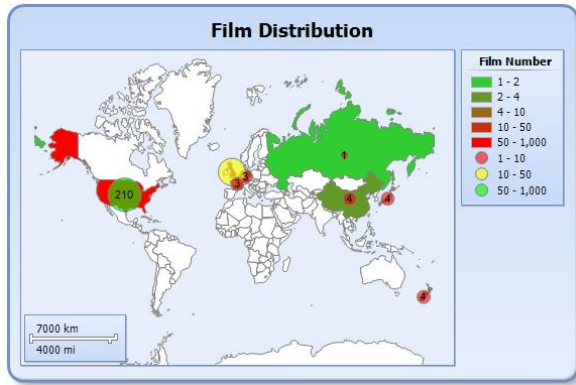
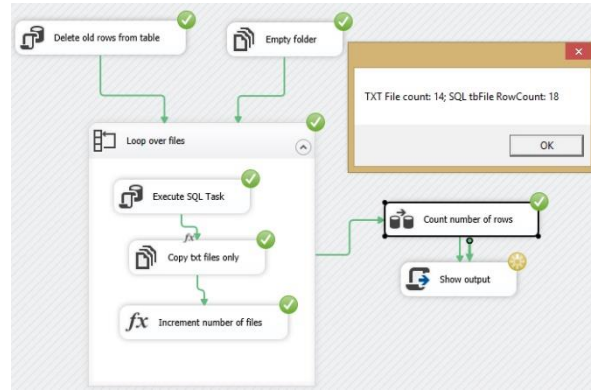


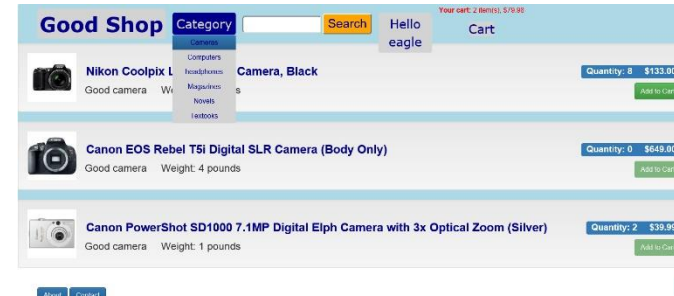
SQL Server and Programming Demos



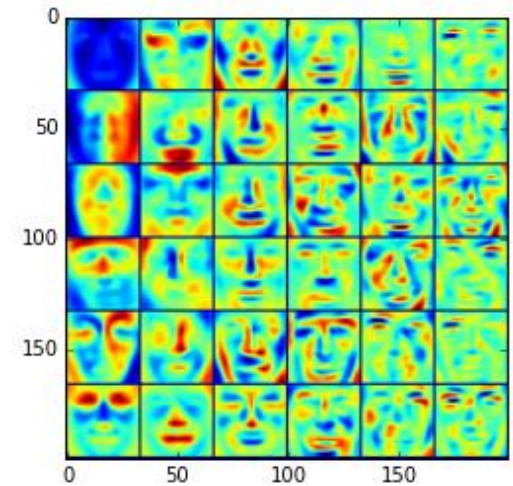
SSRS



SSIS



Website



Python

Qi Charles Sun

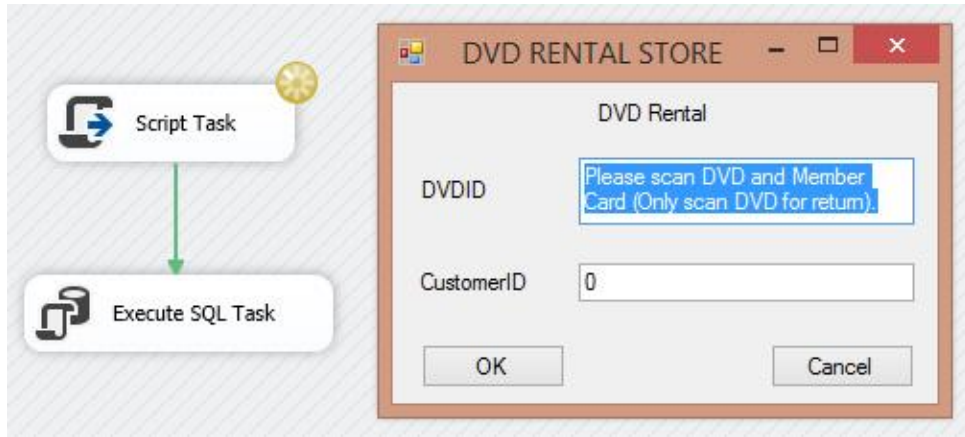
Outline

- 1.Database demos using T-SQL, SSIS, C#
- 2.Website demos
- 3.SSRS demos
- 4.SSIS demos
- 5.SSAS demos
- 6.Excel BI demos
- 7.Programing demos (Python, Matlab, C#, and Labview)

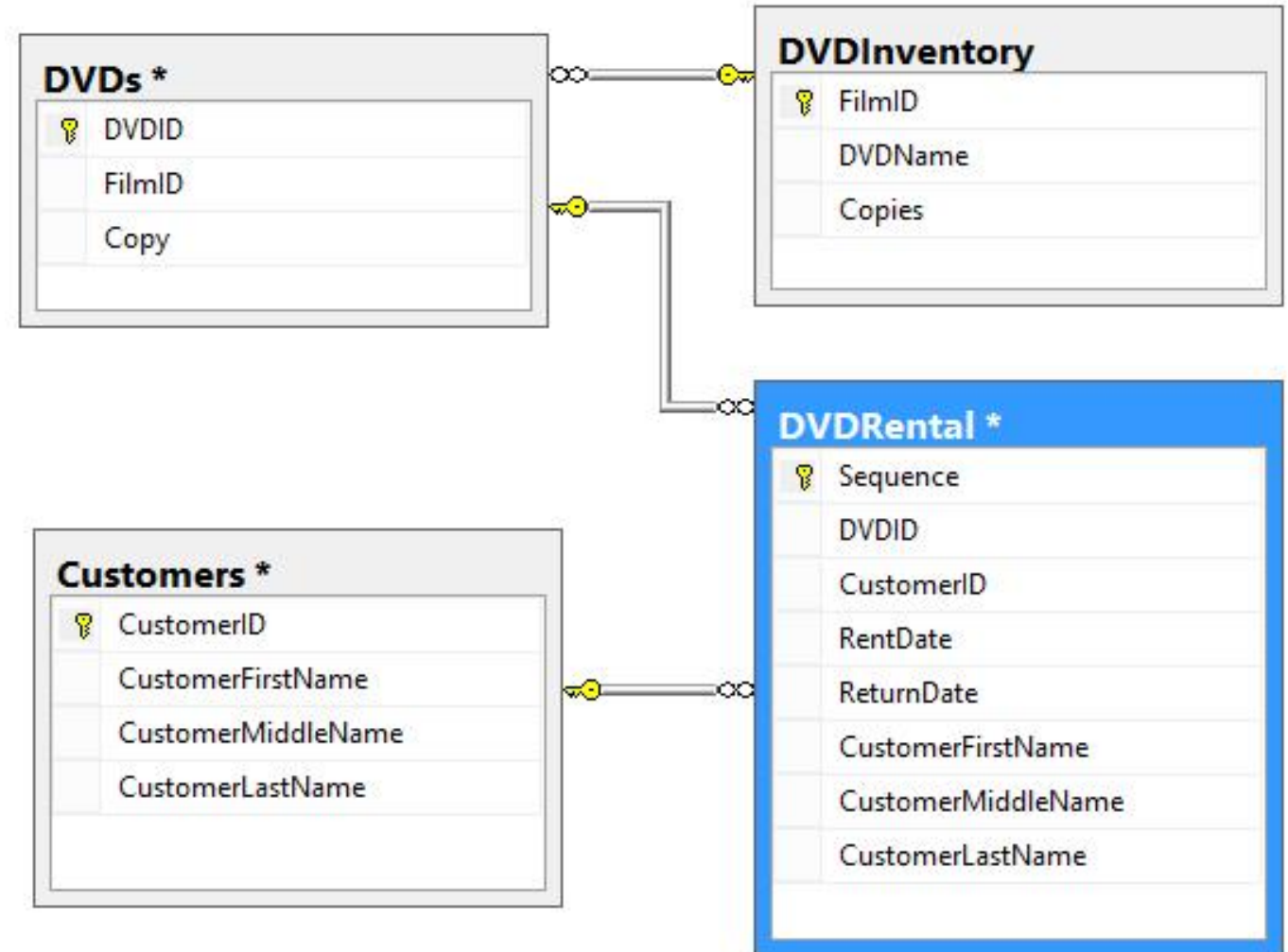
Objective of Project (database1)

1. Build a OLTP database for DVD rental
2. Created 4 tables with Primary Key and Foreign Key constraints
3. Recorded every transaction for return and rental
4. Input the new member to Table Customers
5. Minimized inputs: scan of DVD and member card for rental, and scan of DVD for return
6. Two triggers and one stored procedure to accomplish the goal
7. SSIS and C# generated two prompt windows for inputs

Database diagram and prompt input window



4 tables connected by primary key and foreign key constraints. Tables can contain more columns for details such as DVD condition, member address. The database has its basic functions. The input window was designed using two windows forms by C#.



Different functionalities

INPUT MEMBER INFOR...

New member, please input name.

CustomerFirstName John

CustomerMiddleName NULL

CustomerLastName James

OK Cancel

INPUT MEMBER INFOR...

Return, Just click OK.

CustomerFirstName NULL

CustomerMiddleName NULL

CustomerLastName NULL

OK Cancel

INPUT MEMBER INFOR...

Old member, don't input name.

CustomerFirstName NULL

CustomerMiddleName NULL

CustomerLastName NULL

OK Cancel

DVD RENTAL STORE

DVD Rental

DVDID 18

CustomerID 5

OK Cancel

Two triggers

```
CREATE TRIGGER Rent ON DVD Rental
AFTER INSERT
AS
BEGIN
    UPDATE DVDInventory
    SET DVDInventory.Copies =
    (SELECT DVDInventory.Copies - 1
    FROM DVDs INNER JOIN inserted
    ON DVDs.DVDID = inserted.DVDID
    INNER JOIN DVDInventory ON DVDInventory.FilmID = DVDs
    WHERE DVDInventory.FilmID = (SELECT FilmID FROM inserted
    INNER JOIN DVDs ON DVDInventory.FilmID = DVDs.FilmID))

    UPDATE DVDs
    SET Copy = 0
    WHERE DVDID = (SELECT DVDID FROM inserted)
END;
GO
```

```
CREATE TRIGGER DVDReturn ON DVD Rental
AFTER UPDATE
AS
BEGIN
    UPDATE DVDInventory
    SET DVDInventory.Copies =
    (SELECT DVDInventory.Copies + 1
    FROM DVDInventory AS a INNER JOIN DVDs AS b ON a.FilmID = b.FilmID
    INNER JOIN inserted AS c ON c.DVDID = b.DVDID)
    WHERE DVDInventory.FilmID = (SELECT FilmID FROM inserted
    ON a.DVDID = b.DVDID)

    UPDATE DVDs
    SET Copy = 1
    WHERE DVDID = (SELECT DVDID FROM inserted)
END;
GO
```

One stored procedure

Used IF clause to finish different INSERT and UPDATE, and remind the DVD is out of stock.

```
CREATE PROCEDURE dbo.uspRental @DVDID int,
    @CustomerID int,
    --@ReturnDate datetime = NULL,
    @CustomerFirstName nvarchar(255) = NULL,
    @CustomerMiddleName nvarchar(255) = NULL,
    @CustomerLastName nvarchar(255) = NULL
AS
BEGIN
    SET NOCOUNT ON;
    IF (SELECT Copy FROM DVDs WHERE DVDID = @DVDID) = 1
    BEGIN
        IF @CustomerID IN (SELECT CustomerID FROM Customers)
        BEGIN
            INSERT INTO DVDrental (DVDID, CustomerID, RentDate, CustomerFirstName, CustomerMiddleName, CustomerLastName)
            VALUES(@DVDID, @CustomerID, GETDATE(),
                (SELECT CustomerFirstName FROM Customers WHERE CustomerID = @CustomerID),
                (SELECT CustomerMiddleName FROM Customers WHERE CustomerID = @CustomerID),
                (SELECT CustomerLastName FROM Customers WHERE CustomerID = @CustomerID))
        END
        ELSE
        BEGIN
            SET NOCOUNT ON;
            INSERT INTO Customers
            VALUES(@CustomerID, @CustomerFirstName, @CustomerMiddleName, @CustomerLastName)
            INSERT INTO DVDrental (DVDID, CustomerID, RentDate, CustomerFirstName, CustomerMiddleName, CustomerLastName)
            VALUES (@DVDID, @CustomerID, GETDATE(), @CustomerFirstName, @CustomerMiddleName, @CustomerLastName)
        END
    END
    ELSE
    BEGIN
        UPDATE DVDrental
        SET ReturnDate = GETDATE(), DVDID = @DVDID
        WHERE Sequence = (SELECT MAX(Sequence) FROM DVDrental)
    END
    IF (SELECT Copies FROM DVDInventory AS a INNER JOIN DVDs AS b ON a.FilmID = b.FilmID WHERE b.DVDID = @DVDID) = 0
    PRINT 'OUT OF STOCK'
END;
GO
```

C# code demo

Used two windows forms to accomplished the goal. The only input is scan of DVD and member car for rental, and scan of DVD for return. If the member is new, the system can record his/her information.

```
public Form2()
{
    InitializeComponent();
}
1 reference
private void Form2_Load(object sender, EventArgs e)
{
    CustomerFirstName.Text = "NULL";
    CustomerMiddleName.Text = "NULL";
    CustomerLastName.Text = "NULL";

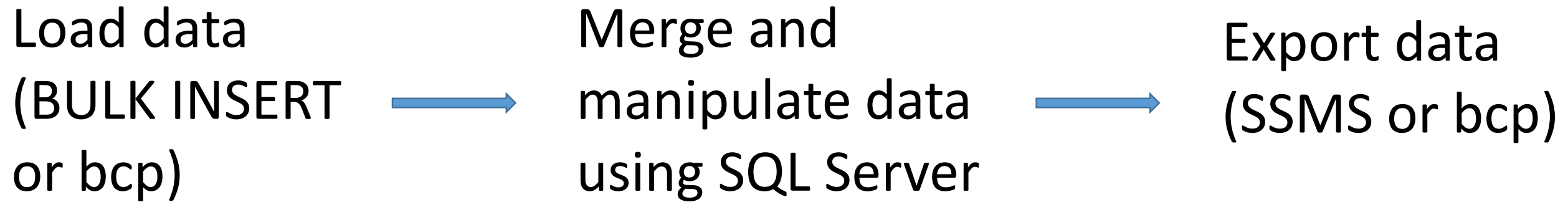
    SqlConnection myConnection = new SqlConnection("server = eagle;" + "Trusted_Connection = yes;"
        + "database = DVDLibrary;" + "Connection timeout = 30");
    myConnection.Open();
    SqlCommand myCommand = new SqlCommand("SELECT CustomerID FROM Customers", myConnection);
    SqlDataReader NewOrOld = myCommand.ExecuteReader();
    ScriptMain IDs = new ScriptMain();
    bool Showed = true;
    var b = IDs.FirstForm(Showed);
    int a = b.Item2;

    while (NewOrOld.Read())
    {
        if (NewOrOld.GetValue(0).ToString() != Convert.ToString(a))
        {
            Indicator.Text = "New member, please input name.";
        }
        else
        {
            Indicator.Text = "Old member, don't input name.";
            break;
        }
    }
    if (a == 0)
    {
        Indicator.Text = "Return, Just click OK.";
    }
}
```


Objective of Project (database2)

1. Build a database with the function to store error information
 2. Created tables with setting Primary Key and Non-Clustered Index
 3. Loaded all frequency range spectroscopic data to the database using “BULK INSERT” or “bcp” in PowerShell
 4. Using SQL commends to calculate and correct data, and saved the results to new tables
 5. Merged the data using “UNION” commend for the FIR range data, and saved the result to the created table
 6. Merged the shifted and shortened data using “UNION ALL” for all frequency range data, and saved the result to the created table
- Tables: All frequency range tables and merged tables saved to the schema [2H]

Design Diagram



Create Database

- Creating a database that specifies the data and transaction log files
- Alter the database SET options:

RECOVERY SIMPLE: A simple backup strategy that uses minimal log

ANSI_NULLS: All comparisons to a null value evaluate to UNKNOWN

ANSI_PADDING: Strings are padded to the same length

ANSI_WARNINGS: Errors or warnings are issued (divide-by-zero or null values in aggregate functions).

ARITHABORT: A query is ended (an overflow or divide-by-zero error)

CONCAT_NULL_YIELDS_NULL: The result of a concatenation operation is NULL when either operand is NULL

QUOTED_IDENTIFIER: Double quotation marks can be used to enclose delimited identifiers.

NUMERIC_ROUNDABORT: An error is generated when loss of precision occurs in an expression.

PAGE_VERIFY CHECKSUM: Discovers damaged database pages caused by disk I/O path errors

ALLOW_SNAPSHOT_ISOLATION: Turns off the Snapshot option at the database level

```
CREATE DATABASE [Spectroscopy]
    ON (NAME = 'Spectroscopy_Data', FILENAME =
N'E:\SQL_Server_Practice\Spectroscopy_Data.mdf'
, SIZE = 5, FILEGROWTH = 8) -- SIZE = 120,
    LOG ON (NAME = 'Spectroscopy_Log', FILENAME
=
N'E:\SQL_Server_Practice\Spectroscopy_Log.ldf'
, SIZE = 2, FILEGROWTH = 96);
GO
```

```
ALTER DATABASE [Spectroscopy]
SET RECOVERY SIMPLE,
    ANSI_NULLS ON,
    ANSI_PADDING ON,
    ANSI_WARNINGS ON,
    ARITHABORT ON,
    CONCAT_NULL_YIELDS_NULL ON,
    QUOTED_IDENTIFIER ON,
    NUMERIC_ROUNDABORT OFF,
    PAGE_VERIFY CHECKSUM,
    ALLOW_SNAPSHOT_ISOLATION OFF;
GO
```

Create table to store error information

ON [PRIMARY] will create the structures on the "Primary" filegroup

Alter the table to add a clustered index

Create a stored procedure

uspPrintError prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.

```
CREATE TABLE [dbo].[ErrorLog](
    [ErrorLogID] [int] IDENTITY (1, 1) NOT NULL,
    [ErrorTime] [datetime] NOT NULL
    CONSTRAINT [DF_ErrorLog_ErrorTime] DEFAULT (GETDATE()),
    [UserName] [sysname] NOT NULL,
    [ErrorNumber] [int] NOT NULL,
    [ErrorSeverity] [int] NULL,
    [ErrorState] [int] NULL,
    [ErrorProcedure] [nvarchar](126) NULL,
    [ErrorLine] [int] NULL,
    [ErrorMessage] [nvarchar](4000) NOT NULL
) ON [PRIMARY];
GO

ALTER TABLE [dbo].[ErrorLog] WITH CHECK ADD
    CONSTRAINT [PK_ErrorLog_ErrorLogID] PRIMARY KEY CLUSTERED
    (
        [ErrorLogID]
    ) ON [PRIMARY];
GO
CREATE PROCEDURE [dbo].[uspPrintError]
AS
BEGIN
    SET NOCOUNT ON;

    -- Print error information.
    PRINT 'Error ' + CONVERT(varchar(50), ERROR_NUMBER()) +
        ', Severity ' + CONVERT(varchar(5), ERROR_SEVERITY()) +
        ', State ' + CONVERT(varchar(5), ERROR_STATE()) +
        ', Procedure ' + ISNULL(ERROR_PROCEDURE(), '-') +
        ', Line ' + CONVERT(varchar(5), ERROR_LINE());
    PRINT ERROR_MESSAGE();
END;
GO
```


Create a stored procedure

uspLogError logs error information in the ErrorLog table about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. This should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.

```
CREATE PROCEDURE [dbo].[uspLogError]
    @ErrorLogID [int] = 0 OUTPUT -- contains the ErrorLogID of the row inserted
                                -- by uspLogError in the ErrorLog table
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorLogID = 0;
    BEGIN TRY
        IF ERROR_NUMBER() IS NULL
            RETURN;
        IF XACT_STATE() = -1
            BEGIN
                PRINT 'Cannot log error since the current transaction is in an uncommittable
state. '
                    + 'Rollback the transaction before executing uspLogError in order to
successfully log error information.';
                RETURN;
            END
        INSERT [dbo].[ErrorLog]
            (
                [UserName],
                [ErrorNumber],
                [ErrorSeverity],
                [ErrorState],
                [ErrorProcedure],
                [ErrorLine],
                [ErrorMessage]
            )
        VALUES
            (
                CONVERT(sysname, CURRENT_USER),
                ERROR_NUMBER(),
                ERROR_SEVERITY(),
                ERROR_STATE(),
                ERROR_PROCEDURE(),
                ERROR_LINE(),
                ERROR_MESSAGE()
            );
        SET @ErrorLogID = @@IDENTITY;
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred in stored procedure uspLogError: ';
        EXECUTE [dbo].[uspPrintError];
        RETURN -1;
    END CATCH
END;
GO
```

Create Schema [2H]

AUTHORIZATION: Specifies the name of the database-level principal that will own the schema

Create Tables for storing all frequency range experimental data

All tables are stored in the schema [2H] on the "Primary" filegroup

There are two NOT NULL columns for each table

```
CREATE SCHEMA [2H] AUTHORIZATION [dbo];
GO

CREATE TABLE [2H].[UV](
    [Frequency] [float] NOT NULL,
    [Intensity] [float] NOT NULL
) ON [PRIMARY];
GO

CREATE TABLE [2H].[NIR](
    [Frequency] [float] NOT NULL,
    [Intensity] [float] NOT NULL
) ON [PRIMARY];
GO

CREATE TABLE [2H].[MIR](
    [Frequency] [float] NOT NULL,
    [Intensity] [float] NOT NULL
) ON [PRIMARY];
GO

CREATE TABLE [2H].[FIR_35](
    [Frequency] [float] NOT NULL,
    [Intensity] [float] NOT NULL
) ON [PRIMARY];
GO
```

Load data

Insert values to the table using
“BULK INSERT”

```
BULK INSERT [Spectroscopy].[2H].[UV_A1] FROM
N'E:\SQL_Server_Practice\2H\_2H_AL_U.SP_xy.asc'
WITH (
    KEEPIDENTITY,
    TABLOCK
);
```

Alter the table and set a clustered
index on Frequency column

```
ALTER TABLE [2H].[UV_A1] WITH CHECK ADD
    CONSTRAINT [PK_UV_A1_Frequency] PRIMARY KEY
    CLUSTERED
    (
        [Frequency]
    ) ON [PRIMARY];
GO
```

Create a non-clustered index to
enhance the operating efficiency.

```
CREATE NONCLUSTERED INDEX [IX_UV_A1_Intensity]
ON [2H].[UV_A1]([Intensity]) ON [PRIMARY];
GO
```

Merge FIR range data

Merge the different FIR
frequency range data using
“UNION” command

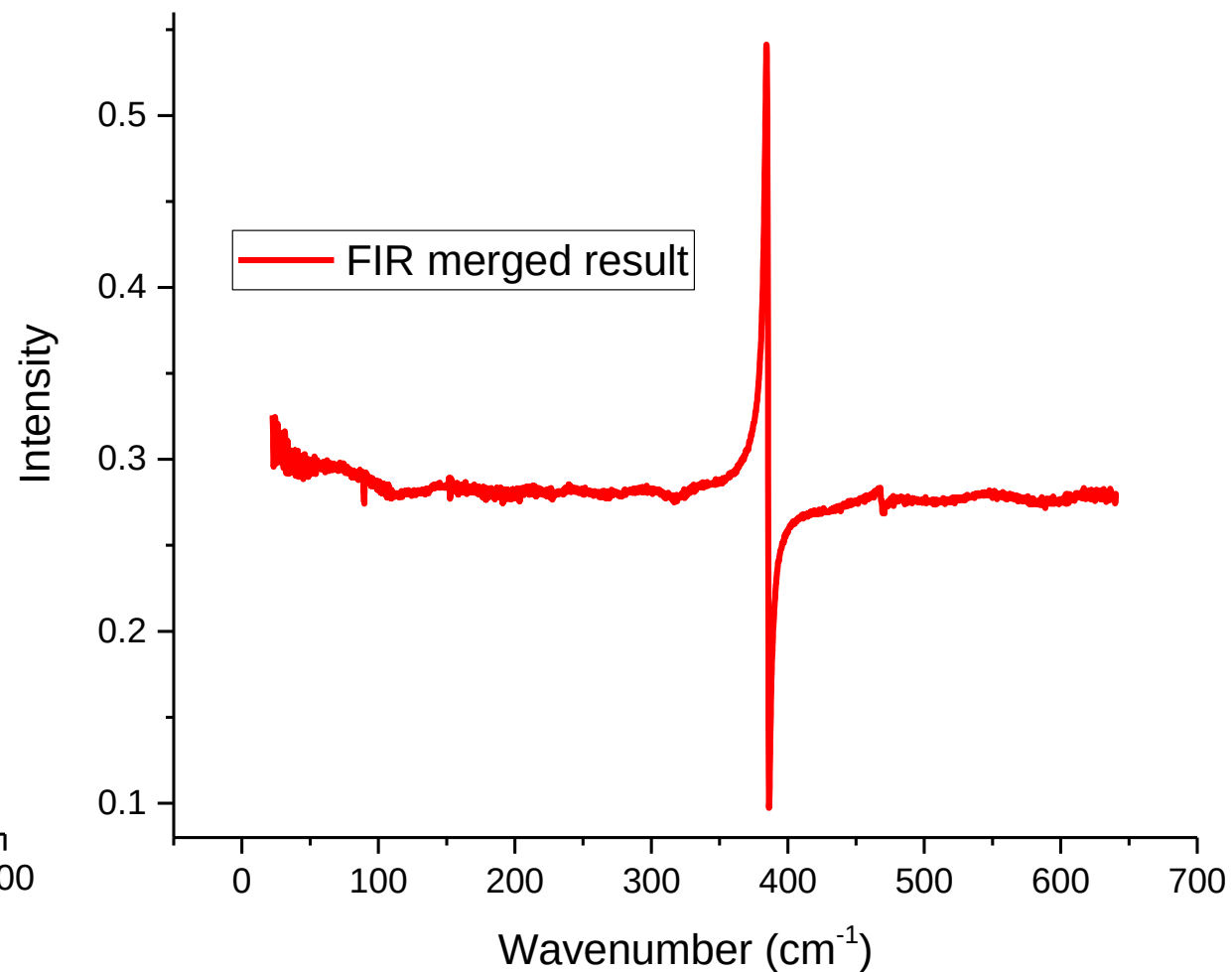
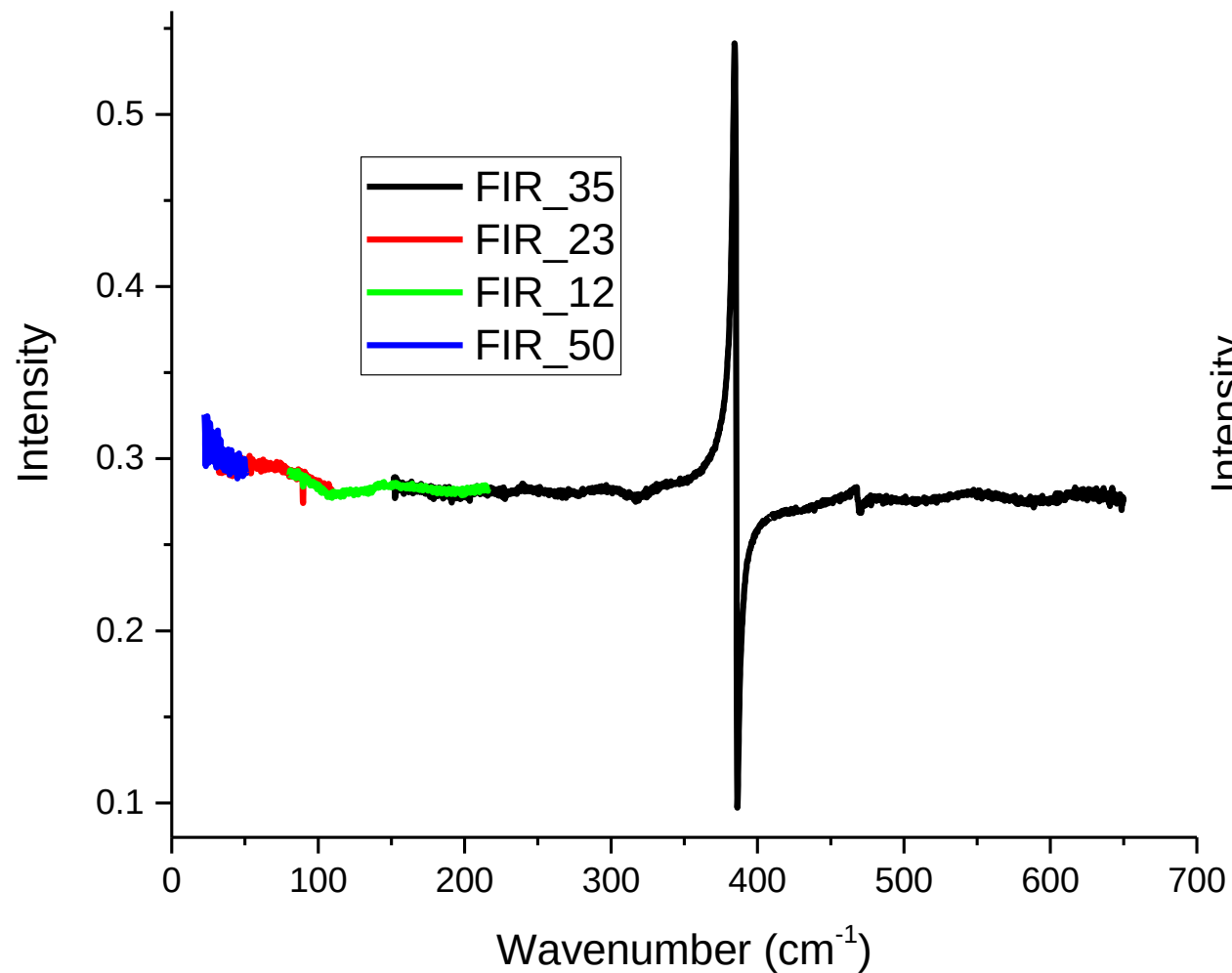
Saved the result to Table
“FIR_MERGE”

Created a non-clustered
index for the new table

```
INSERT INTO [2H].[FIR_MERGE]
SELECT [Frequency], [Intensity] FROM
[2H].[FIR_35]
UNION
SELECT [Frequency], [Intensity] FROM
[2H].[FIR_23]
UNION
SELECT [Frequency], [Intensity] FROM
[2H].[FIR_12]
UNION
SELECT [Frequency], [Intensity] FROM
[2H].[FIR_50];
GO

CREATE NONCLUSTERED INDEX
[IX_FIR_MERGE_Intensity] ON
[2H].[FIR_MERGE]([Intensity]) ON [PRIMARY];
GO
```

FIR merged result



Light scattering correction

Conducted the scattering correction using [2H].[UV] divided by [2H].[UV_AI]. The corrected results were saved to a table for the further analysis

Alter the table and set a clustered index on the Frequency column

Created a non-clustered index for the new table

```
INSERT INTO [Spectroscopy].[2H].[UV_Corrected]
SELECT [2H].[UV].Frequency,
[2H].[UV].Intensity/[2H].[UV_AI].Intensity AS
Intensity
FROM [2H].[UV], [2H].[UV_AI]
WHERE [2H].[UV].Frequency =
[2H].[UV_AI].Frequency;
```

```
ALTER TABLE [2H].[UV_Corrected] WITH CHECK ADD
    CONSTRAINT [PK_UV_Corrected_Frequency]
PRIMARY KEY CLUSTERED
(
    [Frequency]
) ON [PRIMARY];
GO
```

```
CREATE NONCLUSTERED INDEX
[IX_UV_Corrected_Intensity] ON
[2H].[UV_Corrected]([Intensity]) ON [PRIMARY];
GO
```

Merge all frequency range data

Calculated the spectrum shift using FIR data as the baseline and the result assign to the parameters

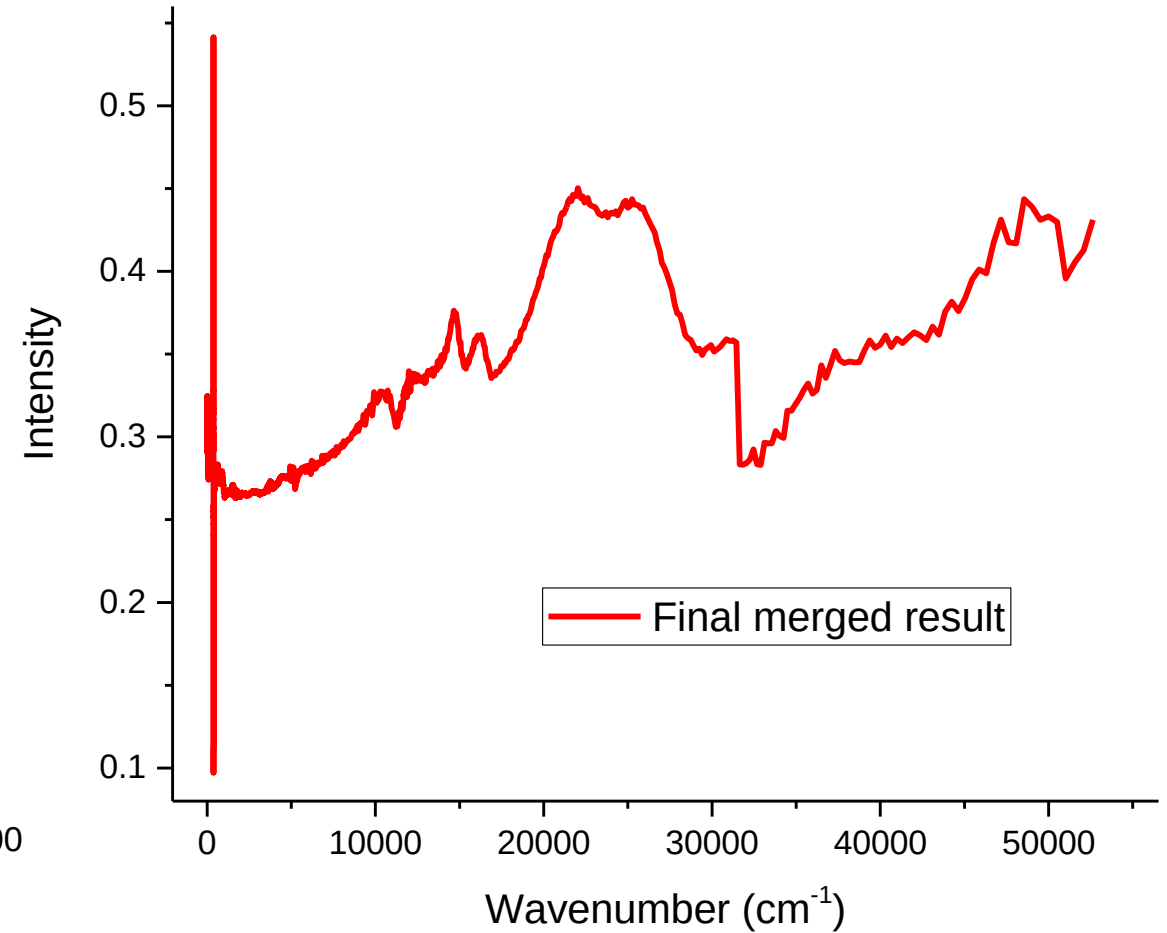
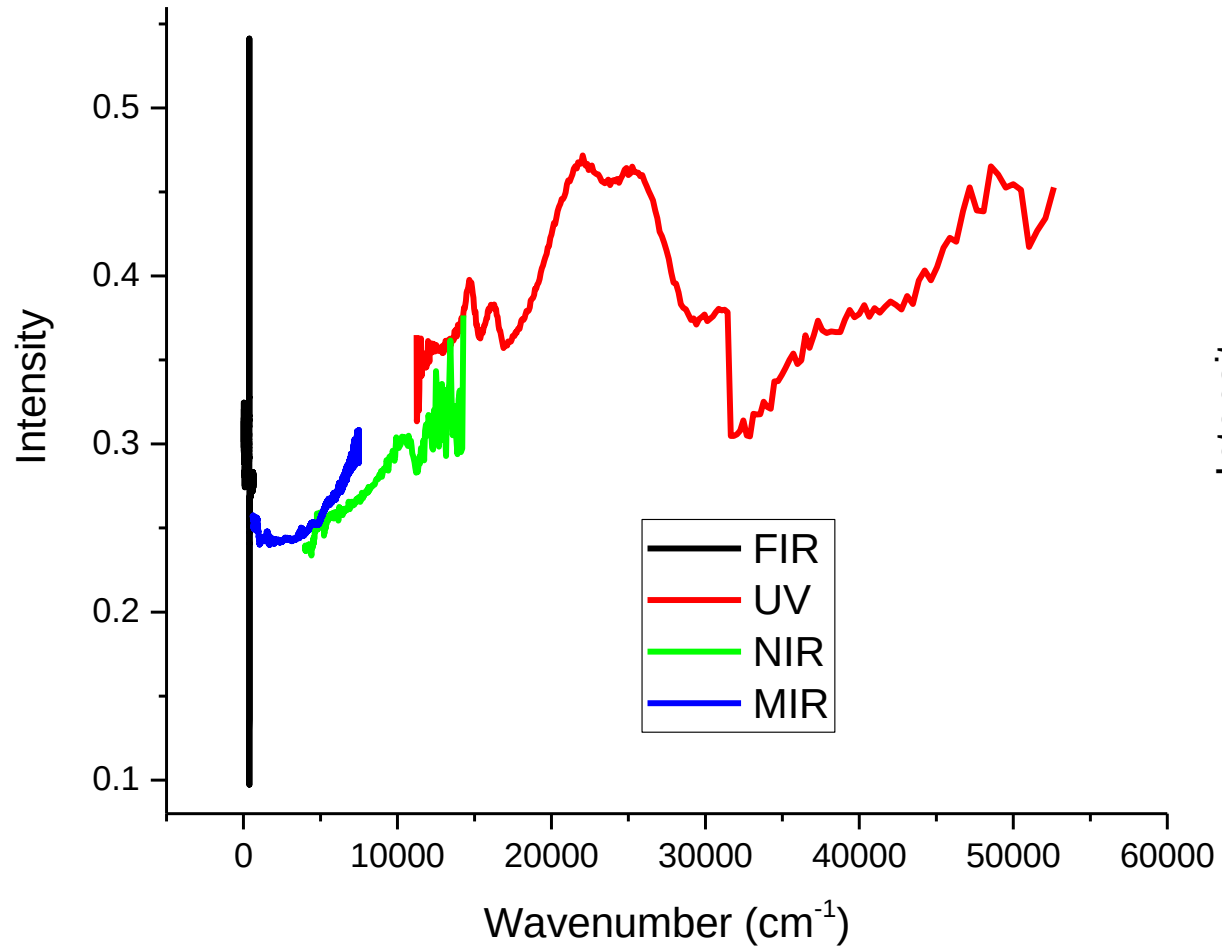
```
DECLARE @MIR float = (SELECT A.[Intensity] - B.[Intensity] AS Diff
FROM [2H].[FIR_MERGE] AS A, [2H].[MIR_Corrected] AS B
WHERE A.[Frequency] = 640.37 AND B.[Frequency] = 640.3677)
, @NIR float = (SELECT A.[Intensity] - B.[Intensity] AS Diff
FROM [2H].[MIR_Corrected] AS A, [2H].[NIR_Corrected] AS B
WHERE A.[Frequency] = 4930.05973 AND B.[Frequency] = 4930.966469)
, @UV float = (SELECT A.[Intensity] - B.[Intensity] AS Diff
FROM [2H].[NIR_Corrected] AS A, [2H].[UV_Corrected] AS B
WHERE A.[Frequency] = 11682.24299 AND B.[Frequency] = 11655.01166)
;
```

Merge the different frequency range data using “UNION ALL” command

```
INSERT INTO [2H].[MERGE]
SELECT [Frequency], [Intensity] FROM [2H].[FIR_MERGE]
UNION ALL
SELECT [Frequency], [Intensity] + @MIR AS Intensity
FROM [2H].[MIR_Corrected]
UNION ALL
SELECT [Frequency], [Intensity] + @MIR + @NIR AS Intensity
FROM [2H].[NIR_Corrected]
UNION ALL
SELECT [Frequency], [Intensity] + @MIR + @NIR + @UV AS Intensity
FROM [2H].[UV_Corrected];
GO
```

Saved the results to Table “Merge”

Final merged result



Website demos: flexible design

Qi Sun (Charles)

[Home](#)[Resume](#)[Certificates](#)[GitHub and Linkedin](#)[Experience Summary](#)[Publications](#)

INTRODUCE MYSELF

I was born in Huadian city, Jilin province, China. A beautiful city lies in the northeast of China. I got my PhD degree in Physical Chemistry from University of Tennessee in December 2012.

MY EXPERIENCE

I have a real passion for database technology and programming.

I spent most of my time to analyze the data, and found science behind them using Python and SQL Server. Because I measured the samples using different instruments which cover different frequency ranges, I need to merge the data. I also need to correct data from the scattering and convert data into different units. SQL Server can quickly finish these assignments and helped me save a lot of time on data analysis. From these processes, I practiced a lot of SQL commands, database building, maintenance, and backup. I also analyzed data using different numerical models such as Lorentz model, Gaussian, Kramers-Kronig model. I chose Python as the programming language because it is free and has good libraries. The data presentation was done by many tools such as Origin, Excel, Photoshop, and Diamond. Beautiful figures are always required by high quality journals. During my PhD study, I got well trained on data analysis and presentation.

After my PhD study at University of Tennessee, I joined University of Colorado, Boulder as a Research Associate. I worked on very different projects. At work I use Matlab as data analysis tool because it is very

NEWS

[My experience summary.](#)

[12/29/2015 - I created an online shopping](#)

Website demos: flexible design

[Home](#) [Resume](#) [Certificates](#) [GitHub and LinkedIn](#) [Experience Summary](#) [Publications](#)

CERTIFICATES



Microsoft® Certified Solutions Expert:
Business Intelligence



Microsoft® Certified Solutions
Associate: SQL Server 2012



Microsoft Certified Solutions Developer:
Web Applications



Microsoft Specialist: Programming in
HTML5 with JavaScript and CSS3



Microsoft Specialist: Programming in C#



Microsoft Technology Associate:
Database Fundamentals



Microsoft Certified Professional

Qi Sun (Charles)

[Home](#) [Resume](#) [Certificates](#) [GitHub and LinkedIn](#) [Experience Summary](#) [Publications](#)

- [Summary Database&Programming.pdf](#)
- [Hotel booking dystem with database](#)
- [Online shopping system with database.](#)
- [DVD rental database using C# with two windows forms as inputs.](#)
- [Machine Learning Python codes.](#)

[About](#) [Useful Links](#) [Help](#) [Contact](#)

Website demos: ASP.NET code demos

MyWebSite (Running) - Microsoft Visual Studio

File Edit View Website Build Debug Team Tools Test Analyze Window Help

Process: [9256] iisexpress.exe Lifecycle Events Thread: Stack Frame: DOCTYPE: HTML5 No Application Insights Events

Toolbox Search Toolbox General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Default.aspx experience.aspx otherWeb.aspx about.aspx stylesheet.css links.aspx help.aspx contact.aspx

```
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
  <div class="sideright">
    
  </div>
  <div class="sideright">
    <h2>News</h2>
    <p> 6/3/2015 I opened a Github blog. </p>
    <p> 7/8/2015 I rewrote Machine Learning codes using Python.</p>
    <p> 9/2/2015 - I got my MCSA certificate!</p>
    <p> 9/4/2015 - I got my MS: C# certificate! </p>
    <p> 12/1/2015 - I got my MCSE certificate! </p>
    <p> 12/7/2015 - I got my MS: HTML5 with CSS and JavaScript certificate! </p>
    <p> 12/15/2015 - More excited news on the way!!! </p>
  </div>
  <div class="main">
    <h1> INTRODUCE MYSELF </h1>
    <p> I was born in Huadian city, Jilin province, China. A beautiful city lies in the northeast of China.</p>
    <p> I got my PhD degree in Physical Chemistry from University of Tennessee in 2012.</p>
  </div>
  <div class="main">
    <h1> MY EXPERIENCE </h1>
    <p> I have a real passion for database technology and programming.</p>
    <p> I spent most of my time to analyze the data, and found science behind them using Python and SQL Server. Because I measured the samples using different instruments which cover different frequency ranges, I need to merge the data. I also need to correct data from the scattering and convert data into different units. SQL Server can quickly finish these assignments and helped me save a lot of time on data analysis. From these processes, I practiced a lot of SQL commands, database building, maintenance, and backup. I also analyzed data using different statistical models such as Logistic model, Gaussian, K-Nearest Neighbors model, etc.</p>
  </div>
</asp:Content>
```

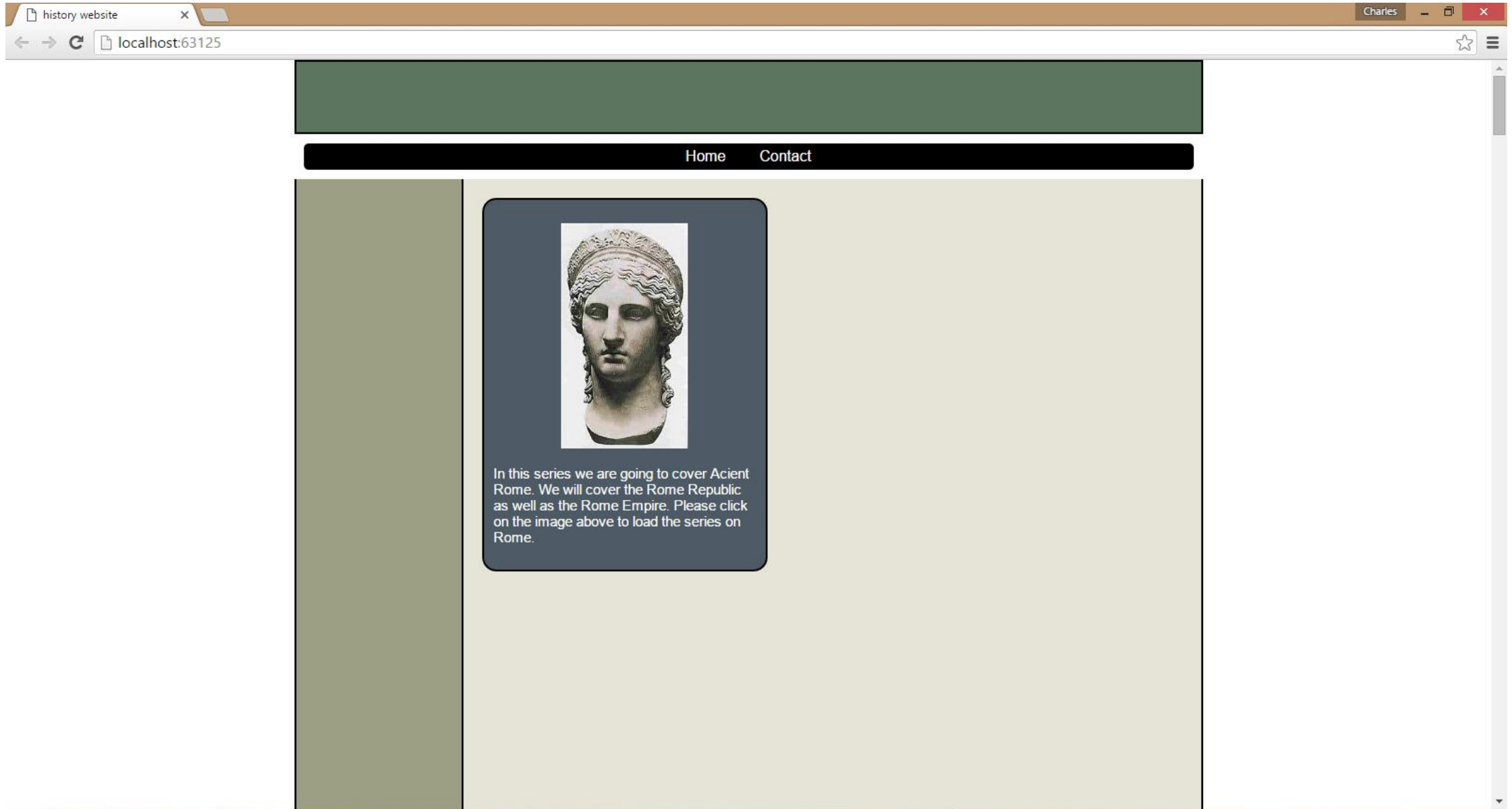
Solution Explorer Search Solution Explorer (Ctrl+;) Solution 'MyWebSite' (1 project) MyWebSite Archive Images PDF files Summary_Database&Programming.pdf Webpages Certificates certificates.aspx Experience experience.aspx Misc about.aspx contact.aspx help.aspx links.aspx OtherWeb otherWeb.aspx Publications publications.aspx Resume resume.aspx Default.aspx MasterPage.master stylesheet.css Web.config

Design Split Source <asp:Content#Content2> <div.sideright> <p>

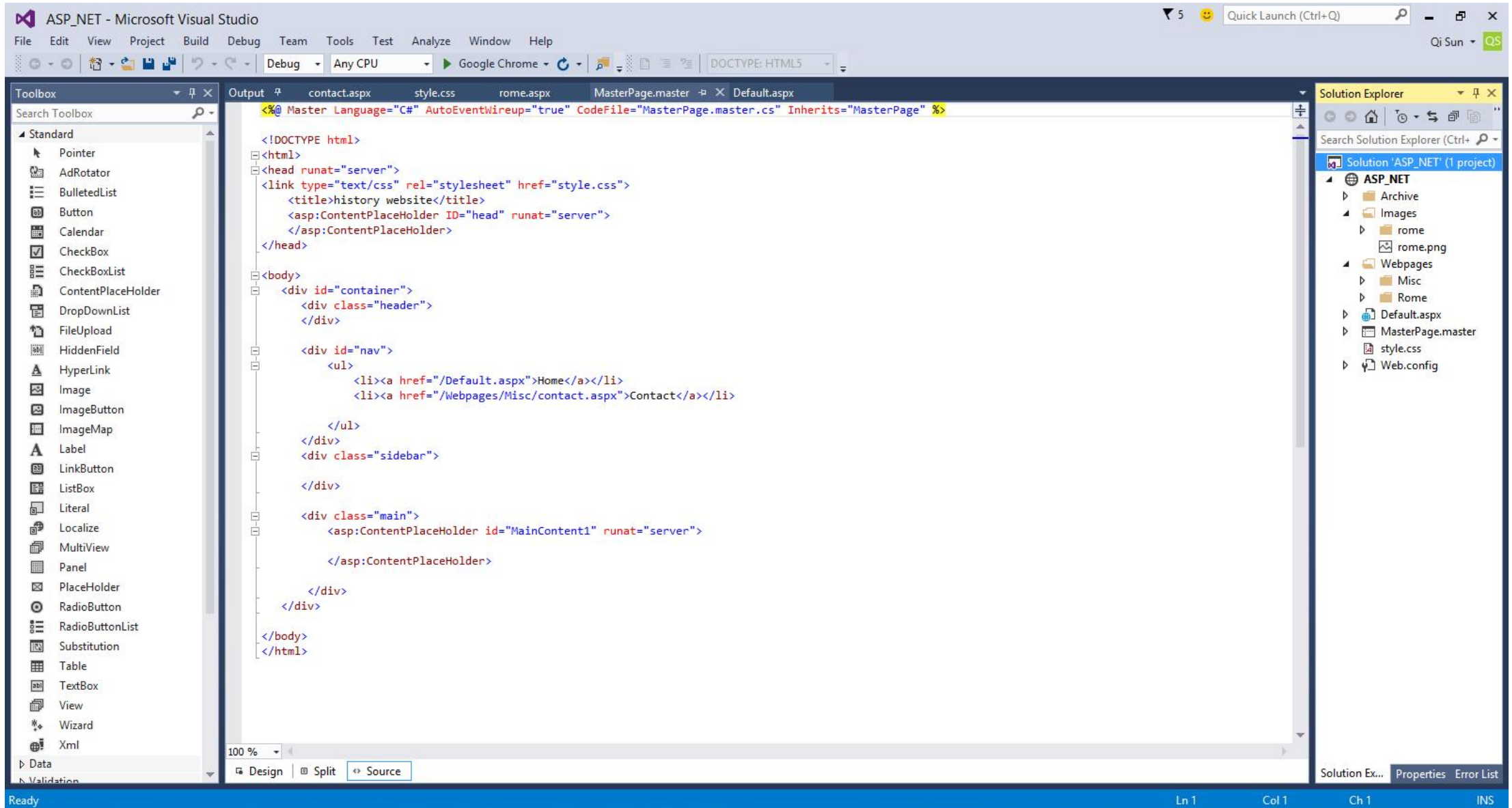
Call Stack Immediate Window Command Window Exception Settings Breakpoints Autos Locals Watch 1

Ready Ln 10 Col 22 Ch 22 INS

Website demos: fixed design



Website demos: ASP.NET code demos



Website demos: online shopping

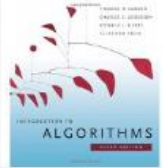
Good Shop Category Search Sign in Your cart: 0 item(s), \$0.00 Cart



Algorithms
Good book Weight: 3 pounds

Quantity: 8 \$63.08

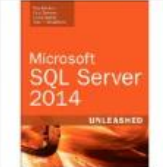
Add to Cart



Introduction to Algorithms
Good book Weight: 3 pounds

Quantity: 5 \$58.08

Add to Cart



Microsoft SQL Server 2014 Unleashed
Good book Weight: 2 pounds

Quantity: 12 \$43.88

Add to Cart



Training Kit (Exam 70-461) Querying Microsoft SQL Server 2012
Good book Weight: 3 pounds

Quantity: 0 \$44.34

Add to Cart

Website demos: online shopping

Good Shop Category Search Hello eagle Your cart: 2 item(s), \$79.98 Cart

**Nikon Coolpix L110 Digital Camera, Black**
Good camera Weight: 3 pounds
Quantity: 8 \$133.00
Add to Cart

**Canon EOS Rebel T5i Digital SLR Camera (Body Only)**
Good camera Weight: 4 pounds
Quantity: 0 \$649.00
Add to Cart

**Canon PowerShot SD1000 7.1MP Digital Elph Camera with 3x Optical Zoom (Silver)**
Good camera Weight: 1 pounds
Quantity: 2 \$39.99
Add to Cart

About Contact

Good Shop Category Search Hello eagle Your cart: 3 item(s), \$184.24 Cart

Your cart

Quantity	Item	Price	Subtotal	
2	Algorithms	\$63.08	\$126.16	Remove
1	Introduction to Algorithms	\$58.08	\$58.08	Remove
		Total:	\$184.24	

Continue shopping Checkout now

Website demos: online shopping

Good Shop Category [Cart](#) Your cart: 0 item(s), \$0.00



Microsoft SQL Server 2014 Unleashed
Good book Wieght: 2 pounds

Quantity: 11 \$43.88



Training Kit (Exam 70-461) Querying Microsoft SQL Server 2012
Good book Wieght: 3 pounds

Quantity: 3 \$44.34



Training Kit (Exam 70-462) Administering Microsoft SQL Server 2012 Databases
Good book Wieght: 3 pounds

Quantity: 3 \$47.28

1

Website demos: online shopping

Good Shop Category Search Sign in **Your cart:** 0 item(s), \$0.00 **Cart**

Register
Account

User name
User name is required.

Email
Email is required.

Email again
Please enter email again.

Password
Password is required.

Password again
Please enter password again.

Create

[Back to Sign in page](#)

User name
User name is already in use.

Email

Email again
Please confirm your email.

Password

Password again
Please confirm your password.

Create

Good Shop Category Search **Hello eagle** **Your cart:** 0 item(s), \$0.00 **Cart**

Welcome eagle!!!

Continue Shopping Check Out

About Contact

Website demos: online shopping

Good Shop Category

Check Out Now

Please enter your details, and we'll ship your goods right away!

Shipping Details

Name
Please enter a Name.

Line 1
Please enter the first Address Line.

Line 2

Line 3

City
Please enter a City name.

State
Please enter a State name.

Zip Code

State
Please enter a State name.

Zip Code
Please enter a Zip Code.

Country
Please enter a Country name.

Phone Number
Please enter Phone Number.

Credit Card Information

Credit Card Number
Please enter Credit Card Number.

Credit Card Type Please enter Credit Card Type.

Expiration Month Please enter Expiration Month.

Expiration Year Please enter Expiration Year.

Security Code
Please enter Security Code.

GiftWrap ☐

Good Shop Category Hello eagle

Cart

Orders Submitted

Your cart: 0 item(s), \$0.00

Website demos: online shopping

Good Shop

Create

Product

Name

Quantity

0

Price

0.00

Color

ProductWeight

0

ProductWeightUnit

Category

Description

Create

Good Shop

charleseagleLogOut

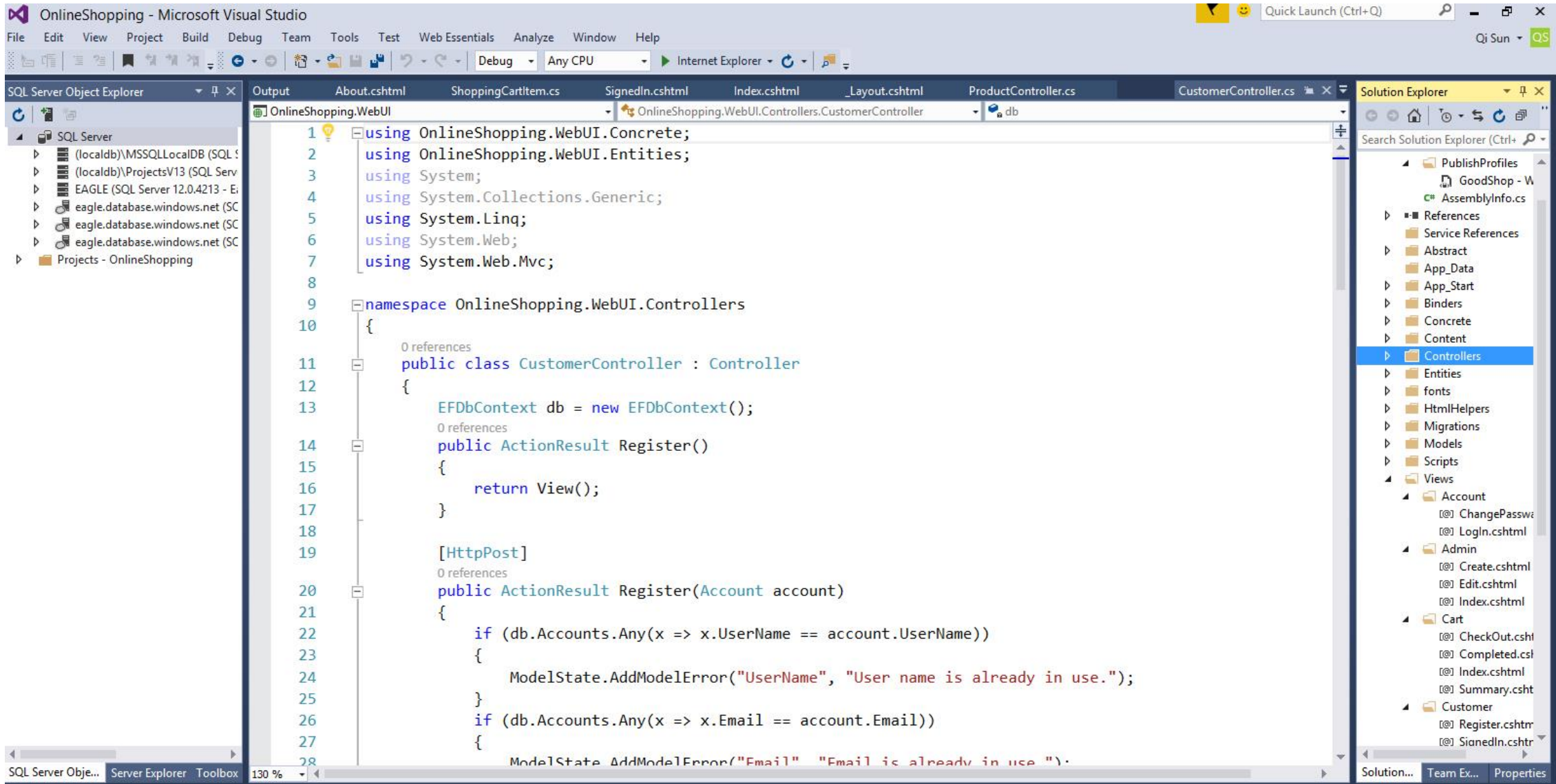
All Products

ID	Name	Price	Actions
1	Algorithms	\$63.08	Delete
2	Introduction to Algorithms	\$58.08	Delete
3	Microsoft SQL Server 2014 Unleashed	\$43.88	Delete
4	Training Kit (Exam 70-461) Querying Microsoft SQL Server 2012	\$44.34	Delete
5	Training Kit (Exam 70-462) Administering Microsoft SQL Server 2012 Databases	\$47.28	Delete
6	Training Kit (Exam 70-463) Implementing a Data Warehouse	\$45.08	Delete
7	Expert ASP.NET Web API 2 for MVC Developers	\$43.08	Delete
8	ASP.NET MVC 5 with Bootstrap and Knockout.js	\$23.08	Delete

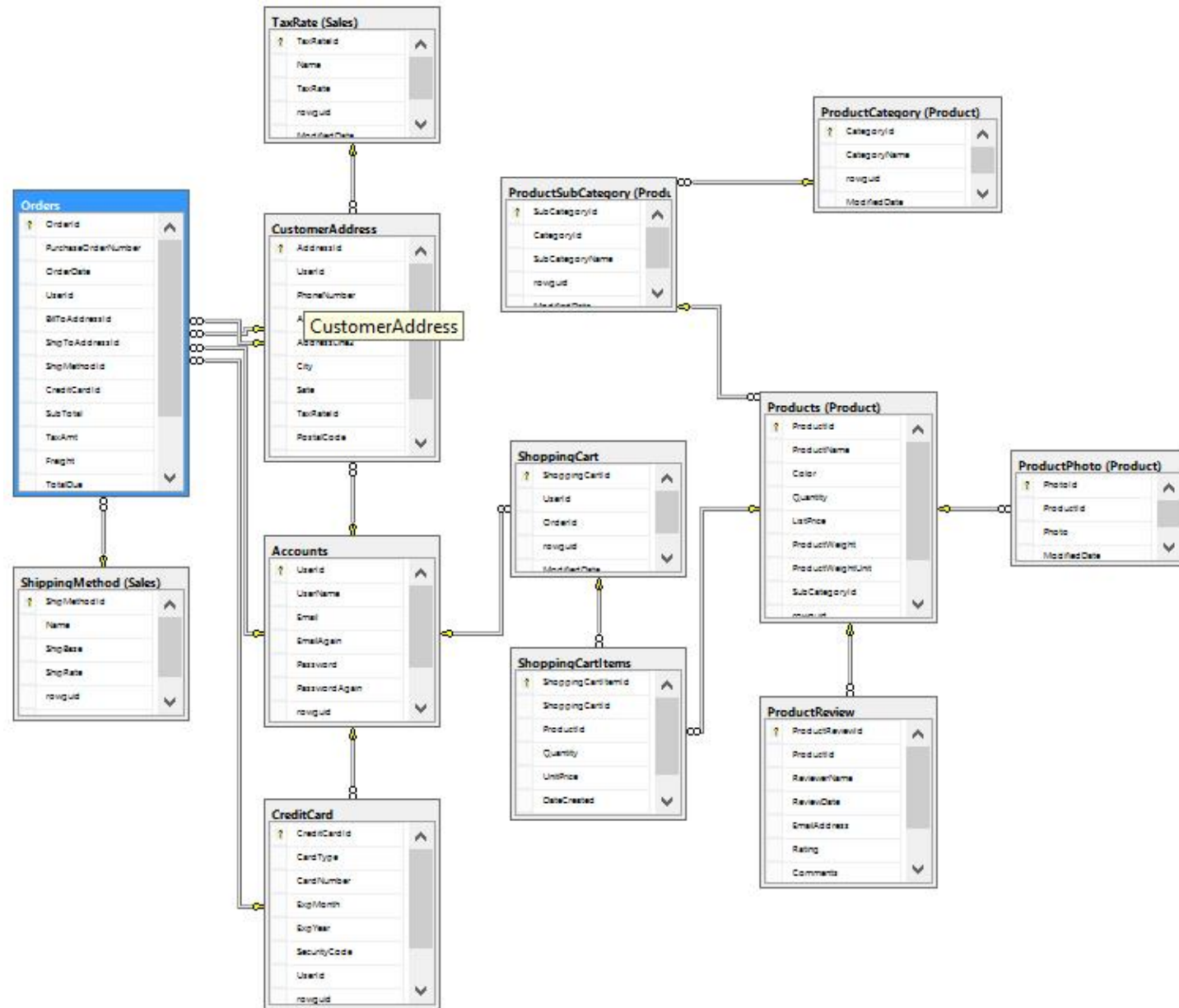
37	JBL J33a BLK Premium In Ear Headphones with JBL Drivers and Microphone, Black	\$22.05	Delete
38	Yurbuds (CE) Inspire 300 Noise Isolating In-Ear Headphones, Mossy Oak Orange	\$23.00	Delete
39	Panasonic RP-HJE125E-V Headphone	\$10.00	Delete

Add a New Product

Website demos: online shopping



Website demos: online shopping



Website demos: online hotel booking

Good Hotel Demo

ReservationsOffersMembersContact

Destination:

Please select a city

Check In:

Check Out:

Adults:

Room:

Please fill out this field.

February 2016

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2		1	2	3	4	5	6
3	4	5	6	7	8	9	7	8	9	10	11	12	13
10	11	12	13	14	15	16	14	15	16	17	18	19	20
17	18	19	20	21	22	23	21	22	23	24	25	26	27
24	25	26	27	28	29	30	28	29					
31													

Website demos: online hotel booking

Good Hotel Demo

ReservationsOffersMembersContact

Final Price: \$ 80

Membership Number:

Contact information

First Name:
First Name is Required

Address:
Address is required

Postal Code:
Postal Code is required

Email:
Email is required

Last Name:
Last Name is required

City/Town:
City/Town is required

Country:
Please select a Country

Phone Number:
Phone Number is required

Payment Information

Payment Card Type:
Please select Payment Card Type

Expiration Date:
Please select a Month

Payment Card Number:
Payment Card Number is required

Please select a Year

Terms and Conditions

You must agree to the terms and conditions

☒ I certify that:

- I am at least 18 years of age and that at least one guest in my party will be at least 21 years of age upon check in.
- I have read and understand the rate description and rate rules for my reservation.

Website demos: online hotel booking

Good Hotel Demo

[Reservations](#)[Offers](#)[Members](#)[Contact](#)

Congratulations Qi Sun !

Your order processd sucessfully.

Enjoy your upcoming trip!

PLease remember our order number: 25 !!!

Hotel Name: Mart Plaza River North

Room
Number: 1001

Check In: 12/30/2015 12:00:00 AM

Check Out 12/31/2015 12:00:00 AM

Final Price: \$ 100

We also sent you a confirmation email to sunqicharles@gmail.com

Please check your emamil box.

Website demos: online hotel booking

Good Hotel Demo

[Reservations](#)[Offers](#)[Members](#)[Contact](#)

Your Confirmation Number:

Frist Name:

Qi

Last Name:

Sun

Hotel Name:

Mart Plaza River North

Room Number:

1001

Check In

12/30/2015 12:00:00 AM

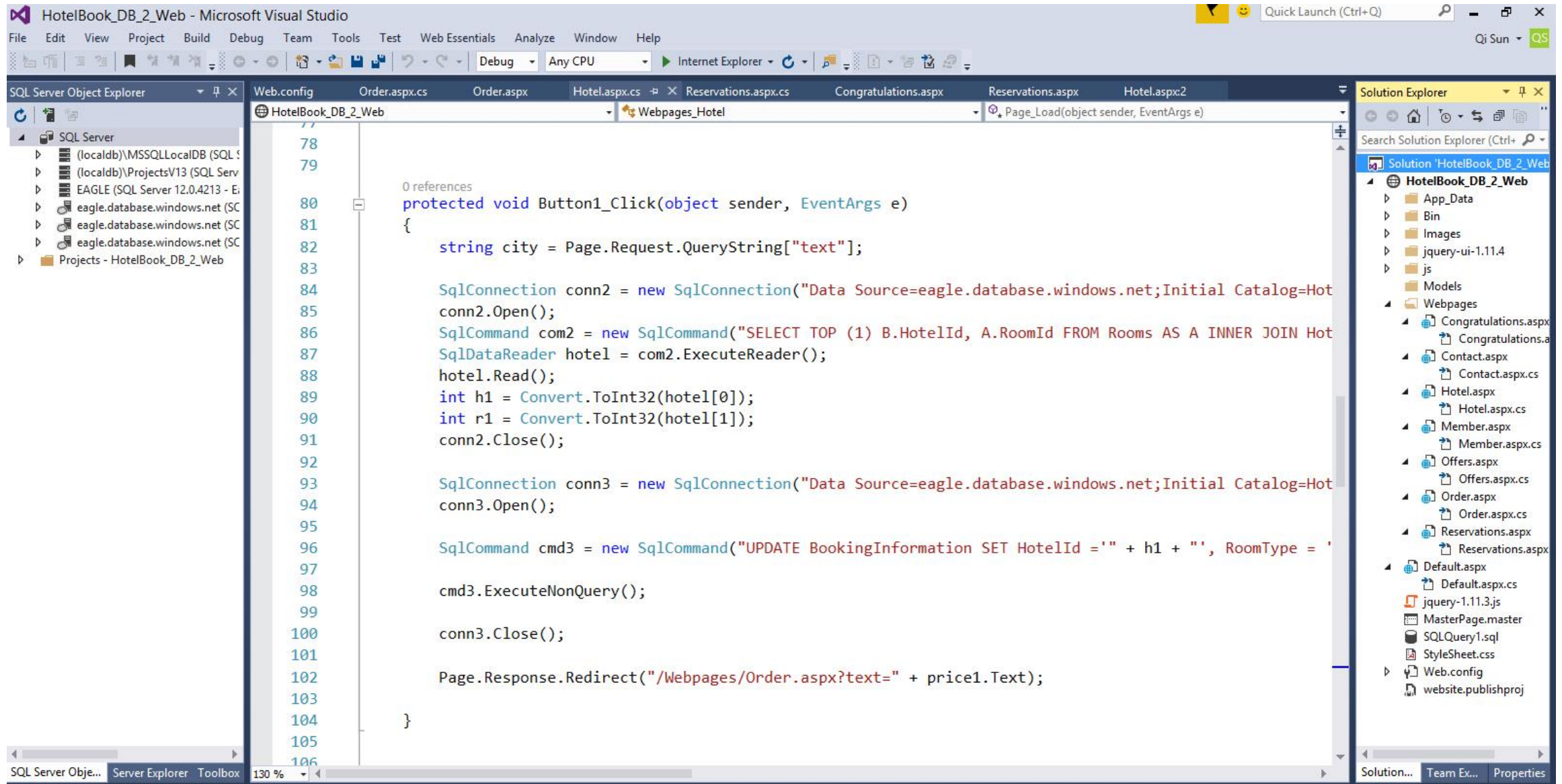
Check Out

12/31/2015 12:00:00 AM

Final Price:

\$ 100

Website demos: online hotel booking



The screenshot shows the Microsoft Visual Studio IDE with the following components:

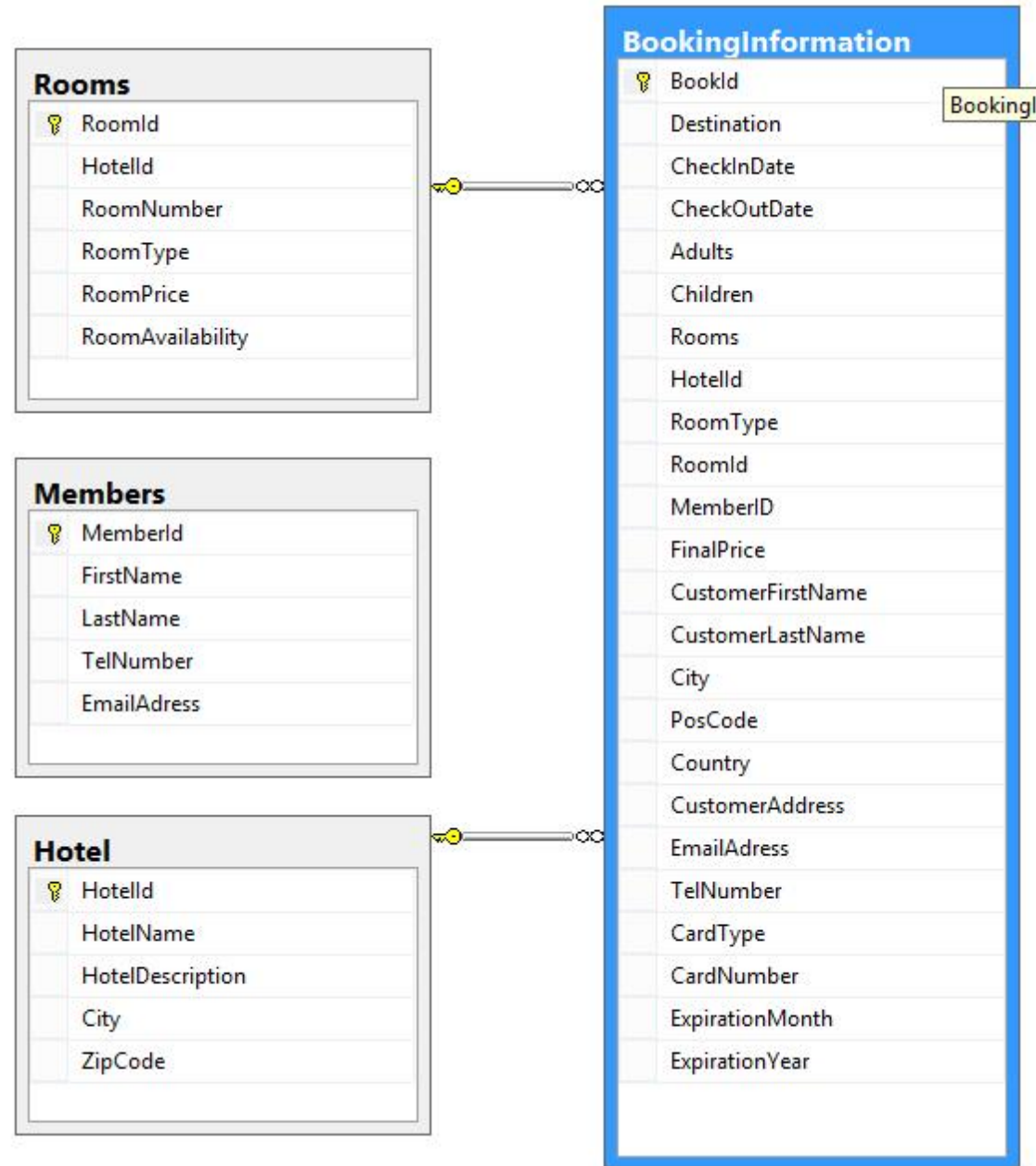
- SQL Server Object Explorer:** Displays the local database structure, including the 'HotelBook_DB_2_Web' project and its associated databases.
- Web.config:** The active file in the editor, showing the configuration for the application.
- Order.aspx.cs:** The code file for the Order.aspx page, showing the Page_Load method.
- Reservations.aspx.cs:** The code file for the Reservations.aspx page, showing the Page_Load method.
- Hotel.aspx2:** The code file for the Hotel.aspx2 page, showing the Page_Load method.
- Webpages_Hotel:** The code file for the Webpages_Hotel page, showing the Page_Load method.
- Page_Load(object sender, EventArgs e):** The method being edited, which contains the following code:

```
78  
79  
80 0 references  
81 protected void Button1_Click(object sender, EventArgs e)  
82 {  
83     string city = Page.Request.QueryString["text"];  
84  
85     SqlConnection conn2 = new SqlConnection("Data Source=eagle.database.windows.net;Initial Catalog=Hot  
86     conn2.Open();  
87     SqlCommand com2 = new SqlCommand("SELECT TOP (1) B.HotelId, A.RoomId FROM Rooms AS A INNER JOIN Hot  
88     SqlDataReader hotel = com2.ExecuteReader();  
89     hotel.Read();  
90     int h1 = Convert.ToInt32(hotel[0]);  
91     int r1 = Convert.ToInt32(hotel[1]);  
92     conn2.Close();  
93  
94     SqlConnection conn3 = new SqlConnection("Data Source=eagle.database.windows.net;Initial Catalog=Hot  
95     conn3.Open();  
96  
97     SqlCommand cmd3 = new SqlCommand("UPDATE BookingInformation SET HotelId='" + h1 + "', RoomType = '  
98     cmd3.ExecuteNonQuery();  
99  
100    conn3.Close();  
101  
102    Page.Response.Redirect("/Webpages/Order.aspx?text=" + price1.Text);  
103  
104 }  
105  
106
```

The Solution Explorer on the right shows the project structure, including the following files and folders:

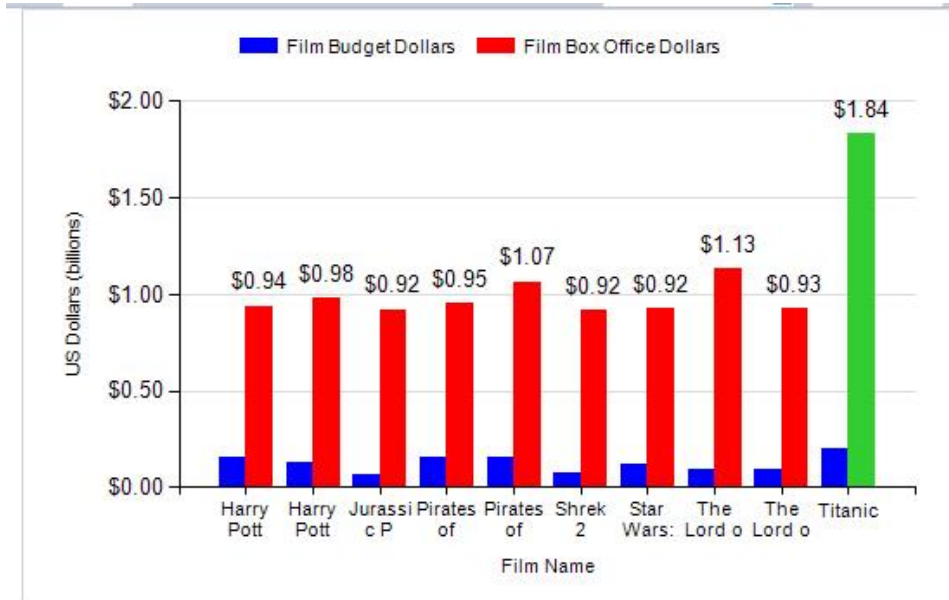
- App_Data
- Bin
- Images
- jquery-ui-1.11.4
- js
- Models
- Webpages
 - Congratulations.aspx
 - Congratulations.aspx.cs
 - Contact.aspx
 - Contact.aspx.cs
 - Hotel.aspx
 - Hotel.aspx.cs
 - Member.aspx
 - Member.aspx.cs
 - Offers.aspx
 - Offers.aspx.cs
 - Order.aspx
 - Order.aspx.cs
 - Reservations.aspx
 - Reservations.aspx.cs
 - Default.aspx
 - Default.aspx.cs
 - jquery-1.11.3.js
 - MasterPage.master
 - SQLQuery1.sql
 - StyleSheet.css
- Web.config
- website.publishproj

Website demos: online hotel booking



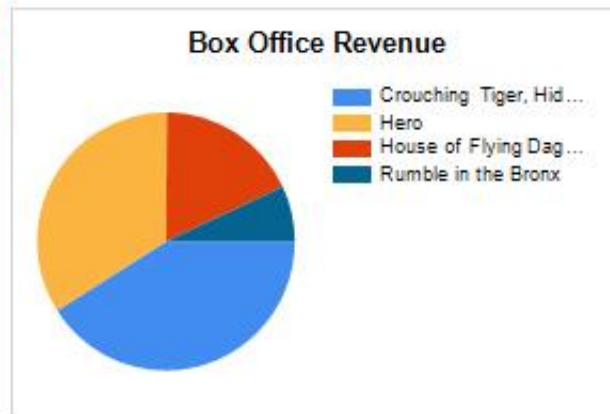
SSRS demos

Chart



List

China	
Film Name	Film Release Date
Hero	09/24/2004
Crouching Tiger, Hidden Dragon	01/05/2001
House of Flying Daggers	01/14/2005
Rumble in the Bronx	07/04/1997



Gauge

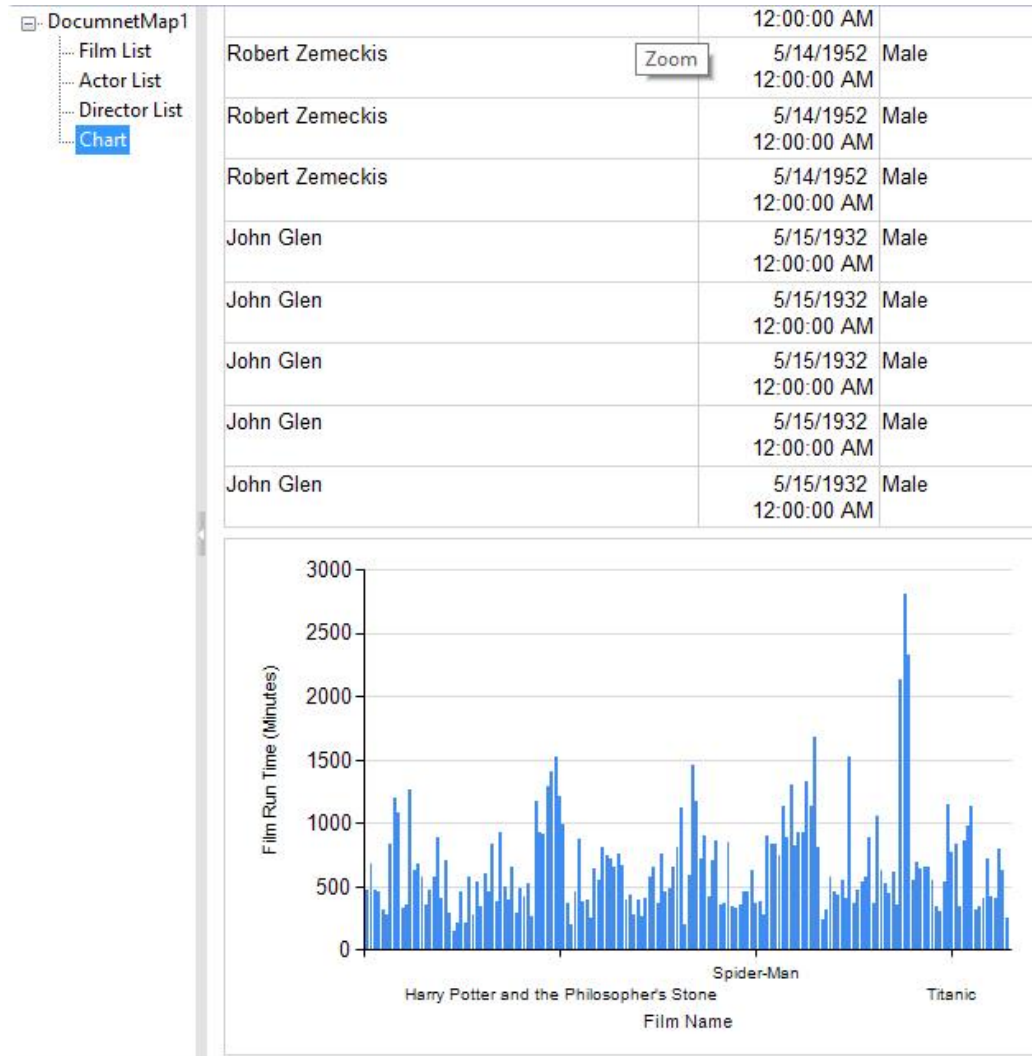
Film Name	Nomination	Wins	Indicator
Jurassic Park	3	3	
Spider-Man	2	0	
King Kong	4	3	
Superman Returns	1	0	

<https://github.com/charleseagle/SQL-Server-SSRS>

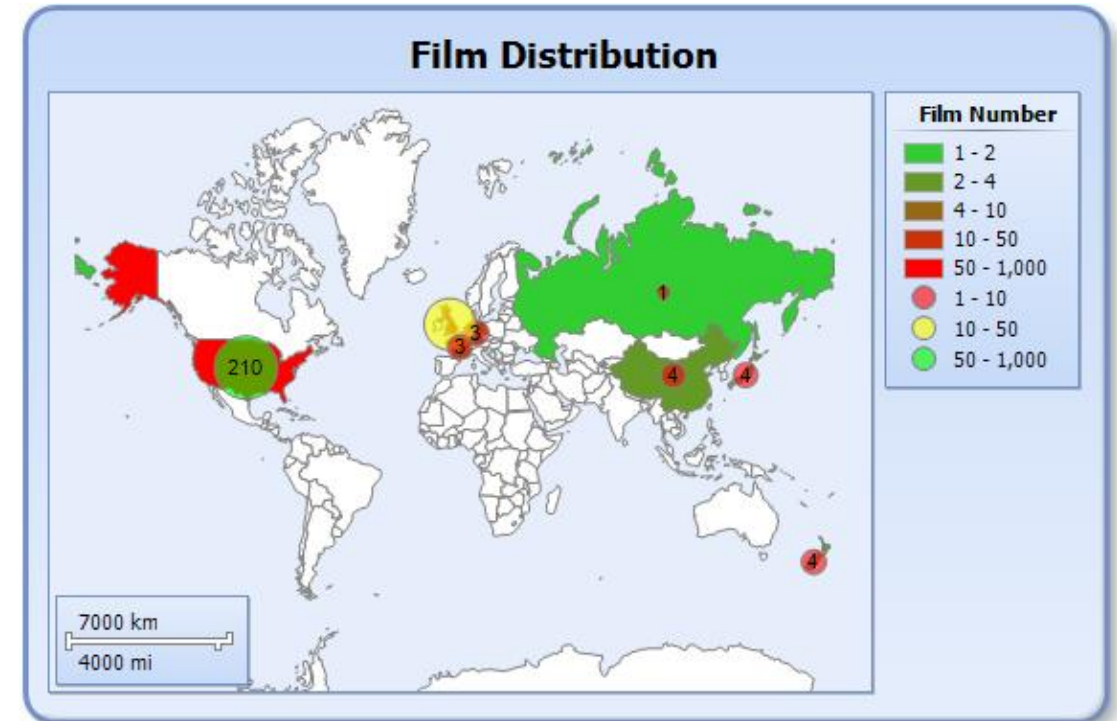
SSRS demos

Map

Document Map



<https://github.com/charleseagle/SQL-Server-SSRS>



Dropdown Parameters

Choose a day from the list Tuesday Select a month March

1 of 1 100%

Film Name	Film Release Date	Month	Day
Star Wars: Episode IV - A New Hope	12/27/1977	December	Tuesday
The Living Daylights	06/30/1987	June	Tuesday
You Only Live Twice	06/13/1967	June	Tuesday

SSRS demos

Matrix

Studio Name	Film certificate								
	12		12A		15		18		PG
	Total	Average	Total	Average	Total	Average	Total	Average	Total
Warner Bros. Pictures	376	125.33	988	141.14	1430	130.00	151	151.00	
Walt Disney Pictures	143	143.00	318	159.00					
Universal Pictures	271	135.50	774	129.00	355	118.33	335	167.50	
Touchstone Pictures	549	137.25	478	119.50	132	132.00	139	139.00	
Revolution Studios			122	122.00	124	124.00			
Paramount Pictures	485	121.25	471	117.75	522	130.50	96	96.00	
New Line Cinema	90	90.00	471	157.00	119	119.00	357	119.00	
Morgan Creek Productions	112	112.00							
Miramax Films							414	138.00	
MGM	377	125.67	277	138.50	134	134.00			
Lucasfilm			140	140.00					
Jerry Bruckheimer Films					396	132.00			
Imagine Entertainment			100	100.00					
Gaumont									
Dreamworks	288	144.00	144	144.00	738	147.60			
Disney Pixar									
Columbia Pictures			872	124.57	287	143.50			
Carolco Pictures					250	125.00	113	113.00	
Avalon Studios							99	99.00	
20th Century Fox	443	147.67	510	127.50	1036	129.50	399	133.00	
Total	3134	130.58	5665	131.74	5523	131.50	2103	131.44	

Pivot table

Type in part of a film name: King
Choose a start date: 1/1/1900
Highlight films with this many Oscars: 3
Show Films that are at least this long: 100
Choose an end date: 11/12/2015
NULL
View Report

Film Name	Release Date	Run Time Minutes	Oscar Wins
King Kong	12/17/1976	134	0
King Kong	04/07/1933	100	0
King Kong	12/15/2005	187	3
Kingdom of Heaven	05/06/2005	145	0
The Lord of the Rings: Return of the King	12/17/2003	201	11

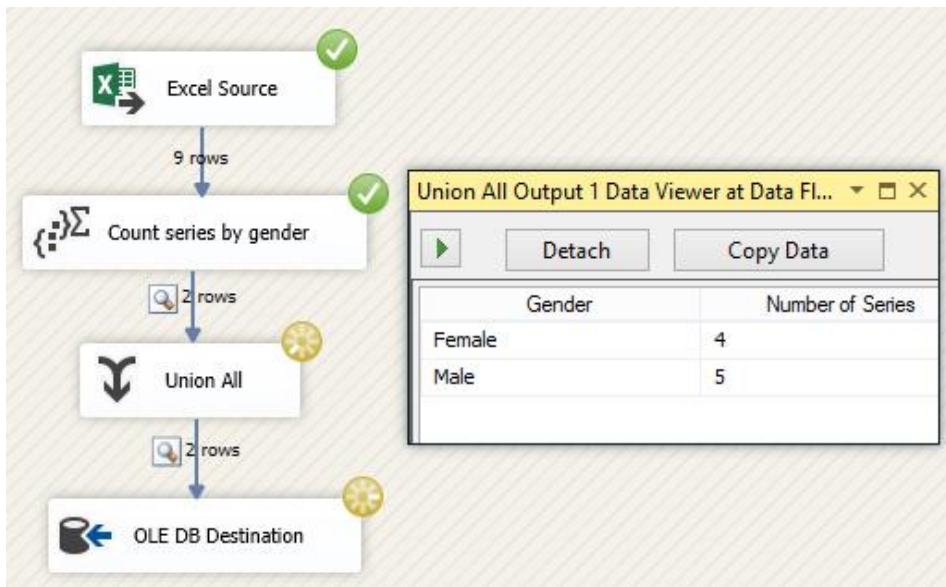
Parameters

Row Field: Country
Data Field: Box Office
Column Field: Certificate
Function: Average

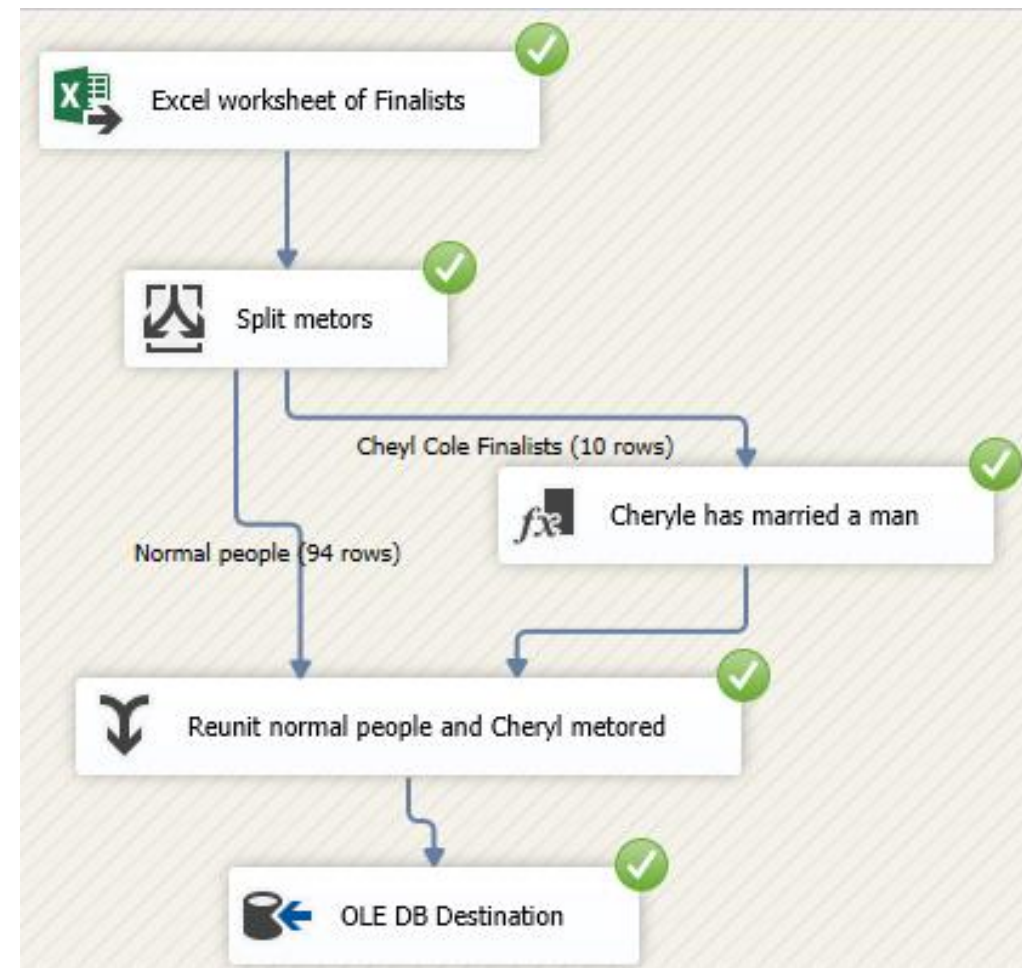
Average Box Office						
Country	PG	12A	12	U	15	18
France	263,900,000.00					
New Zealand	868,621,686.00	1,029,655,851.00				
United Kingdom	883,661,574.33	654,853,875.00	352,554,959.00		31,959,646.00	
United States	342,690,890.09	418,570,414.14	429,714,973.85	585,795,658.50	286,897,077.71	165,775,906.31

SSIS demos

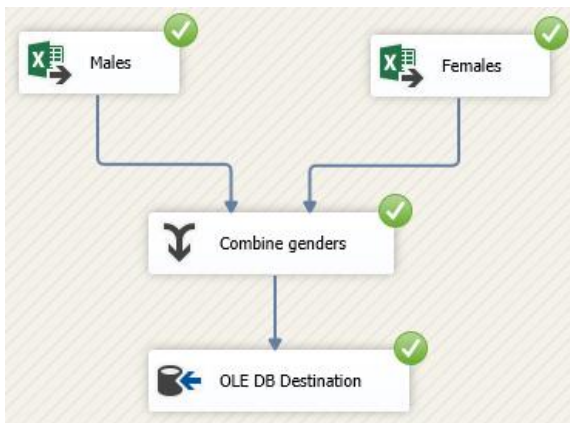
Count series by gender



Conditional split



Combine genders



Expression



Error Handling

Data Conversion Error Output Data Viewer at Import finalists

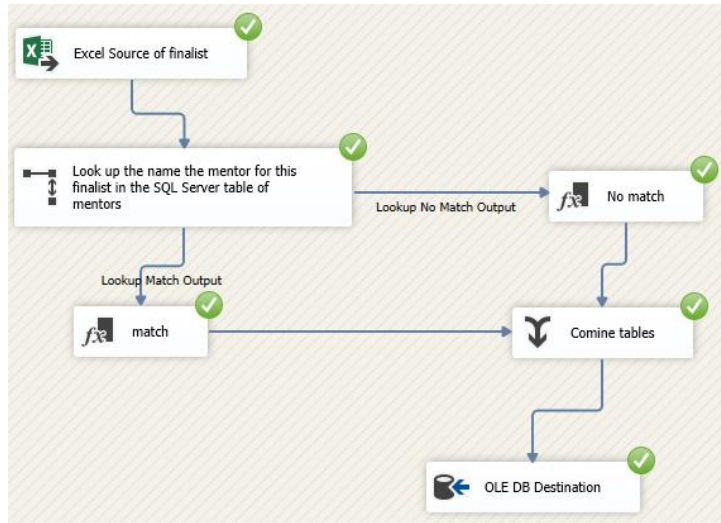
Mentor	Fini...	Ero...	Ero...	ErrorCode - Description
Simon Cowell	1	-1...	6	The data was truncated.
Louis Walsh	1	-1...	6	The data was truncated.
Simon Cowell	1	-1...	6	The data was truncated.
Tulisa Contostavlos	1	-1...	6	The data was truncated.
Louis Walsh	2	-1...	6	The data was truncated.
Sharon Osbourne	2	-1...	6	The data was truncated.
Simon Cowell	2	-1...	6	The data was truncated.
Louis Walsh	2	-1...	6	The data was truncated.
Simon Cowell	2	-1...	6	The data was truncated.
Sharon Osbourne	3	-1...	6	The data was truncated.
Simon Cowell	3	-1...	6	The data was truncated.
Sharon Osbourne	3	-1...	6	The data was truncated.
Simon Cowell	3	-1...	6	The data was truncated.
Simon Cowell	3	-1...	6	The data was truncated.
Gary Barlow	3	-1...	6	The data was truncated.

* LineageIDs that appear in the grid may not be up to date.

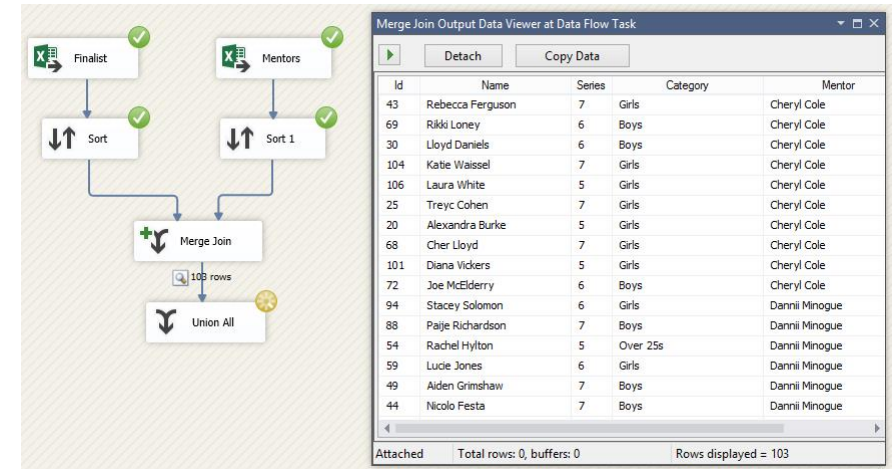
Attached Total rows: 0, buffers: 0 Rows displayed = 72

SSIS demos

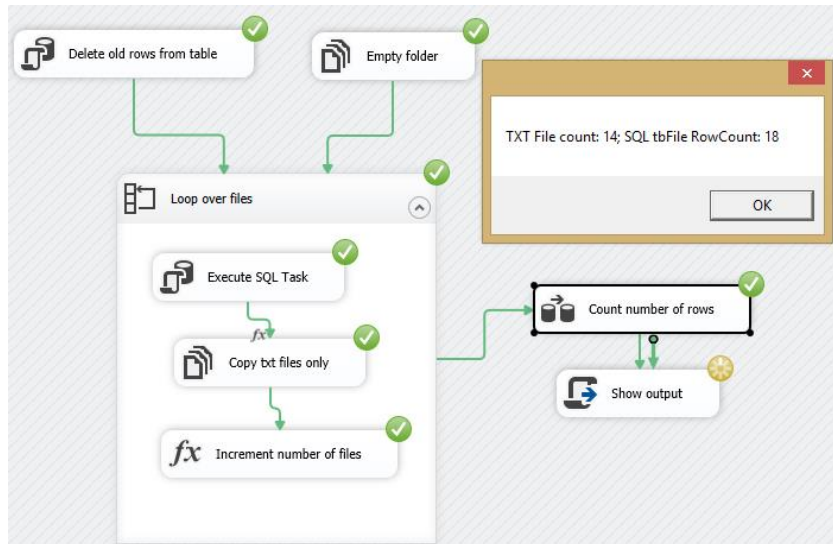
Lookup Transform



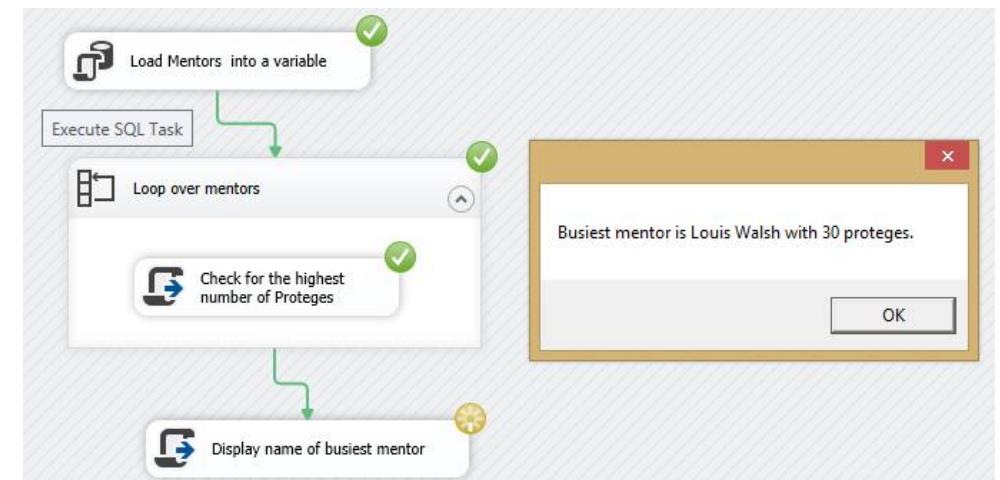
Merge join



Looping over files

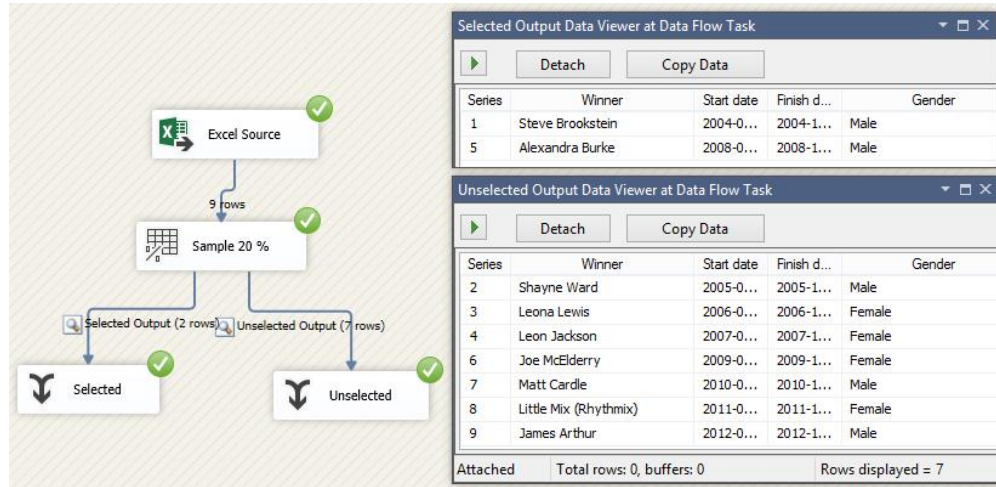


Foreach loop

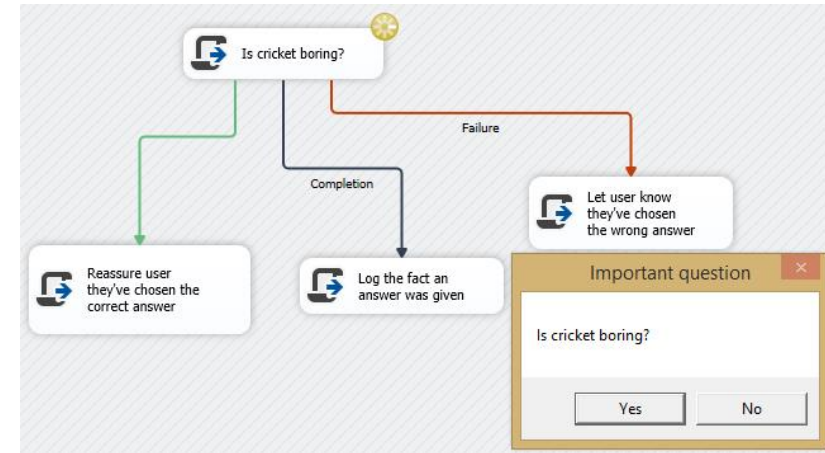


SSIS demos

Percentage sampling



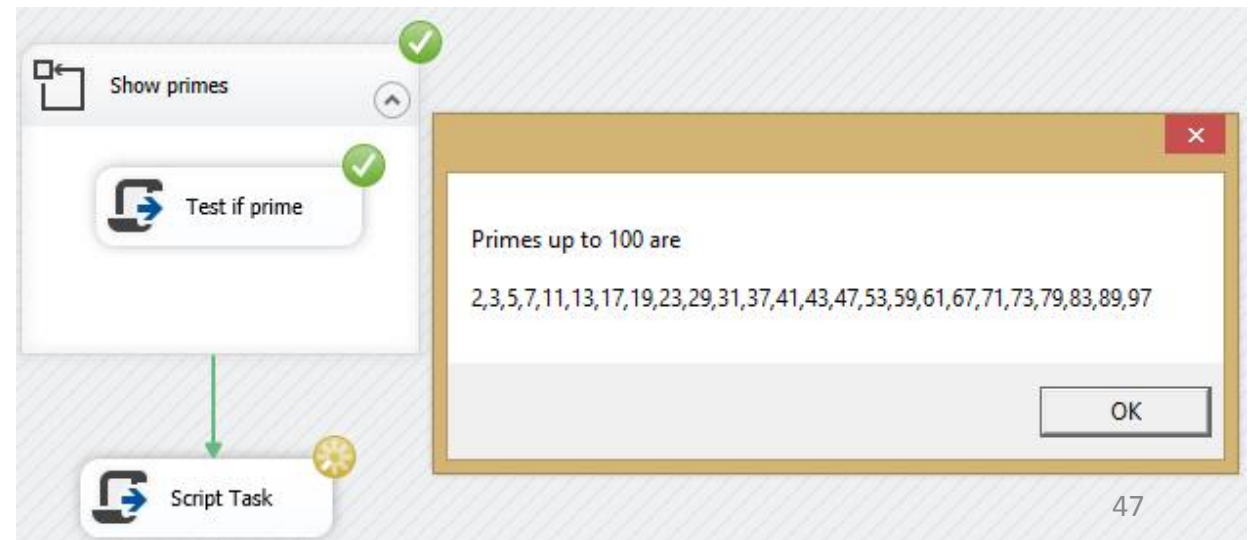
Multiple choice



Parameters deployment

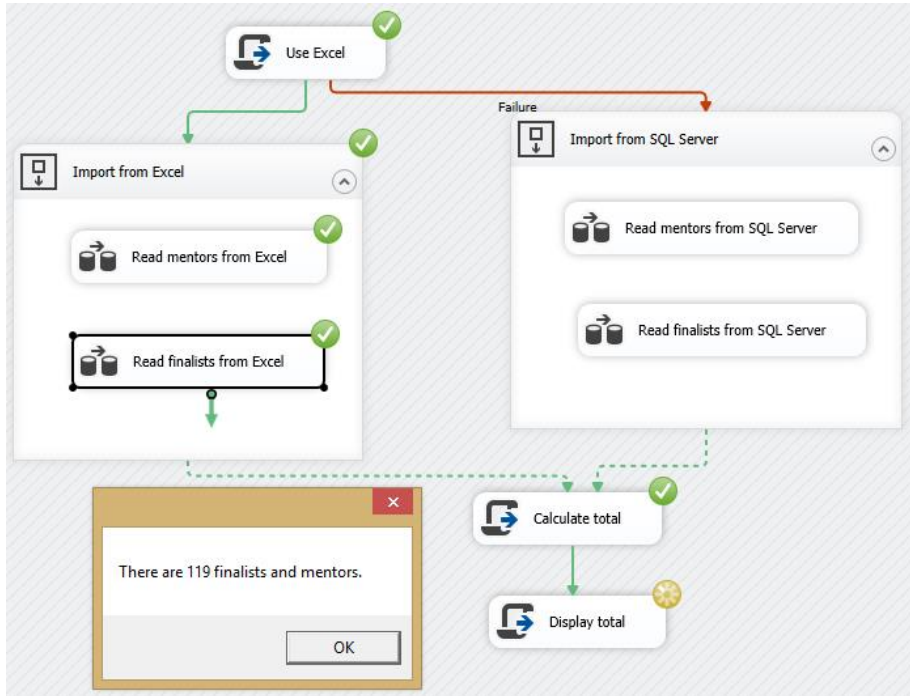


For loop

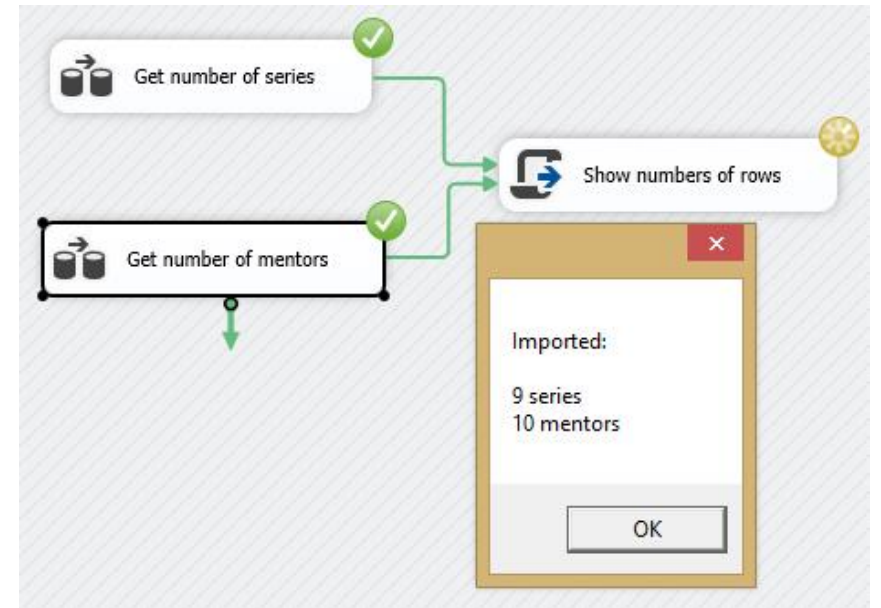


SSIS demos

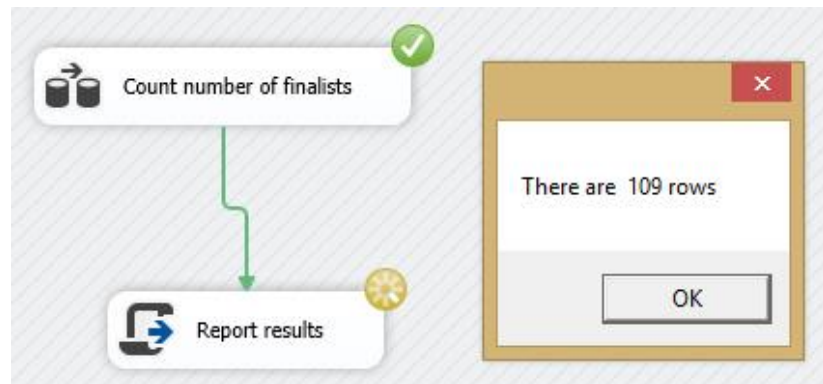
Script task using C#



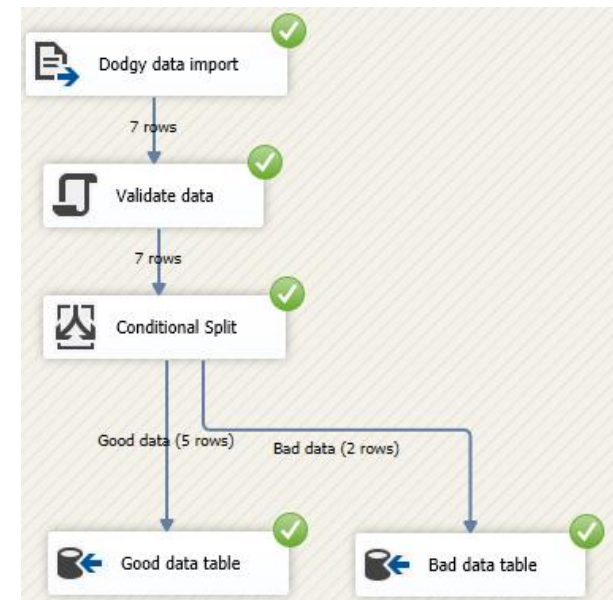
Show row number



Variables



Script component



SSAS demo (CUBE)

Data Source View

Customer

- CustomerKey
- GeographyKey
- CustomerAlternateKey
- Title
- FirstName
- MiddleName
- LastName
- NameStyle
- BirthDate
- MaritalStatus

Geography

- GeographyKey
- City
- StateProvinceCode
- StateProvinceName
- CountryRegionCode
- EnglishCountryRegionName
- SpanishCountryRegionName
- FrenchCountryRegionName
- PostalCode
- SalesTerritoryKey

InternetSales

- SalesOrderNumber
- SalesOrderLineNumber
- ProductKey
- OrderDateKey
- DueDateKey
- ShipDateKey
- CustomerKey
- PromotionKey
- CurrencyKey
- SalesTerritoryKey

Date

- DateKey
- FullDateAlternateKey
- DayNumberOfWeek
- EnglishDayNameOfWeek
- SpanishDayNameOfWeek
- FrenchDayNameOfWeek
- DayNumberOfMonth
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName

Product

- ProductKey
- ProductAlternateKey
- ProductSubcategoryKey
- WeightUnitMeasureCode
- SizeUnitMeasureCode
- EnglishProductName
- SpanishProductName
- FrenchProductName
- StandardCost
- FinishedGoodsFlag

Product.dim [Design] Employee.dim [Design] Date.dim [Design]* Customer.dim [Design]

Dimension Structure Attribute Relationships Translations Browser

Attributes

Date

- Calendar Quarter
- Calendar Semester
- Calendar Year
- Date Key
- Day Number Of Month
- Day Number Of Week
- Day Number Of Year
- English Day Name Of Week
- English Month Name
- Fiscal Quarter
- Fiscal Semester
- Fiscal Year
- French Day Name Of Week
- French Month Name
- Full Date Alternate Key
- Month Number Of Year
- Spanish Day Name Of Week
- Spanish Month Name
- Week Number Of Year

Hierarchies

Date

- Calendar Year
- Calendar Semester
- Calendar Quarter
- Month
- Day
- <new level>

To create a new hierarchy, drag an attribute here.

Data Source View

Date

- DateKey
- FullDateAlternateKey
- DayNumberOfWeek
- EnglishDayNameOfWeek
- SpanishDayNameOfWeek
- FrenchDayNameOfWeek
- DayNumberOfMonth
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName

Current level: Calendar Year

All

2005

2

3

July

August

September

- 2005-09-01
- 2005-09-02
- 2005-09-03
- 2005-09-04
- 2005-09-05
- 2005-09-06
- 2005-09-07
- 2005-09-08
- 2005-09-09
- 2005-09-10
- 2005-09-11
- 2005-09-12
- 2005-09-13
- 2005-09-14
- 2005-09-15
- 2005-09-16
- 2005-09-17
- 2005-09-18
- 2005-09-19
- 2005-09-20
- 2005-09-21
- 2005-09-22
- 2005-09-23
- 2005-09-24
- 2005-09-25

Excel BI demos

Key Performance Indicators (KPIs):

Row Labels	Ratio	Ratio Goal	Ratio Status
2014	100.80 %	100.00 %	
Q1	105.27 %	100.00 %	
February	103.43 %	100.00 %	
January	104.64 %	100.00 %	
March	107.38 %	100.00 %	
Q2	104.37 %	100.00 %	
April	93.73 %	100.00 %	
June	107.19 %	100.00 %	
May	112.95 %	100.00 %	
Q3	98.83 %	100.00 %	
August	105.13 %	100.00 %	
July	91.82 %	100.00 %	
September	101.17 %	100.00 %	
Q4	95.28 %	100.00 %	
December	89.82 %	100.00 %	
November	96.25 %	100.00 %	
October	99.76 %	100.00 %	
Grand Total	100.80 %	100.00 %	

Hierarchy:

Classification	Frog								
Sum of Quantity	Column Labels								
Row Labels	Q1	Q2	Q3	Q4	2013 Total	2014	Grand Total		
East	15	13	13	14	55	83	138		
East Anglia	4	2	5	7	18	17	35		
Barhill			5		5	2	7		
Cambridge				1	1	10	11		
Ipswich	2				2	1	3		
Lowestoft	2			1	3		3		
Norwich		1		4	5	3	8		
Peterborough		1		1	2	1	3		
East Midlands	11	11	8	7	37	66	103		
North	51	59	68	70	248	289	537		
South	70	66	88	83	307	220	527		
West	27	28	43	38	136	98	234		
Grand Total	163	166	212	205	746	690	1436		

PowerPivot:

Filter:

CentreTypeName			
Factory ...			
Retail Park			
Shoppin...			
Shoppin...			

Sum of Quantity	West	South	North
Bird	2	13	19
Mammal	2	13	28
Reptile	3	18	19
Amphibian	4	6	3

PosDate						
Q3 - Q4 2013						
2013						
2014						
Q1	Q2	Q3	Q4	Q1	Q2	Q3

Calculated Function DAX (Data Analysis Expressions):

Column Labels	Amphibian	Bird	Mammal	Reptile	Total Right	Total Qty
Row Labels	Right	Qty	Right	Qty	Right	Qty
East	79	79	262	262	428	428
North	332	332	1221	1221	1714	1714
South	404		1400		1972	1426
West	167	167	622	622	952	952
Grand Total	578	982	2105	3505	3094	5066

CentreTypeName						
Factory Outlet						
Retail Park						
Shopping Centre						
Shopping Park						

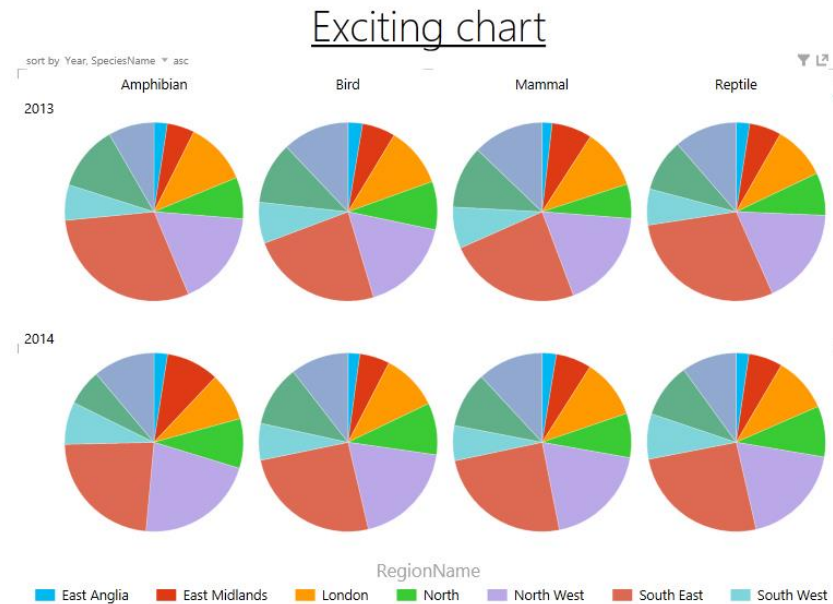
Calculated Columns:

Sum of Amount	Column Labels			
Row Labels	High	Low	Medium	Grand Total
Cat		6130.54	9468.02	15598.56
Possum			9468.02	9468.02
Snowy		6130.54		6130.54
Crocodile	7370.925		6732.33	14103.255
Crocky	7370.925		6732.33	14103.255
Frog			10985.3305	10985.3305
Fred			10985.3305	10985.3305
Owl	230	4955.591	11848.5	17034.091
Speccles	230		11848.5	12078.5
Wol		4955.591		4955.591
Penguin	21779.1873		6052.2	27831.3873
Pingu	8400.48		6052.2	14452.68
The Emperor	13378.7073			13378.7073
Sheep		4622.145	12594.41	17216.555
Sean		4622.145	2546.17	7168.315
Woolly			10048.24	10048.24
Snake	27756.3		9210.525	36966.825
Simon	27756.3			27756.3
Slithery			9210.525	9210.525
Tiger			10915.236	10915.236
Tigger			10915.236	10915.236
Grand Total	57136.4123	15708.276	77806.5515	150651.2398

<https://github.com/charleseagle/Excel-Business-Intelligence-including-Power-Pivot-Power-Vlew-Power-Query-Power-Map>

Excel BI demos

Power view:



Count of StaffId	Column Labels															
Row Labels	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986
1953																
1954		1	1	2	1	2	2		1	1	4	2	1	1	1	1
1955			1	1	2	2	3	1	1	3		1	1		5	1
1956				2	3	1		1	1		1	1	1	2	1	3
1957				3	4	1		1	1	3		3	2		2	1
1958					1	2	2	2	1	1	2	1	1	2	1	1
1959						1					1	2	2	1	2	3
1960								3	2	3	1	2	2	1	1	2
1961							1	2	2	4	1	4		2	1	1
1962								1	2	4	1	1	2			1
1963										2	2		1	1	1	2
1964											2	2	1	4	1	1
1965												2	3	1	1	5
1966												2	1	4	3	1
1967															1	1
1968															3	1
1969																3
1970																
1971																

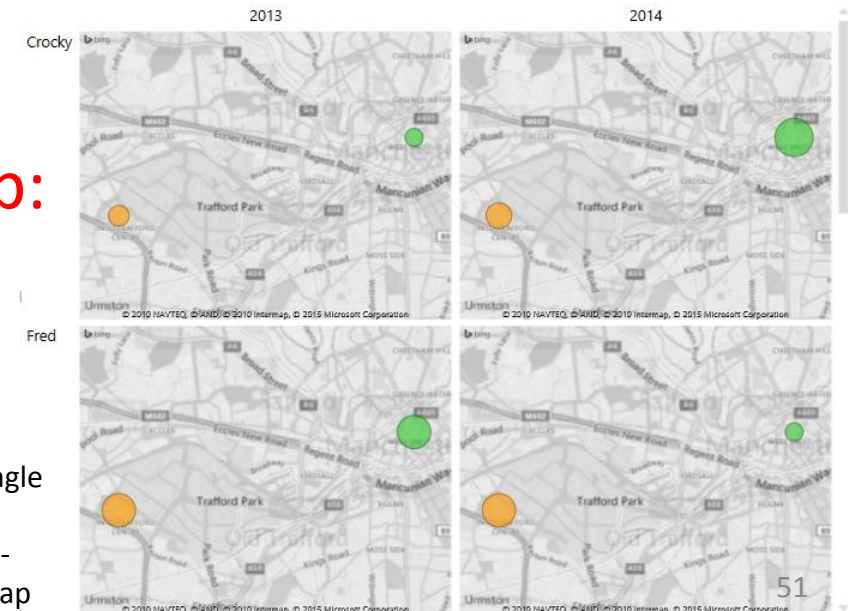
Power pivot: Matrix

Power query:

Rank	Country	Population	% of world population
1	China	1373250000	19
2	India	1280230000	18
3	United States	322257000	4
4	Indonesia	255461700	4
5	Brazil	205209000	3
6	Pakistan	188925000	3
7	Nigeria	182202000	2
8	Bangladesh	159387000	2
9	Russia	146439880	2
10	Japan	126890000	2
11	Mexico	121005000	2
12	Philippines	102370400	1
13	Vietnam	90730000	1
14	Ethiopia	90076012	1
15	Egypt	89873500	1
16	Germany	81197500	1
17	Iran	78795800	1
18	Turkey	77695904	1

Power map:

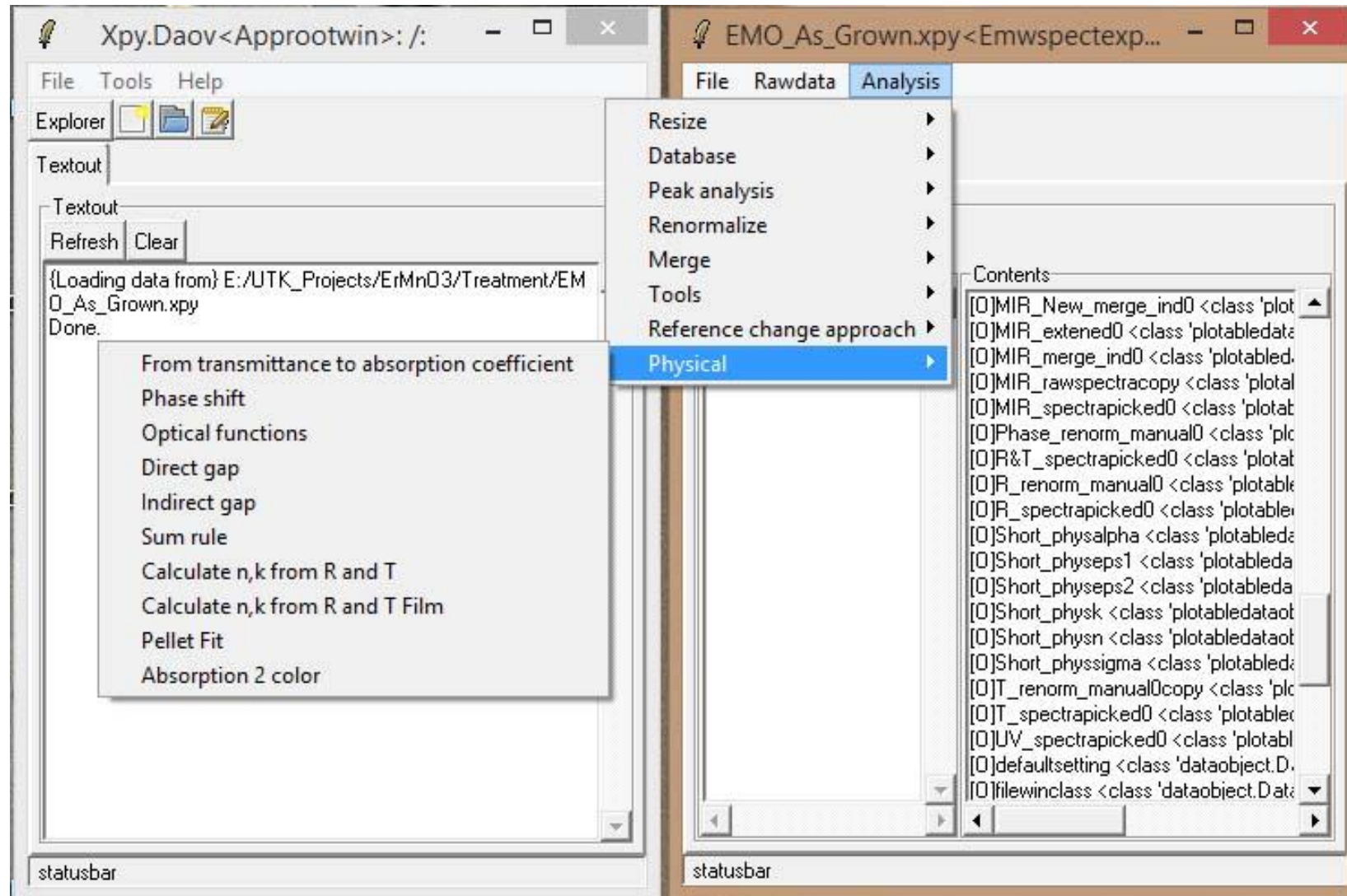
<https://github.com/charleseagle/Excel-Business-Intelligence-including-Power-Pivot-Power-View-Power-Query-Power-Map>



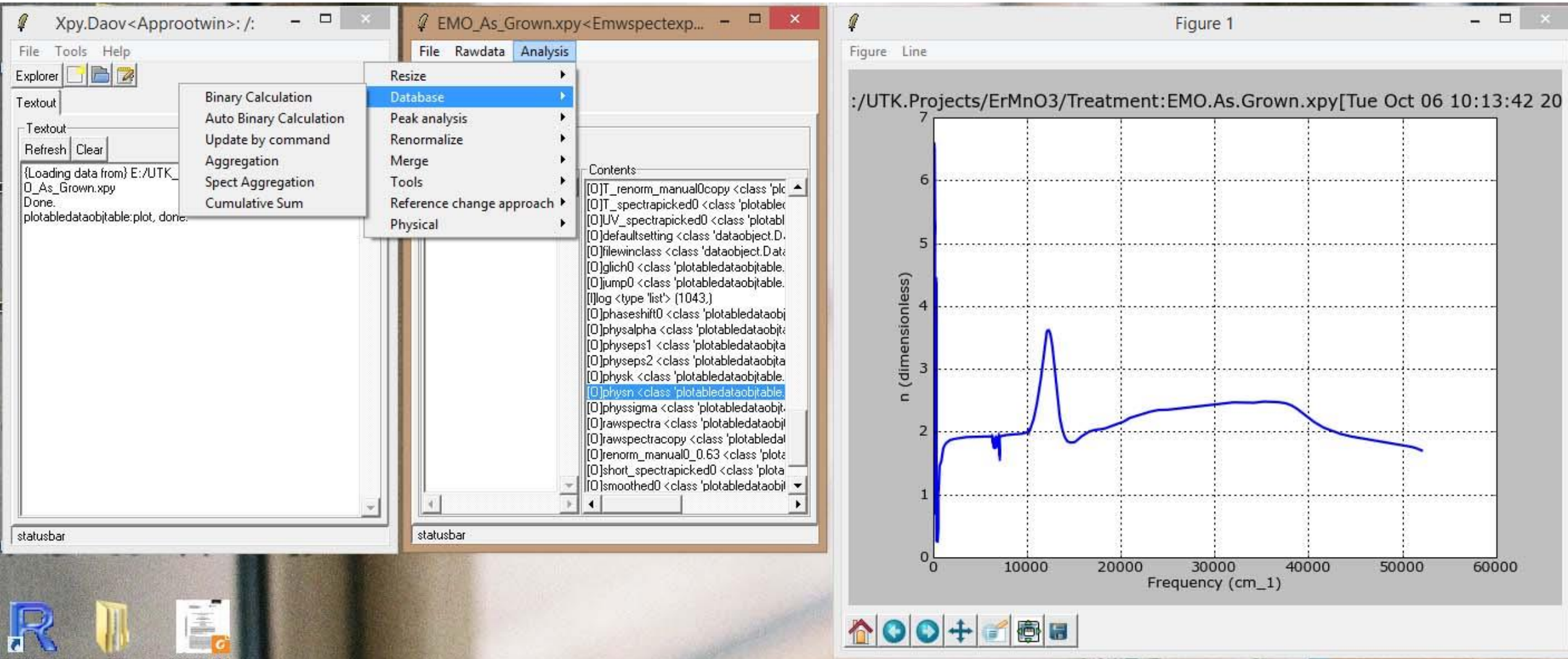
Programing demos

1. Developed software for analyzing data using Python
2. Matlab and Python FDTD simulation
3. Gaussian fit analysis
4. Rewrote Machine Learning codes using Python
5. C# demo for linear regression
6. Labview-controlled photoluminescence system

Analysis software (Python)

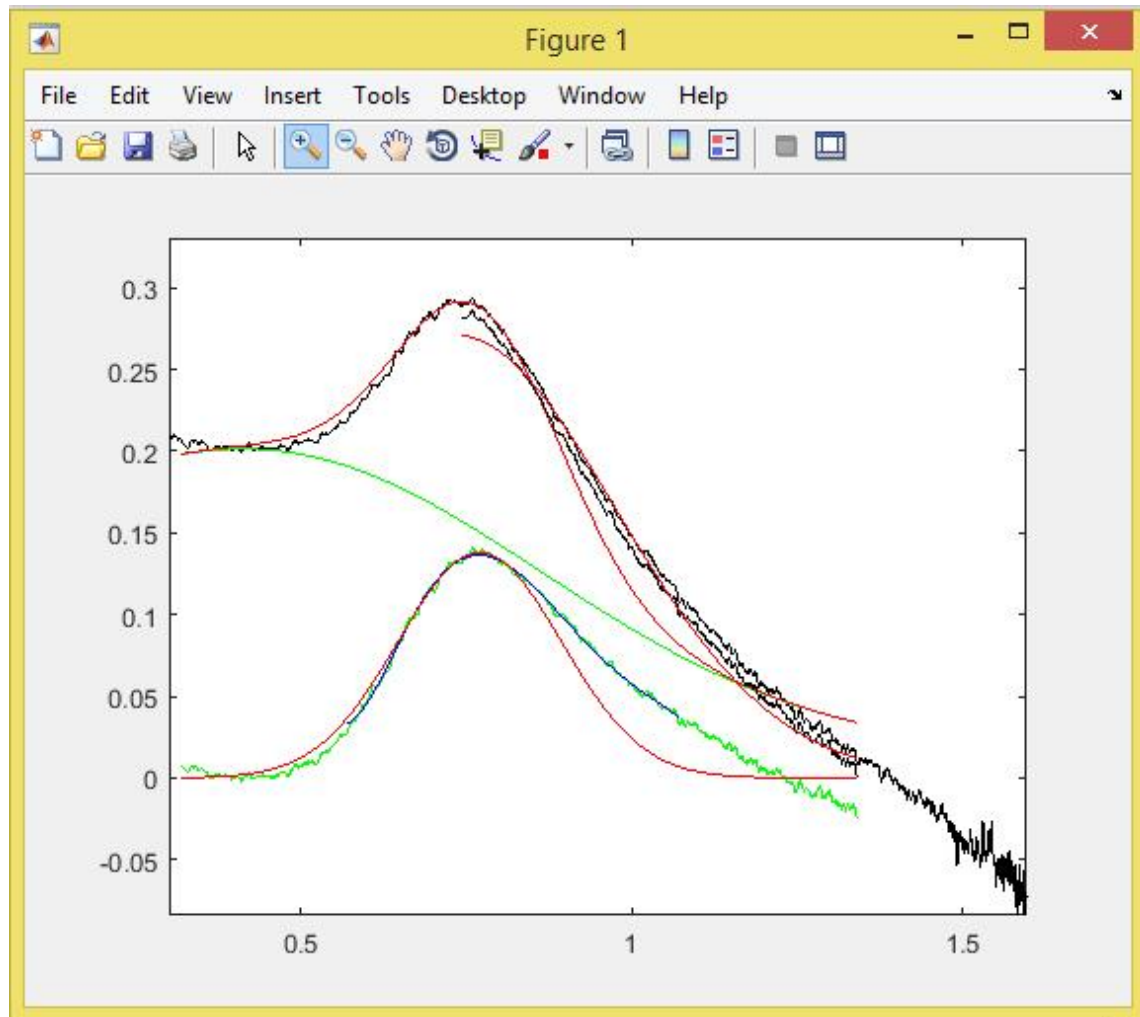


Analysis software (Python)

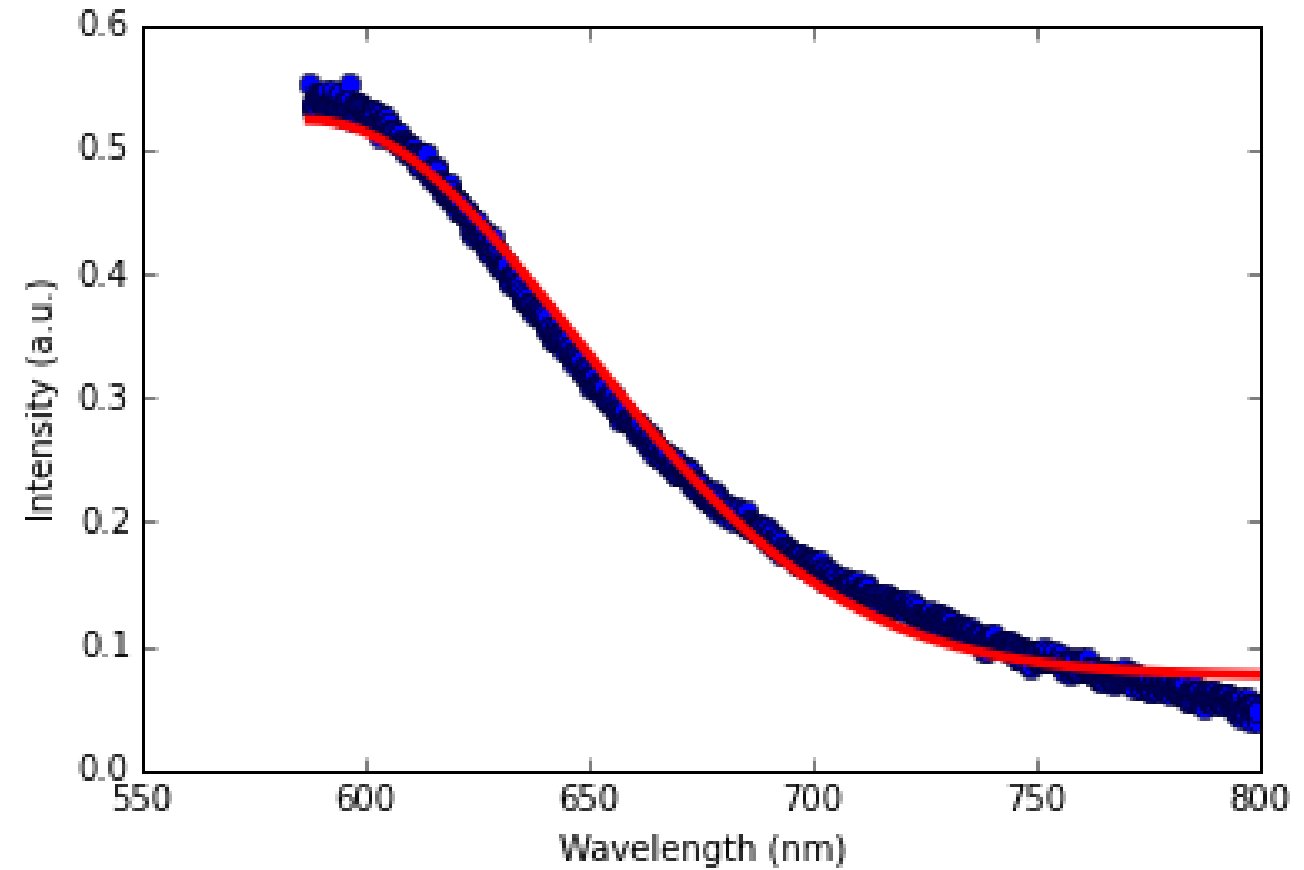


Gaussian fit analysis

Matlab demo



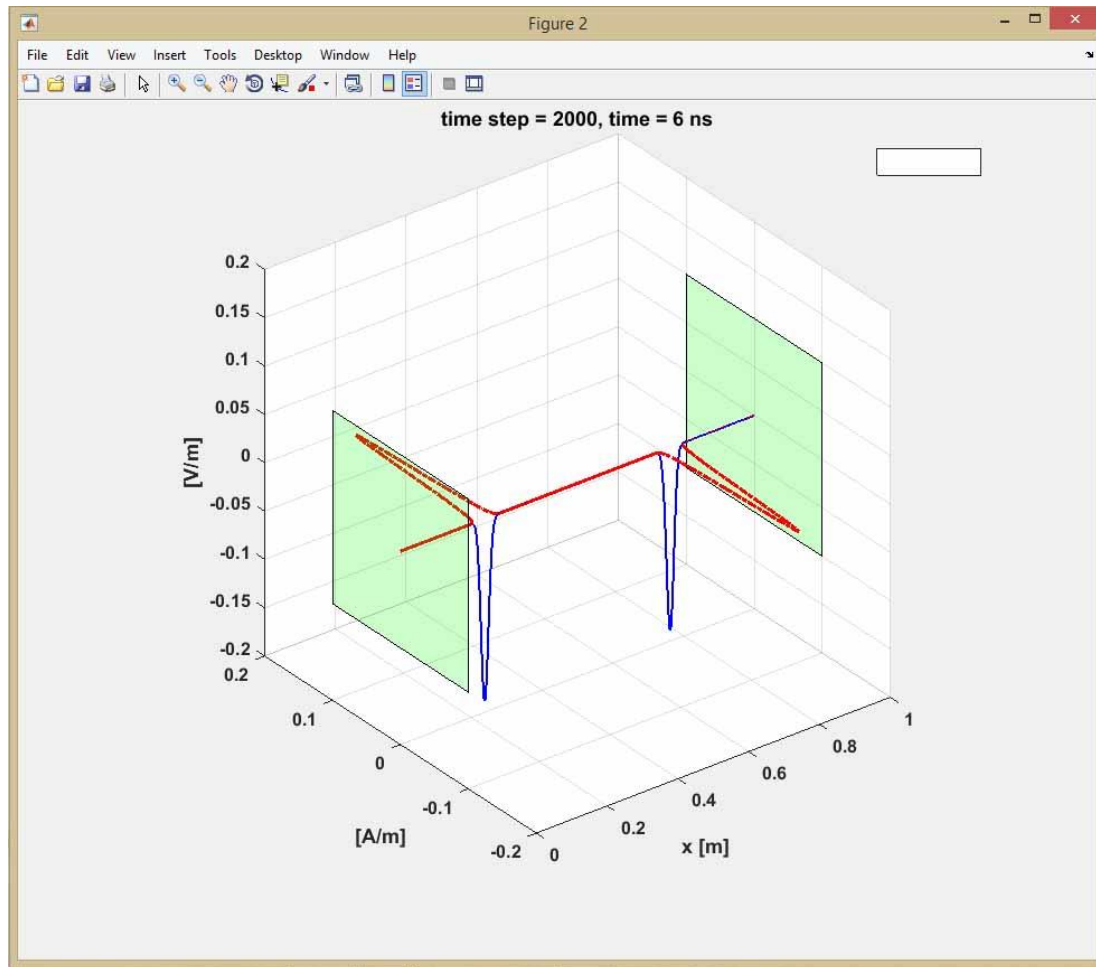
Python demo



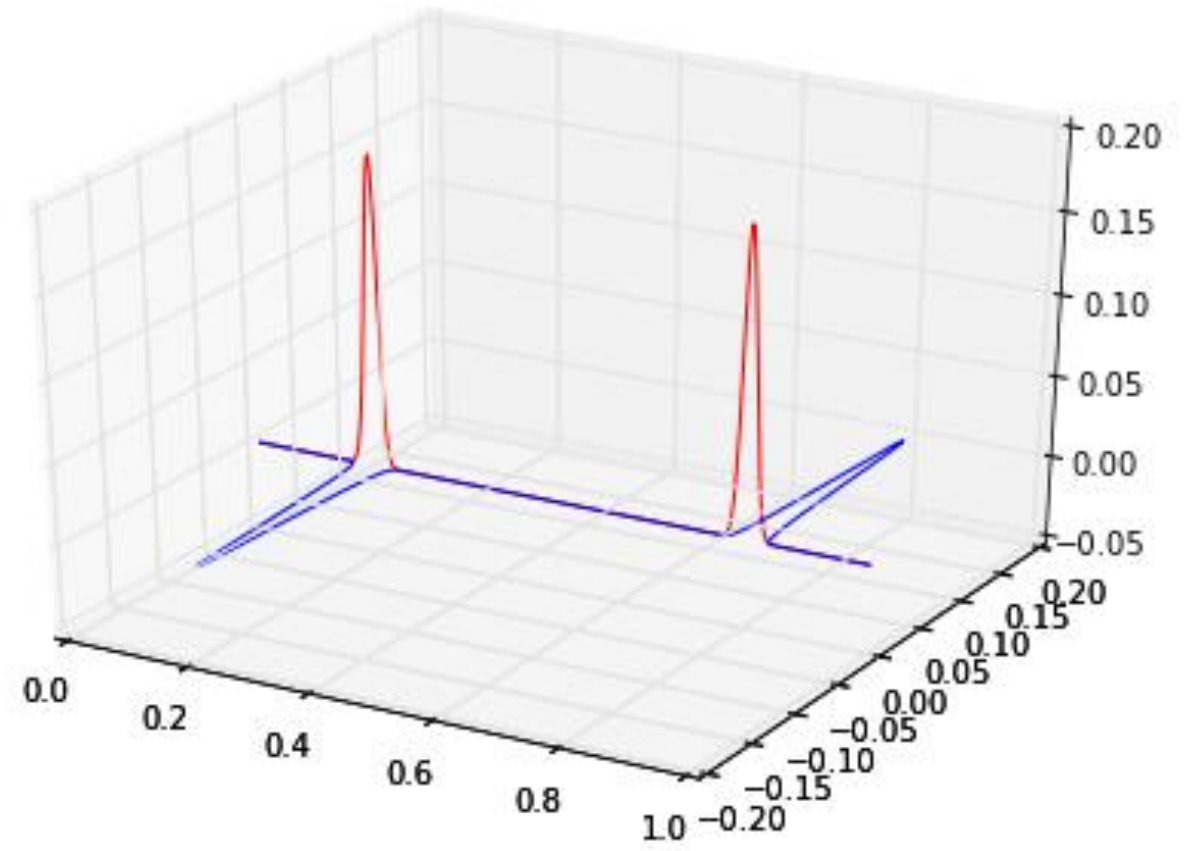
<https://github.com/charleseagle/Fitting-analysis-for-spectroscopic-data-using-Python-and-Matlab>

FDTD simulation

Matlab demo

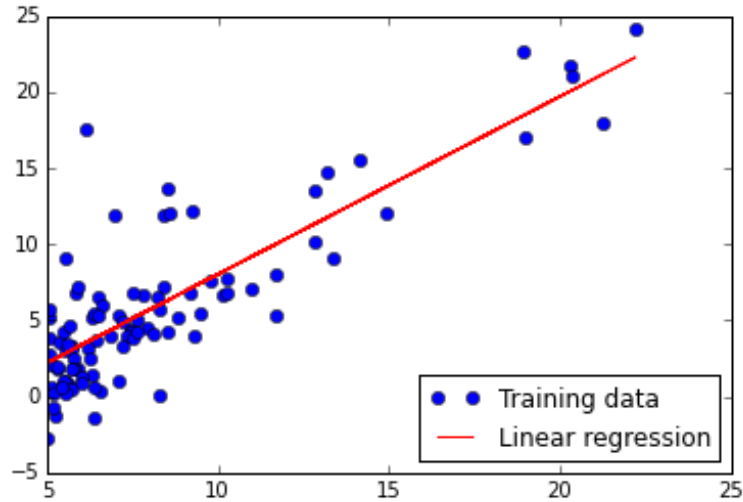


Python demo

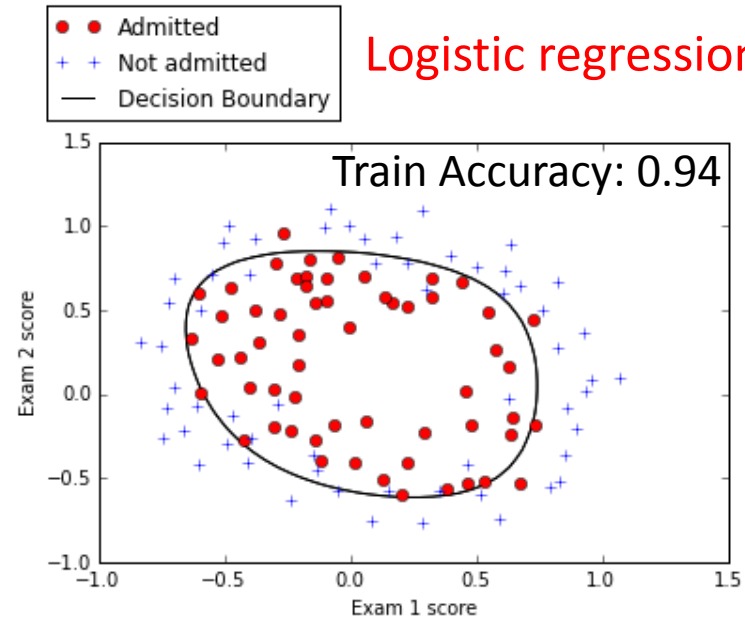


Machine Learning (Python)

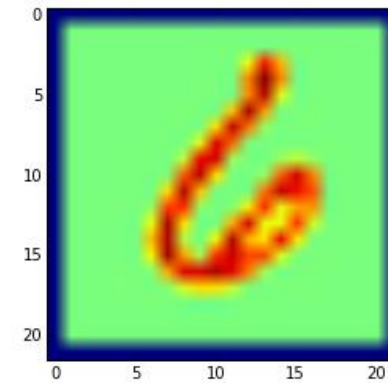
Linear regression



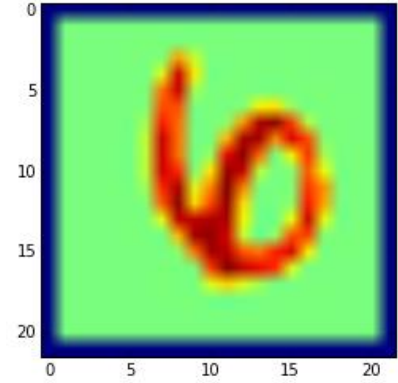
Logistic regression



Neuron network



Prediction: 6



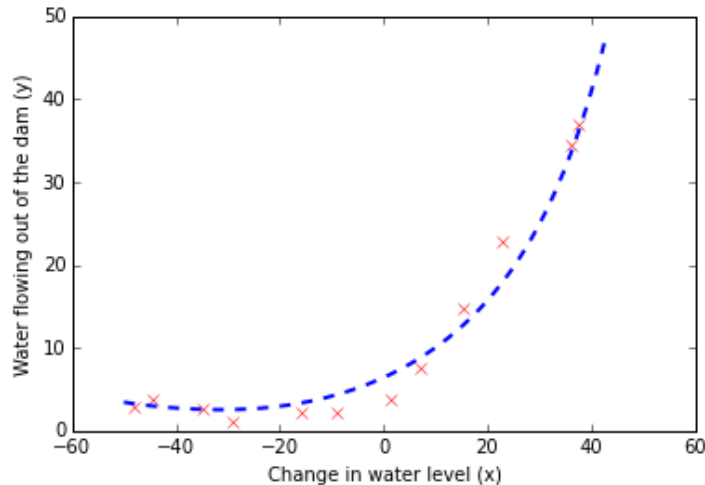
Prediction: 6

Original Data

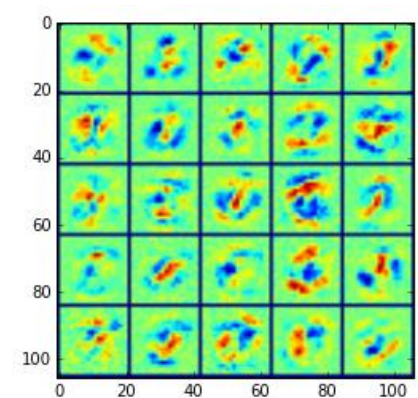
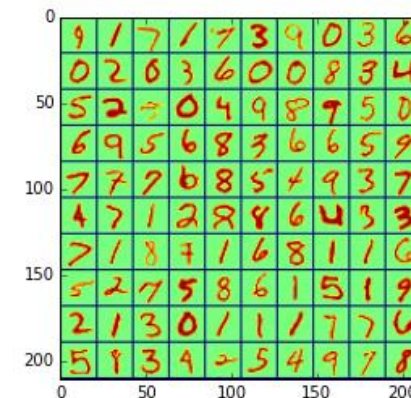
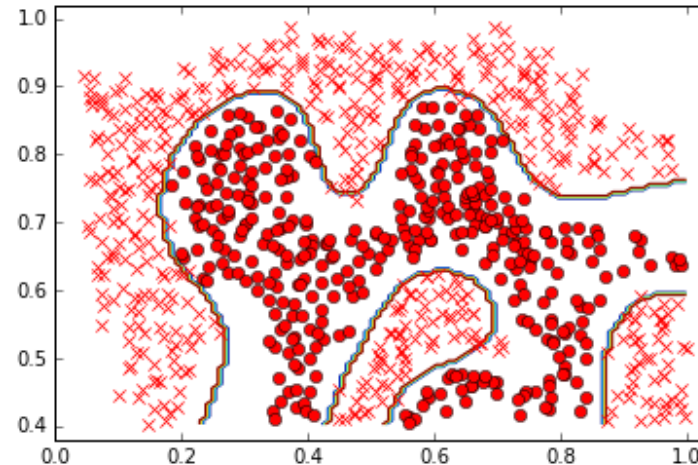
Neural Network

Regularized linear regression

Polynomial Regression Fit Lambda



Support vector machines

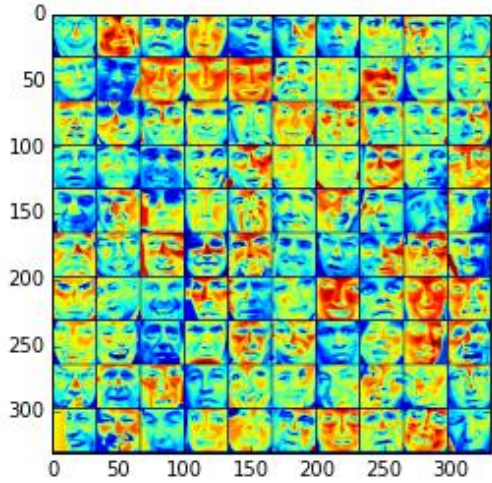


Training Set Accuracy: 0.91

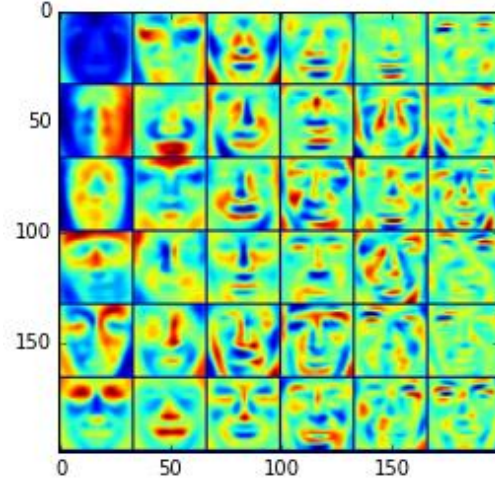
Machine Learning (Python)

Principle component analysis

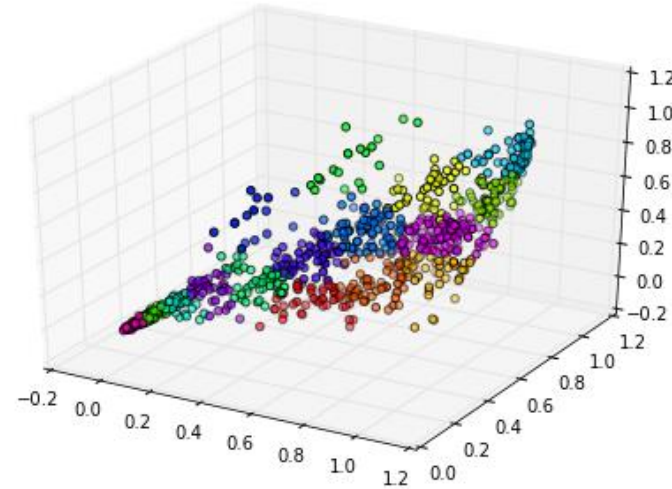
Original data



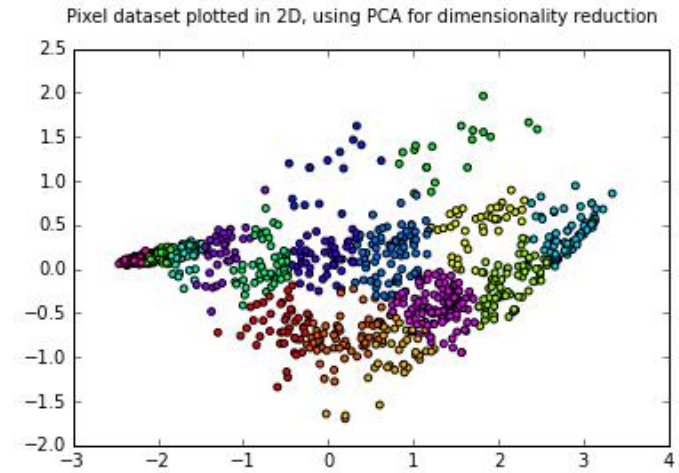
After PCA



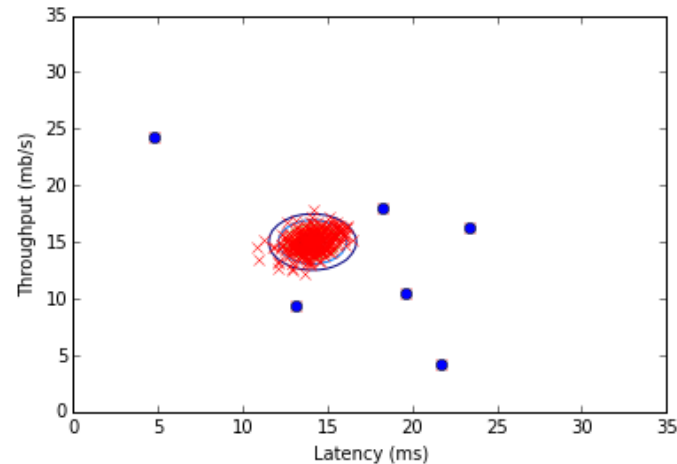
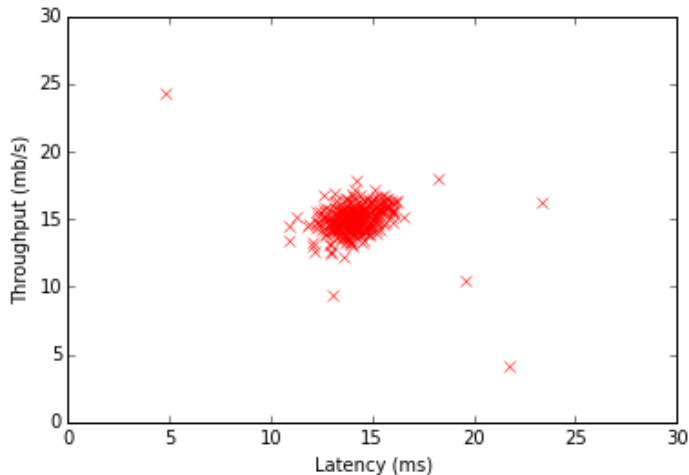
Original data in 3D



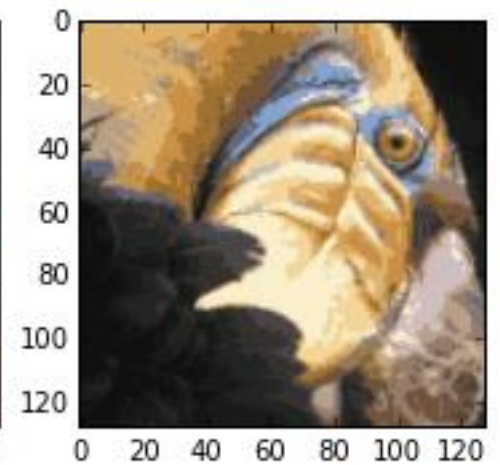
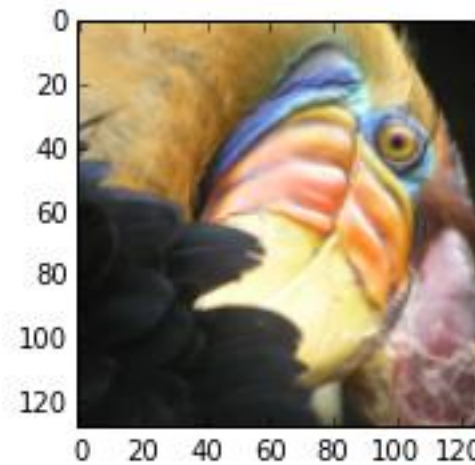
2D visualization using PCA



Anomaly detection



k-mean clustering



Linear regression (C#)

```
//Gradient descent for linear regression.
```

```
1 reference
```

```
public Tuple<double[,], double[,]> Gradient(double[,] X, double[,] y,  
    double[,] theta, double alpha, int num_iters)
```

```
{  
    double m = y.GetLength(0);  
    double[,] J_history = new double[num_iters,1];  
    for (int i = 0; i < num_iters; i++)  
    {  
        double[,] temp1 = ArrayMultiply(X, theta);  
        double[,] temp2 = ArrayMinus(temp1, y);  
        var temp = new double[X.GetLength(1), X.GetLength(0)];  
        for (int k = 0; k < X.GetLength(0); k++)  
        {  
            for (int j = 0; j < X.GetLength(1); j++)  
            {  
                temp[j, k] = X[k, j];  
            }  
        }  
        double[,] temp3 = ArrayMultiply(temp, temp2);  
        for (int j = 0; j < temp3.GetLength(0); j++)  
        {  
            temp3[j, 0] = (alpha / m) * temp3[j, 0];  
        }  
        theta = ArrayMinus(theta, temp3);  
        J_history[i,0] = ComputeCost(X, y, theta);  
    }  
    return new Tuple<double[,], double[,]>(theta, J_history);  
}
```

```
2 references
```

```
public Tuple<double[,], double[,], double, double, double, double[,]> LinearFit()
```

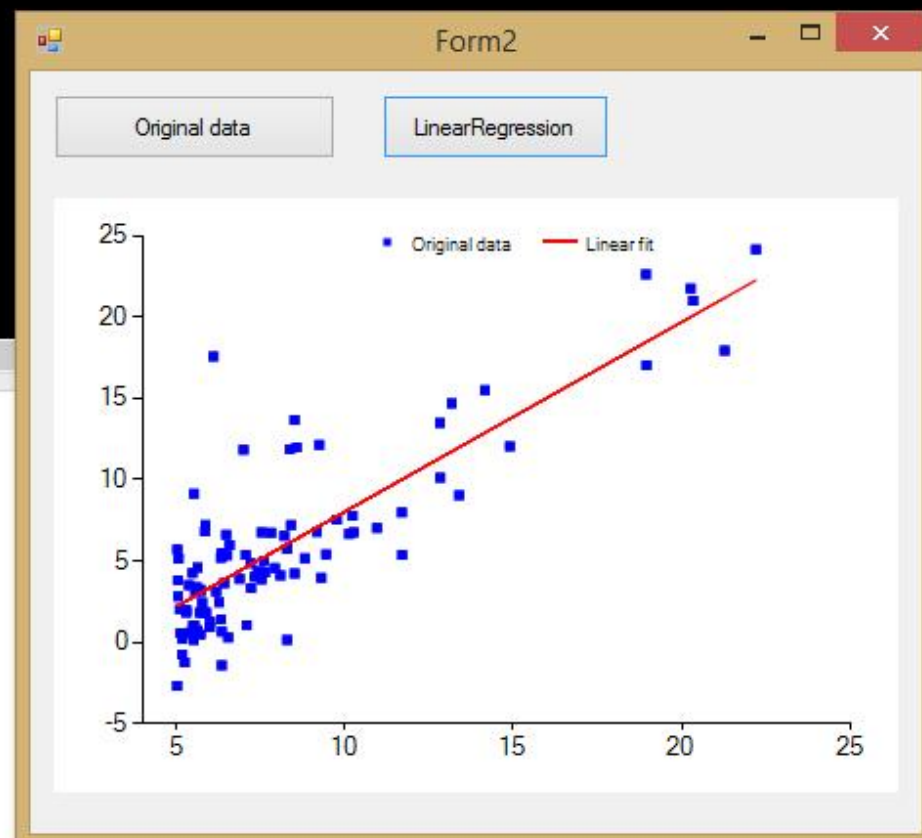
```
{  
    Program t = new Program();  
    double[,] data;  
    t.FillArray(out data);  
    var X = new double[data.GetLength(0), 2];  
    var y = new double[data.GetLength(0), 1];  
}
```

file:///E:/Machine learning/Week_2/C#/LinearRegression/LinearRe...

Initial cost: 32.0727338774557

Running gradient descent.

Theta found by gradient descent: -3.63029143940436, 1.16636235033558



Labview-controlled photoluminescence system

