90

5. Final Product – Density Requirements

Design view:

FinalProduct-DensityRequirements.aspx* FinalProduct.aspx Equipment.aspx

body

	ID	Quality	Resistance
Edit Delete Select	0	abc	abc
Edit Delete Select	1	abc	abc
Edit Delete Select	2	abc	abc
Edit Delete Select	3	abc	abc
Edit Delete Select	4	abc	abc
Edit Delete Select	5	abc	abc
Edit Delete Select	6	abc	abc
Edit Delete Select	7	abc	abc
Edit Delete Select	8	abc	abc
Edit Delete Select	9	abc	abc

1 [2](#)

AccessDataSource - AccessDataSource1

Output:

http://localhost:5927/FinalProduct-Density Requirements localhost

	ID	Quality	Resistance
Edit Delete Select	1	Q1	R1
Edit Delete Select	2	Q2	R2
Edit Delete Select	3	Q3	R3
Edit Delete Select	4	Q4	R4

6. Final Product – RM

Design view:

FinalProduct-RM.aspx FinalProduct-DensityRequirements.aspx FinalProduct.aspx

	column1	column2	Quantity
Edit Delete Select	0	0	0
Edit Delete Select	1	1	1
Edit Delete Select	2	2	2
Edit Delete Select	3	3	3
Edit Delete Select	4	4	4
Edit Delete Select	5	5	5
Edit Delete Select	6	6	6
Edit Delete Select	7	7	7
Edit Delete Select	8	8	8
Edit Delete Select	9	9	9

1 [2](#)

AccessDataSource - AccessDataSource1

Output:

http://localhost:5927/FinalProduct-RM.asp localhost

	column1	column2	Quantity
Edit Delete Select	1	2	60
Edit Delete Select	2	2	40
Edit Delete Select	2	3	50
Edit Delete Select	3	1	50
Edit Delete Select	4	4	20

7. Produce

Design view:

The screenshot shows the design view of an ASP.NET page. At the top, there's a navigation bar with tabs: 'Produce.aspx*', 'FinalProduct-RM.aspx', 'FinalProduct-DensityRequirements.aspx', 'FinalProduct.aspx', 'Default.aspx.vb', and 'Default.aspx'. Below the navigation bar is a table with 10 rows of data. The table has columns labeled 'column1', 'column2', 'column3', 'Transcation ID', 'Date of produce', 'Amount of Final Product', 'Quality of Final Product', 'Cost of RM', and 'Amount of RM'. Each row contains a 'Edit Delete Select' link followed by numerical values. Below the table is a section labeled 'AccessDataSource - AccessDataSource1'. The entire page is enclosed in a 'body' tag.

	column1	column2	column3	Transcation ID	Date of produce	Amount of Final Product	Quality of Final Product	Cost of RM	Amount of RM
Edit Delete Select 0	0	0	abc	11/5/2018 12:00:00 AM	abc	abc	0	abc	
Edit Delete Select 1	1	1	abc	11/5/2018 12:00:00 AM	abc	abc	0.1	abc	
Edit Delete Select 2	2	2	abc	11/5/2018 12:00:00 AM	abc	abc	0.2	abc	
Edit Delete Select 3	3	3	abc	11/5/2018 12:00:00 AM	abc	abc	0.3	abc	
Edit Delete Select 4	4	4	abc	11/5/2018 12:00:00 AM	abc	abc	0.4	abc	
Edit Delete Select 5	5	5	abc	11/5/2018 12:00:00 AM	abc	abc	0.5	abc	
Edit Delete Select 6	6	6	abc	11/5/2018 12:00:00 AM	abc	abc	0.6	abc	
Edit Delete Select 7	7	7	abc	11/5/2018 12:00:00 AM	abc	abc	0.7	abc	
Edit Delete Select 8	8	8	abc	11/5/2018 12:00:00 AM	abc	abc	0.8	abc	
Edit Delete Select 9	9	9	abc	11/5/2018 12:00:00 AM	abc	abc	0.9	abc	

AccessDataSource - AccessDataSource1

Output:

The screenshot shows the output of the page in a web browser. The address bar displays 'http://localhost:5927/Produce.aspx'. The browser interface includes standard controls like back, forward, search, and refresh. Below the address bar is the same table data as shown in the design view, with 10 rows of information. The table has columns labeled 'column1', 'column2', 'column3', 'Transcation ID', 'Date of produce', 'Amount of Final Product', 'Quality of Final Product', 'Cost of RM', and 'Amount of RM'.

	column1	column2	column3	Transcation ID	Date of produce	Amount of Final Product	Quality of Final Product	Cost of RM	Amount of RM
Edit Delete Select 1	8	3	T3	10/12/2018 12:00:00 AM	60	Q1	8000	10	
Edit Delete Select 2	1	1	T1	10/4/2018 12:00:00 AM	10	Q1	1000	20	
Edit Delete Select 2	7	2	T2	10/11/2018 12:00:00 AM	80	Q2	5000	40	
Edit Delete Select 3	4	2	T2	10/4/2018 12:00:00 AM	60	Q4	6000	40	
Edit Delete Select 4	5	4	T2	10/5/2018 12:00:00 AM	20	Q2	2000	20	

8. Raw Material

Design view:

The screenshot shows the ASP.NET Design view for a page named "RawMaterial.aspx". The title bar includes tabs for "RawMaterial.aspx*", "Produce.aspx", "FinalProduct-RM.aspx", "FinalProduct-DensityRequirements.aspx", "FinalProduct.aspx", and "Default.aspx.vb". The main content area contains a "GridView" control with the following columns: ID, Name of Raw Material, Description, Unit Cost, Expected, Monthly consumption, and Inventory level. The "Edit Delete Select" links are visible next to each row. The data in the grid is as follows:

ID	Name of Raw Material	Description	Unit Cost	Expected	Monthly consumption	Inventory level
0	abc	abc	0	abc		0
1	abc	abc	0.1	abc		1
2	abc	abc	0.2	abc		2
3	abc	abc	0.3	abc		3
4	abc	abc	0.4	abc		4
5	abc	abc	0.5	abc		5
6	abc	abc	0.6	abc		6
7	abc	abc	0.7	abc		7
8	abc	abc	0.8	abc		8
9	abc	abc	0.9	abc		9

Below the grid, there are two buttons labeled "1" and "2". At the bottom left, there is a label "AccessDataSource - AccessDataSource1".

Output:

The screenshot shows the browser output for the URL "http://localhost:5927/RawMaterial.aspx". The page displays a table with the same columns and data as the design view. The data is as follows:

ID	Name of Raw Material	Description	Unit Cost	Expected	Monthly consumption	Inventory level
1	chips	10mm dimension	100	50		20
2	flakes	20mm dimension	200	60		10
3	strands	30mm dimension	300	50		40
4	shreds	40mm dimension	150	10		50

Queries

1. Create a query that prompts for the name of a raw material and lists the name of the final products that can be produced using this raw material. The list should also contain the quantity of raw material needed to produce one unit of final product

Query Builder and SQL:

Configure Data Source - AccessDataSource1

?

X

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure
 Specify columns from a table or view

Name: Equipment

Columns:

*
 ID
 Name of Equipment
 Location
 Capacity

Return only unique rows
WHERE...
ORDER BY...
Advanced...

SELECT statement:

```
SELECT * FROM [Equipment]
```

< Previous Next > Finish Cancel

Configure Data Source - AccessDataSource1

? X



Define Custom Statements or Stored Procedures

Click a tab to create a SQL statement for that operation.

SELECT UPDATE INSERT DELETE

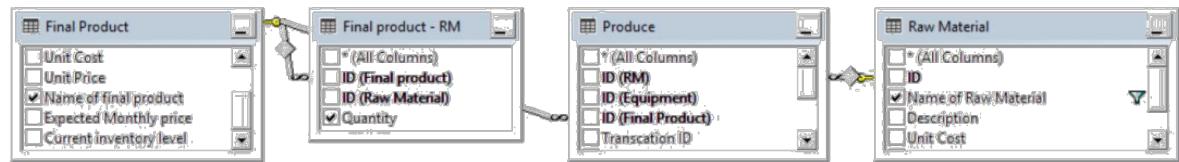
SQL statement:

```
SELECT [Raw Material].[Name of Raw Material], [Final Product].[Name of final product], [Final product - RM].Quantity
FROM ((([Final Product] INNER JOIN [Final product - RM] ON [Final Product].ID = [Final product - RM].[ID (Final product)])) INNER JOIN Produce ON [Final Product].ID = Produce.[ID (Final Product)]) INNER JOIN [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID
WHERE ([Raw Material].[Name of Raw Material] = ?)
```

Query Builder...

< Previous Next > Finish Cancel

Query Builder



Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of Raw Material]		Raw Material	<input checked="" type="checkbox"/>			= ?			
[Name of final product]		Final Product	<input checked="" type="checkbox"/>						
Quantity		Final product - RM	<input checked="" type="checkbox"/>						

```
SELECT [Raw Material].[Name of Raw Material], [Final Product].[Name of final product], [Final product - RM].Quantity
FROM ((([Final Product] INNER JOIN
    [Final product - RM] ON [Final Product].ID = [Final product - RM].[ID (Final product)])) INNER JOIN
    Produce ON [Final Product].ID = Produce.[ID (Final Product)]) INNER JOIN
    [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID
WHERE ([Raw Material].[Name of Raw Material] = ?)
```

Execute Query

OK Cancel

Execution:

Query Parameters ? X

To run this query, enter values for its parameters.

	Name	Value
...	?	chips

OK Cancel

	Name of Raw ...	Name of final ...	Quantity
▶	chips	Ply	50

Design view of Query 1:

Query6.aspx Query3.aspx X Query2.aspx **Query1.aspx*** X

asp:gridview#GridView1

Name of Raw Material	Name of final product	Quantity
abc	abc	0
abc	abc	1
abc	abc	2
abc	abc	3
abc	abc	4
abc	abc	5
abc	abc	6
abc	abc	7
abc	abc	8
abc	abc	9
1	2	

AccessDataSource - AccessDataSource1

Execution view of Query 1:

http://localhost:5927/Query1.aspx

localhost

Name of Raw Material	Name of final product	Quantity
chips	Ply	50

2. Create a query that prompts for the name of a final product and presents a list of the raw materials that can be used. Present the quantity needed of each raw material to produce one unit of the final product.

Query Builder and SQL:

Configure Data Source - AccessDataSource1

?

X

 **Configure the Select Statement**

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure
 Specify columns from a table or view

Name:

Columns:

*
 ID
 Name of Equipment
 Location
 Capacity

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT * FROM [Equipment]

< Previous **Next >** Finish Cancel

QueryBuilder

The QueryBuilder interface displays a query diagram at the top and a results grid below it.

Query Diagram:

- Tables:** Final product - RM, Final Product, Produce, Raw Material.
- Relationships:** Final product - RM is connected to Final Product via ID (Final product). Final Product is connected to Produce via ID (Final Product). Produce is connected to Raw Material via ID (RM).
- Selected Columns:**
 - Final product - RM: * (All Columns), ID (Final product), ID (Raw Material), Quantity.
 - Final Product: Unit Cost, Unit Price, Name of final product (checked), Expected Monthly price, Current inventory level.
 - Produce: * (All Columns), ID (RM), ID (Equipment), ID (Final Product), Transaction ID.
 - Raw Material: All Columns, ID, Name of Raw Material (checked), Description, Unit Cost.

Results Grid:

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of final product]		Final Product	<input checked="" type="checkbox"/>			=?			
[Name of Raw Material]		Raw Material	<input checked="" type="checkbox"/>						
Quantity		Final product - RM	<input checked="" type="checkbox"/>						

SQL Query:

```

SELECT [Final Product].[Name of final product], [Raw Material].[Name of Raw Material], [Final product - RM].Quantity
FROM ((([Final Product] INNER JOIN
    [Final product - RM] ON [Final Product].ID = [Final product - RM].ID (Final product)) INNER JOIN
    Produce ON [Final Product].ID = Produce.ID (Final Product))) INNER JOIN
    [Raw Material] ON Produce.ID (RM) = [Raw Material].ID
  
```

Buttons: Execute Query, OK, Cancel.

Execution view:

Query Parameters ? X

To run this query, enter values for its parameters.

	Name	Value
▶	?	Ply

OK **Cancel**

	Name of...	Name of...	Quantity
▶	Ply	chips	50

Design view of Query 2:

The screenshot shows the Visual Studio IDE with the title bar "Query2.aspx* < X Query1.aspx". On the left, there's a "Toolbox" panel. The main area displays a table with the following data:

Name of final product	Name of Raw Material	Quantity
abc	abc	0
abc	abc	1
abc	abc	2
abc	abc	3
abc	abc	4
abc	abc	5
abc	abc	6
abc	abc	7
abc	abc	8
abc	abc	9
1 2		

Below the table, a placeholder text "AccessDataSource - AccessDataSource1" is visible.

Execution view of Query 2:

The screenshot shows a web browser window with the address bar "http://localhost:5927/Query2.aspx". The page content is a table with the following data:

Name of final product	Name of Raw Material	Quantity
Ply	chips	50

Query 3

Create a query that prompts for the name of an equipment and presents a list of the final products that can be produced using this particular equipment.

Query Builder and SQL:

Configure Data Source - AccessDataSource1 ? X

 **Configure the Select Statement**

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure
 Specify columns from a table or view

Name: Equipment

Columns:

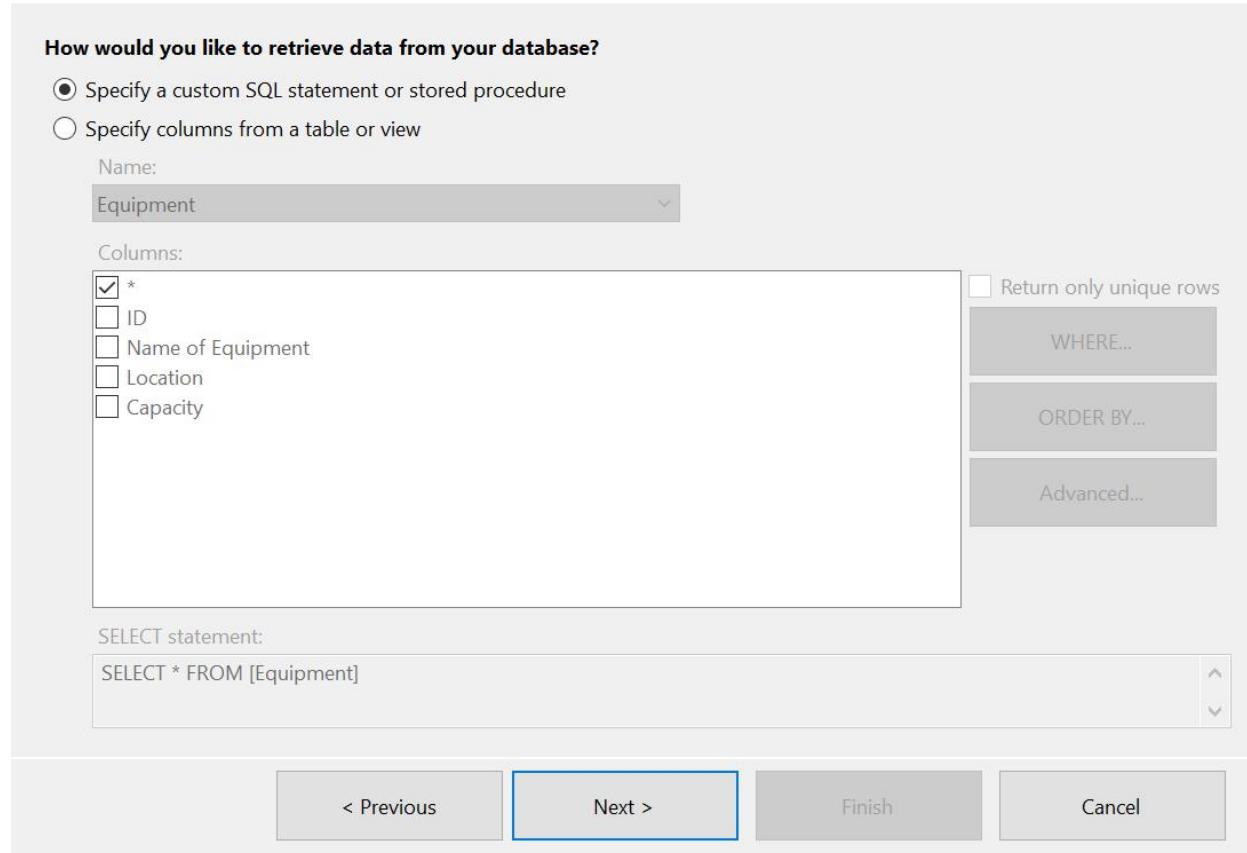
*
 ID
 Name of Equipment
 Location
 Capacity

Return only unique rows
WHERE...
ORDER BY...
Advanced...

SELECT statement:

```
SELECT * FROM [Equipment]
```

< Previous Next > Finish Cancel



Query Builder

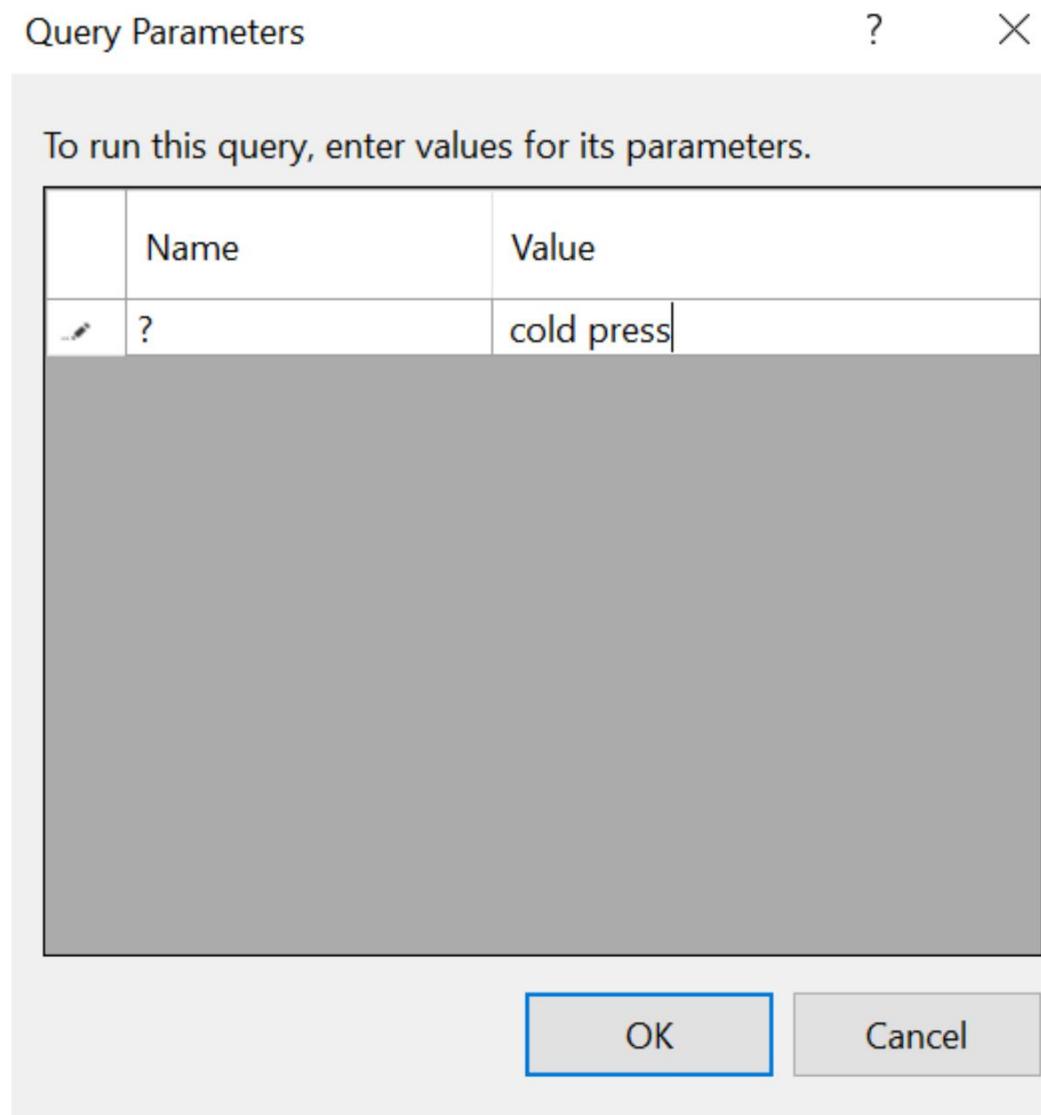
Column	Alias	Table	Equip...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of Equipment]		Equipment	Equipment			=?.			
[Name of product]		Equipment - Products	Equipment						

```

SELECT Equipment.[Name of Equipment], [Equipment - Products].[Name of product]
FROM   ((Equipment INNER JOIN
        [Equipment - Products] ON Equipment.ID = [Equipment - Products].ID) INNER JOIN
        Produce ON Equipment.ID = Produce.[ID (Equipment)]) INNER JOIN
        [Final Product] ON Produce.[ID (Final Product)] = [Final Product].ID
    
```

Execute Query OK Cancel

Execution view:



	Name of...	Name of...
▶	Cold Press	Door

u
n

Design view of Query 3:

The screenshot shows the design view of an ASPX page named "Query3.aspx". The page title bar displays "Query3.aspx*" and "Query2.aspx" and "Query1.aspx". The main content area contains a table with two columns: "Name of Equipment" and "Name of product". Both columns contain the value "abc" for all ten rows. Below the table, there is a link labeled "1 2". At the bottom of the page, there is a section titled "AccessDataSource - AccessDataSource1".

Name of Equipment	Name of product
abc	abc

AccessDataSource - AccessDataSource1

Execution view:

The screenshot shows the execution view of the "Query3.aspx" page in a web browser. The browser address bar shows the URL "http://localhost:5927/Query3.aspx". The page content displays a table with two columns: "Name of Equipment" and "Name of product". The first row shows "Cold Press" and "Door".

Name of Equipment	Name of product
Cold Press	Door

Query 4

Create a query that presents for each final product the total production in the current year.

Query Builder and SQL of Query 4:

Query Builder

The Query Builder interface displays three tables: Equipment, Produce, and Raw Material. The Equipment table has columns: * (All Columns), ID, Name of Equipment, Location, Capacity. The Produce table has columns: * (All Columns), ID (RM), ID (Equipment), ID (Final Product), Transcation ID. The Raw Material table has columns: * (All Columns), ID, Name of Raw Material, Description, Unit Cost. Joins are established between Equipment and Produce (Equipment.ID = Produce.ID (Equipment)), Produce and Raw Material (Produce.ID (RM) = Raw Material.ID). The resulting query table has columns: Column, Alias, Table, Output, Sort Type, Sort Order, Filter, Or..., Or..., Or... with rows: [Name of final product], Final Product, Produce, [Amount of Final Product], Produce, .

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of final product]	Final Product	Produce	<input checked="" type="checkbox"/>						
[Amount of Final Product]	Produce		<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
SELECT [Final Product].[Name of final product], Produce.[Amount of Final Product]
FROM (((Equipment INNER JOIN
        Produce ON Equipment.ID = Produce.[ID (Equipment)]) INNER JOIN
        [Final Product] ON Produce.[ID (Final Product)] = [Final Product].ID) INNER JOIN
        [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID)
```

Execute Query OK Cancel

Query Builder and SQL of Query 4 - embedded:

Configure Data Source - AccessDataSource1

?

X

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure

Specify columns from a table or view

Name:

Columns:

*
 Name of final product
 SumOfAmount of Final Product

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT * FROM [Query4_embedded]

< Previous

Next >

Finish

Cancel

Design view of Query 4:

Since this is an embedded query, we have to use 2 queries.

The screenshot shows the design view of a web page. The title bar includes tabs for "Query4.aspx.vb", "Query4_embedded.aspx*", "Query4.aspx*", "Query6.aspx", and "Query1.aspx*". The main content area contains a table with two columns: "Name of final product" and "Amount of Final Product". Both columns contain the value "abc". Below the table is a "Next" button. A tooltip "AccessDataSource - AccessDataSource1" points to the button.

The screenshot shows the code editor for "Query4.aspx". The code defines a partial class "Query4" that inherits from "System.Web.UI.Page". It contains a single method "Button1_Click" which uses "Response.Redirect" to redirect to "Query4_embedded.aspx".

```
2  Partial Class Query4
3      Inherits System.Web.UI.Page
4
5      References
6      Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7          Response.Redirect("Query4_embedded.aspx")
8      End Sub
9  End Class
```

Query 4_embedded

The screenshot shows the design view of "Query4_embedded.aspx". The title bar includes tabs for "Query4.aspx.vb", "Query4_embedded.aspx*", "Query4.aspx*", "Query6.aspx", and "Query1.aspx*". The main content area contains a table with two columns: "Name of final product" and "SumOfAmount of Final Product". The first column has values "abc", "abc", "abc", "abc", and "abc". The second column has values "0", "0.1", "0.2", "0.3", and "0.4". Below the table is a "Next" button. A tooltip "AccessDataSource - AccessDataSource1" points to the button.

Output:

The screenshot shows a Microsoft Internet Explorer browser window with the URL <http://localhost:5927/Query4.aspx>. The page displays a table with the following data:

	Name of final product	Amount of Final Product
Edit	Wood panel outside	10
Edit	Sturdy	80
Edit	Sturdy	60
Edit	Ply	60
Edit	Door	20

Below the table is a yellow rectangular button containing the text "Next".

The screenshot shows a Microsoft Internet Explorer browser window with the URL http://localhost:5927/Query4_embedded.aspx. The page displays a table with the following data:

Name of final product	SumOfAmount of Final Product
Door	20
Ply	60
Sturdy	140
Wood panel outside	10

Query 5

Create a query that presents for each raw material the total consumption in the current year.

Query Builder and SQL of Query 5

Query Builder

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of Raw Material]		Raw Material	<input checked="" type="checkbox"/>						
[Amount of RM]		Produce	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
SELECT [Raw Material].[Name of Raw Material], Produce.[Amount of RM]
FROM (((Equipment INNER JOIN
    Produce ON Equipment.ID = Produce.[ID (Equipment)])) INNER JOIN
    [Final Product] ON Produce.[ID (Final Product)] = [Final Product].ID) INNER JOIN
    [Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID
```

Execute Query OK Cancel

Query Builder and SQL of Query 5 - embedded:

Configure Data Source - AccessDataSource1

? X



Configure the Select Statement

How would you like to retrieve data from your database?

- Specify a custom SQL statement or stored procedure
- Specify columns from a table or view

Name:

Query5_embedded

Columns:

- *
- Name of Raw Material
- SumOfAmount of RM

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT * FROM [Query5_embedded]

< Previous

Next >

Finish

Cancel

Design view of Query 5

Since this is an embedded query, we have to use 2 queries.

Name of Raw Material Amount of RM

abc	abc

AccessDataSource - AccessDataSource1

Next

```
2  Partial Class Query5
3      Inherits System.Web.UI.Page
4
5      References
6      Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7          Response.Redirect("Query5_embedded.aspx")
8      End Sub
9  End Class
```

Design view of Query 5-embedded

Name of Raw Material SumOfAmount of RM

abc	0
abc	0.1
abc	0.2
abc	0.3
abc	0.4

AccessDataSource - AccessDataSource1

Output:

A screenshot of a Microsoft Internet Explorer browser window. The address bar shows the URL <http://localhost:5927/Query5.aspx>. The page content displays a table with two columns: "Name of Raw Material" and "Amount of RM". The data rows are: chips (10), flakes (20), flakes (40), strands (40), and shreds (20). A "Next" button is visible at the bottom left.

Name of Raw Material	Amount of RM
chips	10
flakes	20
flakes	40
strands	40
shreds	20

A screenshot of a Microsoft Internet Explorer browser window. The address bar shows the URL http://localhost:5927/Query5_embedded.aspx. The page content displays a table with two columns: "Name of Raw Material" and "SumOfAmount of RM". The data rows are: chips (10), flakes (60), shreds (20), and strands (40).

Name of Raw Material	SumOfAmount of RM
chips	10
flakes	60
shreds	20
strands	40

Query 6

Create a query that prompts for the name of a final product, a quality level, and a resistance level and returns the combination (type and quantity) of the raw materials that would produce the required final product.

Query Builder and SQL:

Query Builder

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
[Name of final product]		Final Product				= ?			
Quality		Final Product - Density requirements				= ?			
Resistance		Final Product - Density requirements				= ?			
[ID (Raw Material)]		Final product - RM							
Quantity		Final product - RM							

```
SELECT [Final Product].[Name of final product], [Final Product - Density requirements].Quality, [Final Product - Density requirements].Resistance, [Final product - RM].[ID (Raw Material)], [Final product - RM].Quantity
FROM (((Final Product - Density requirements) INNER JOIN
[Final Product] ON [Final Product - Density requirements].ID = [Final Product].ID) INNER JOIN
[Produce] ON [Final Product].ID = Produce.[ID (Final Product)]) INNER JOIN
[Raw Material] ON Produce.[ID (RM)] = [Raw Material].ID) INNER JOIN
[Final product - RM] ON [Final Product].ID = [Final product - RM].[ID (Final product)]
WHERE ([Final Product].[Name of final product] = ?) AND ([Final Product - Density requirements].Quality = ?) AND ([Final Product - Density requirements].Resistance = ?)
```

Execute Query OK Cancel

Execution view:

Query Parameters

To run this query, enter values for its parameters.

	Name	Value
?	Ply	
?	Q3	
?	R3	

OK **Cancel**

	Name of...	Quality	Resistance	ID (Raw ...	Quantity
▶	Ply	Q3	R3	1	50

Design view of Query 6

	Name of final product	Quality	Resistance	ID (Raw Material)	Quantity
Edit	Delete	Select	abc	0	0
Edit	Delete	Select	abc	1	1
Edit	Delete	Select	abc	2	2
Edit	Delete	Select	abc	3	3
Edit	Delete	Select	abc	4	4
Edit	Delete	Select	abc	5	5
Edit	Delete	Select	abc	6	6
Edit	Delete	Select	abc	7	7
Edit	Delete	Select	abc	8	8
Edit	Delete	Select	abc	9	9

1 2

AccessDataSource - AccessDataSource1

Execution view of Query 6

	Name of final product	Quality	Resistance	ID (Raw Material)	Quantity		
Edit	Delete	Select	Ply	Q3	R3	1	50