

*Hong Kong Baptist University
Department of Computer Science*

COMP 7810/4096 Business Intelligence (2019-20)

SQL Server Management Studio (SSMS) & SQL Server Integration Services (SSIS)

Introduction

SQL Server Management Studio (SSMS) is a tool for administrating and developing SQL Server database. It includes some functions like creating databases, design tables, adding data, updating data and running queries.

SQL Server Data Tools (SSDT) is a modern development tool for building Analysis Services (AS) data models, Integration Services (IS) packages, and Reporting Services (RS) reports. It is used for creating and maintaining SQL server Integration Services (SSIS) projects. You can use Integration Services for building high performance data integration solutions, including extraction, transformation, and loading (ETL) of data for data warehousing.

Learning Outcomes

By finishing this lab session, you should be able to

- Get familiar with SQL Server Management Studio (SSMS)
- Import excel spreadsheet into SQL Server
- Use SSDT (Integration Services) to extract and transform data

Tools

- Microsoft Excel 2013
- Microsoft SQL Server Management Studio 2012
- Visual Studio 2010 with SQL Server Data Tools (SSDT)

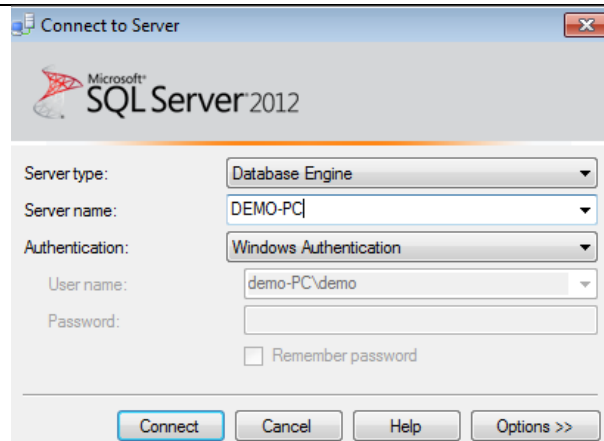
Source file required

- datasource.xlsx (save it in C:\) from <http://buelearning.hkbu.edu.hk/>
- commission.txt (save it in C:\)

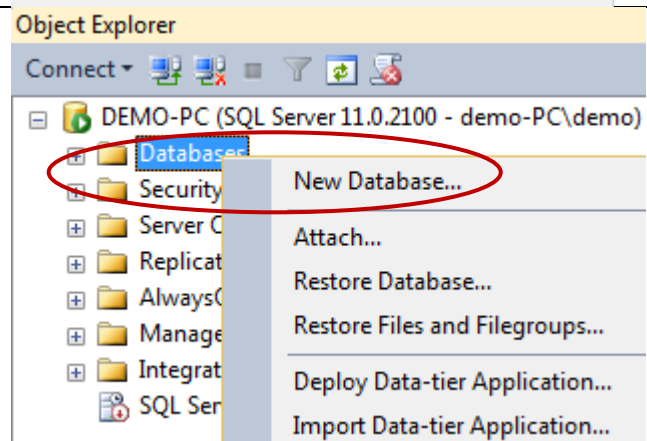
Part A: SQL Server Management Studio (SSMS)

I. Create a new database and import data to the database

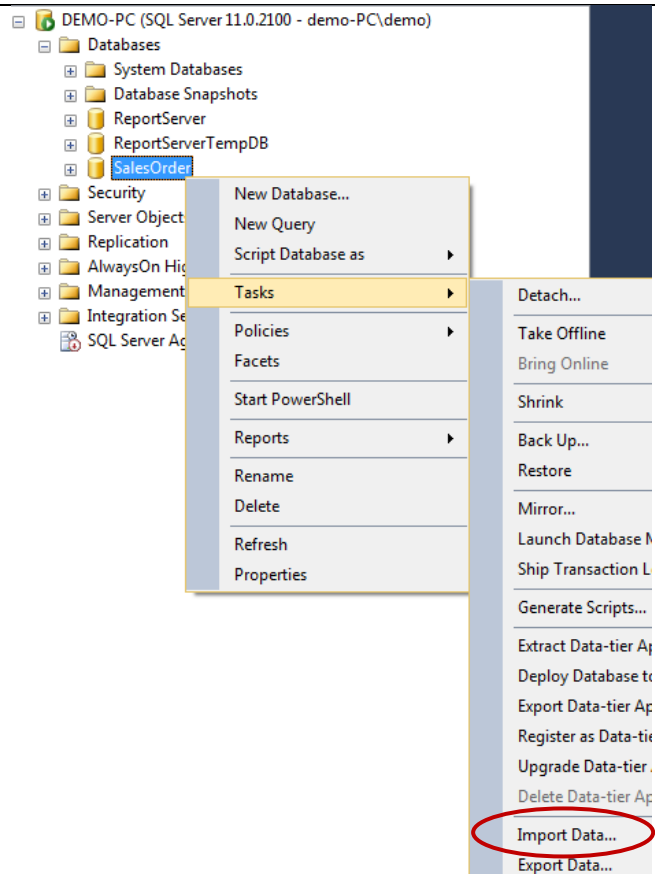
1. Open **SQL Server Management Studio** and press **Connect**.



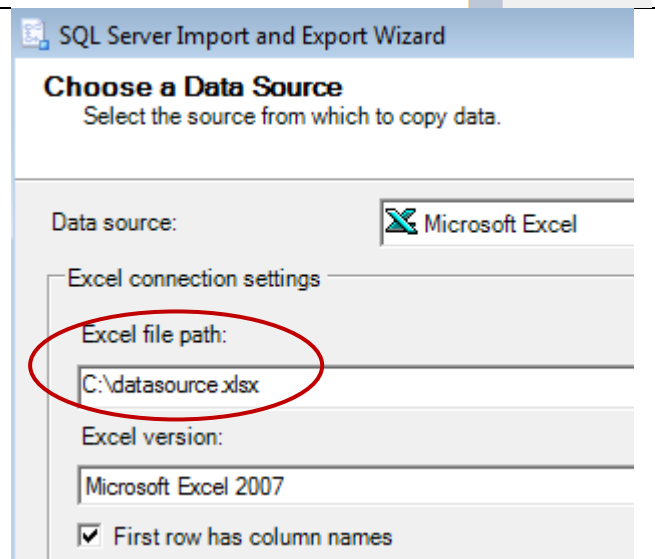
2. Right click **"Databases"**, select **New Database** to create a new database named **SalesOrder**

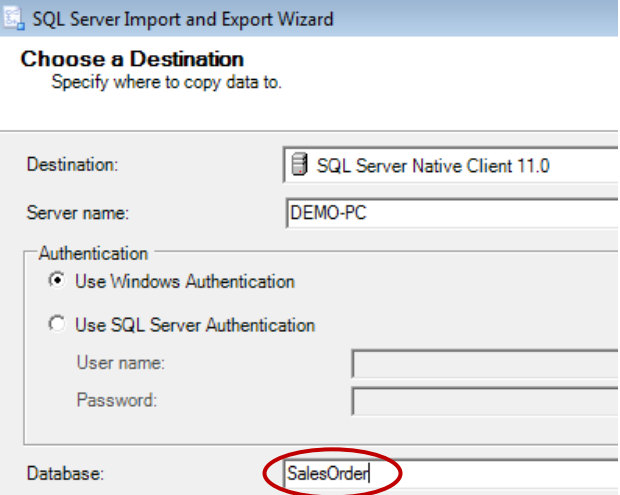
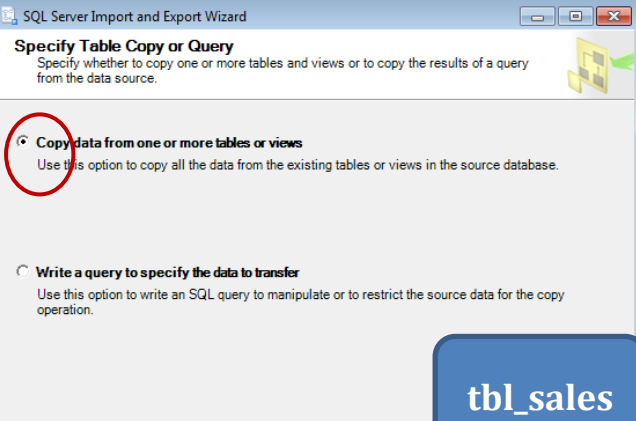
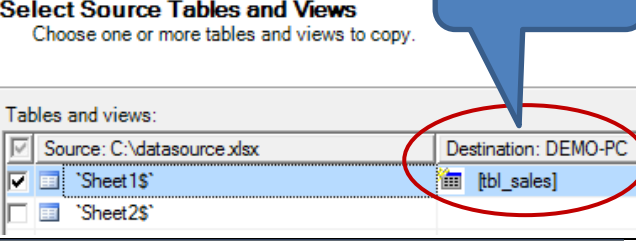
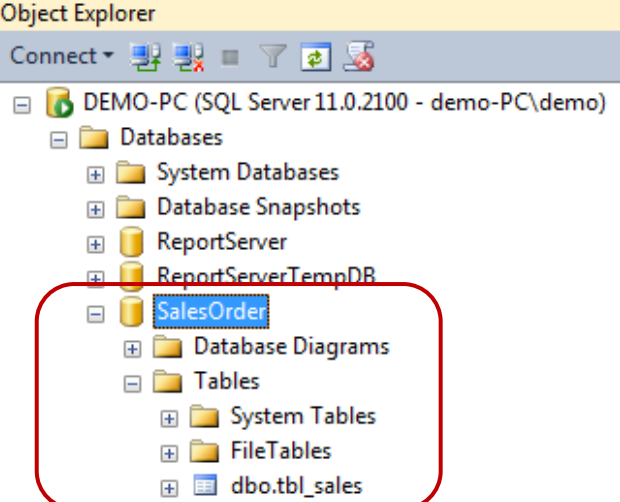


3. **Right click** the **SalesOrder** database, select **Tasks** → **Import Data** to import an excel file. Press **Next** to continue.



4. Select **Microsoft Excel** as data source, change the excel version to **Microsoft Excel 2007**, choose the file **datasource.xlsx** in **C:** as the excel file path. Press **Next** to continue.

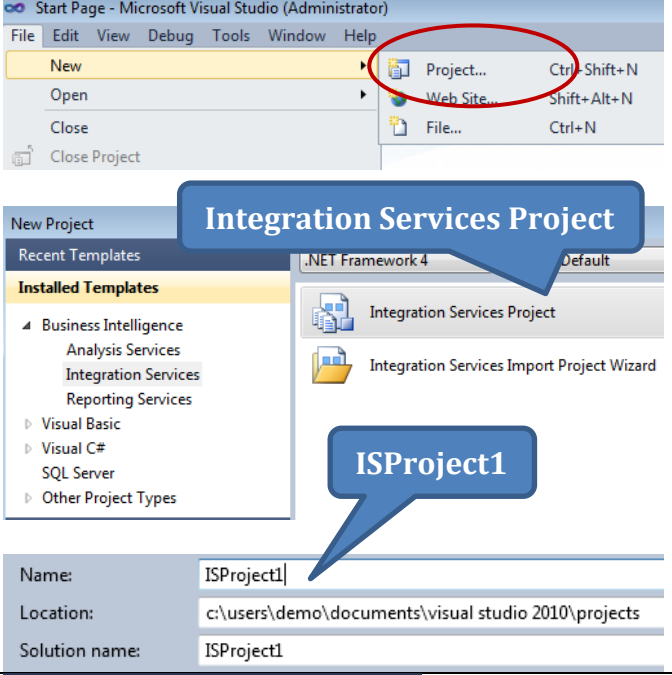
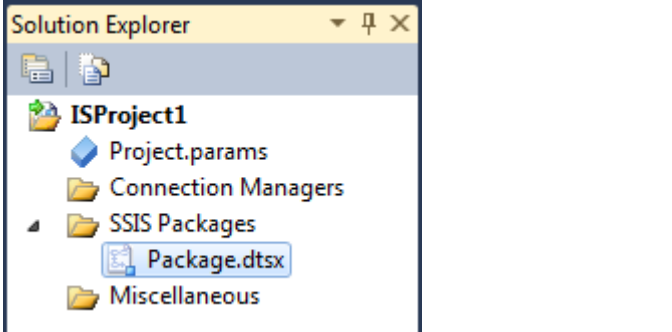


<p>5. Select SalesOrder database as the destination. Press Next to continue.</p>	
<p>6. Select Copy data from one or more tables or views. Press Next to continue.</p>	
<p>7. Check the source 'Sheet1\$'. Enter the name of table i.e. tbl_sales under Destination. Select appropriate data types by clicking Edit Mappings. Press OK then Next to continue, then Next → Finish → Close</p>	
<p>8. The table tbl_sales is created in the SalesOrder database and 799 records are imported.</p>	

Part B: SQL Server Integration Service (SSIS)

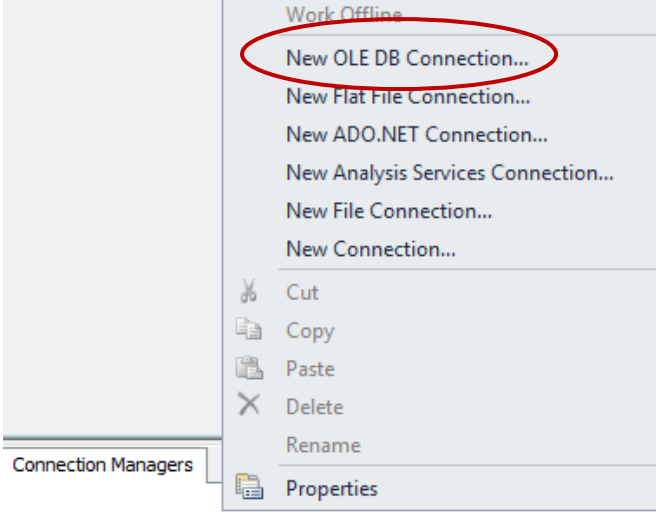
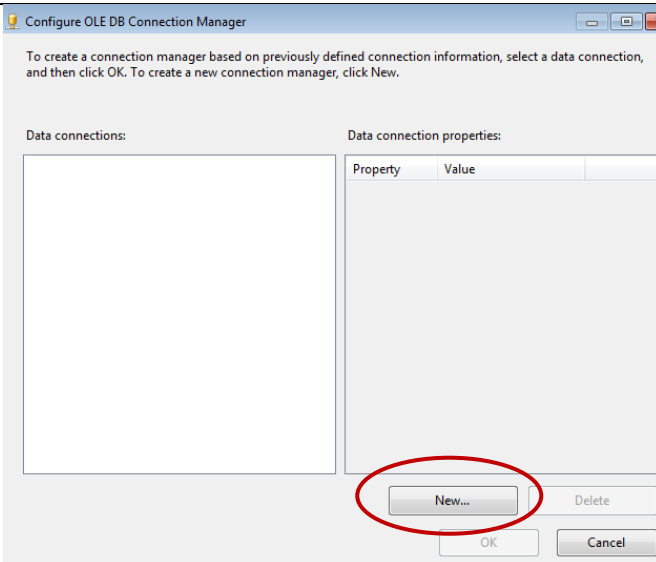
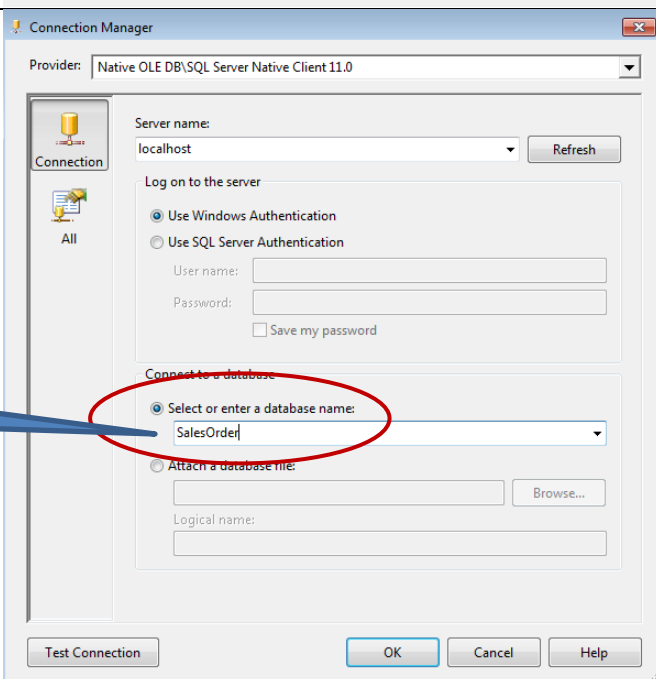
I. Create a new SSIS package

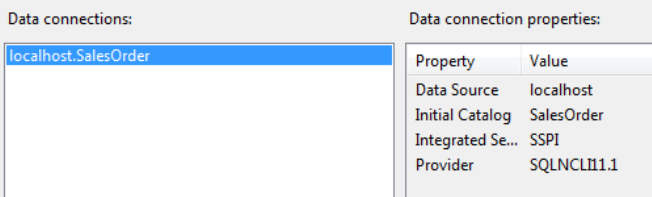
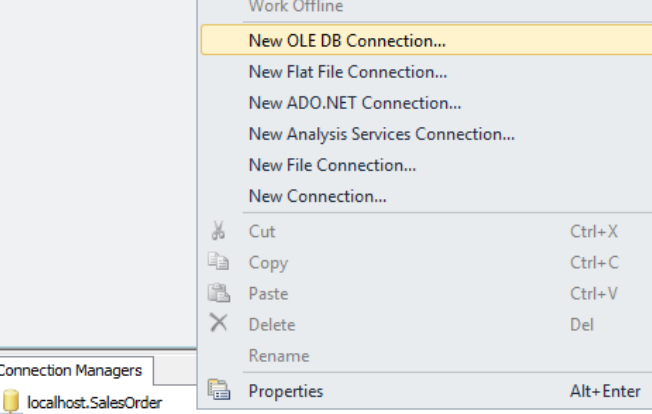
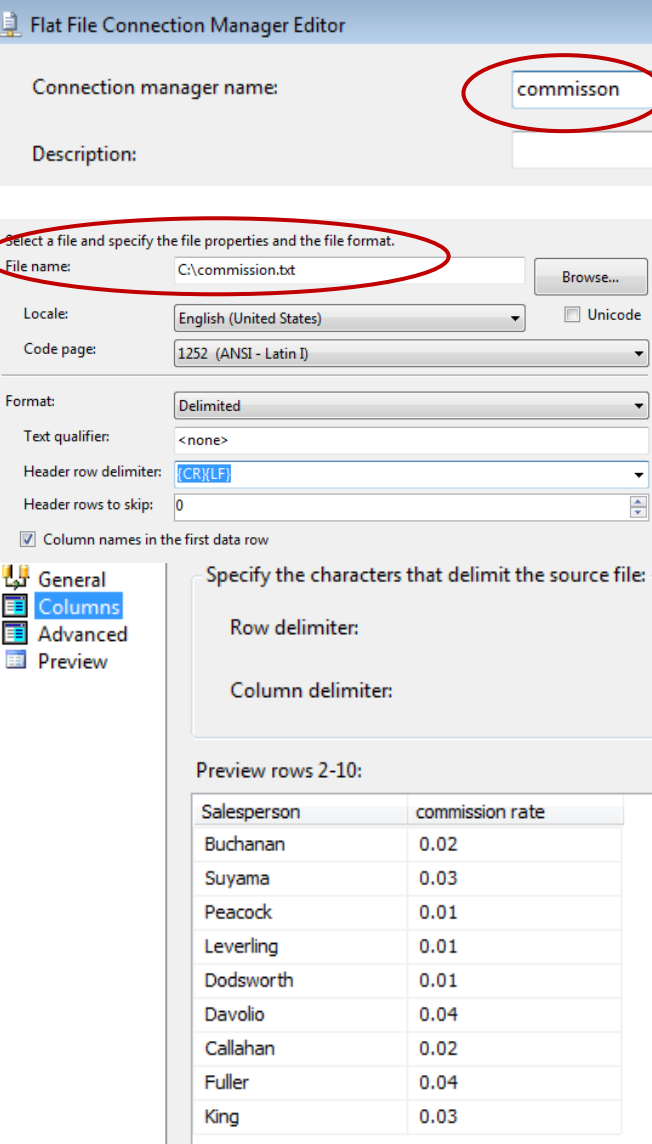
For most users, SQL Server Data Tools (SSDT) is installed during Visual Studio installation. To perform ETL, you need to create a new Integration Services project:

<ol style="list-style-type: none"> 1. Close the program SQL Server Management Studio (SSMS). 2. Open Visual Studio with SQL Server Data Tools (SSDT) 3. Select File → New → Project. 4. Select the Integration Services under Business Intelligence templates. 5. Type ISProject1 as the project name (The solution name will be ISProject1) and click OK. 	 <p>The screenshot shows the Visual Studio 'New Project' dialog. The 'File' menu is open, and 'New' is selected. The 'Integration Services Project' template is chosen under 'Business Intelligence'. The project name is 'ISProject1'.</p>
<ol style="list-style-type: none"> 6. In Solution Explorer, a new SSIS Packages Package.dtsx is created. 	 <p>The screenshot shows the Solution Explorer with the project 'ISProject1'. It contains a 'Package.dtsx' file, which is highlighted.</p>

II. Create a Connection

Connection managers integrates different data sources into packages. An OLE DB connection manager enables a package to extract data from the database:

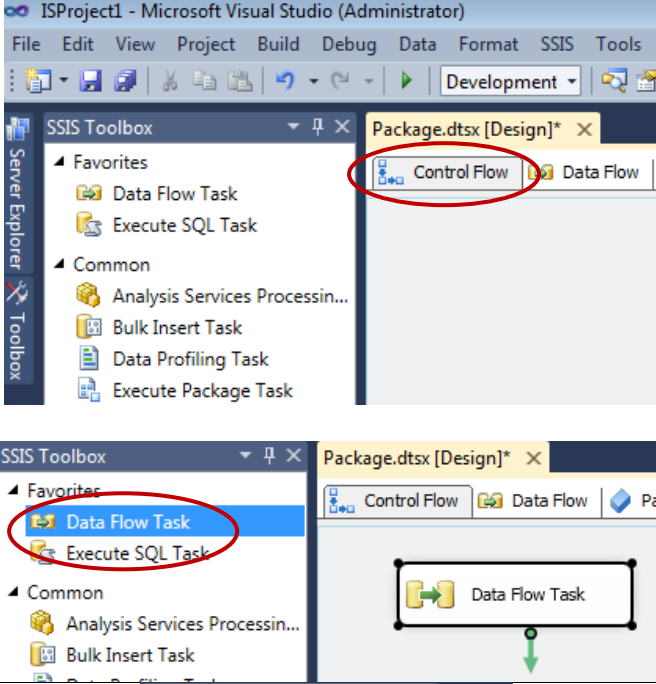
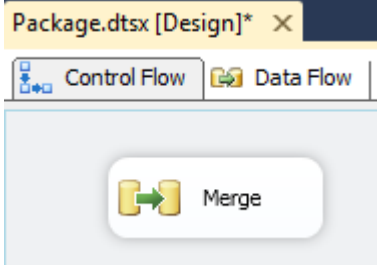
<p>1. Right click anywhere in the Connection Managers area and choose New OLE DB Connection.</p>	
<p>2. Press New to create new connection to the SQL server.</p>	
<p>3. In the Connection Manager dialog box, select the default provider Native OLE DB/SQL Server Native Client 11.0</p> <p>4. Type localhost as server name</p> <p>5. Select the SalesOrder database and press OK</p>	

<p>6. The connection is created. Press OK</p>	
<p>7. Repeat the previous steps to add another connection (New Flat File Connection)</p>	
<p>8. Type commission as the Connection manager name.</p> <p>9. Click Browse button to select the txt file (commission.txt) you have downloaded.</p> <p>10. Check the box Column names in the first data row.</p> <p>11. Select Columns option on the left hand side to preview the result.</p> <p>12. And press OK.</p>	

III. Add a Data Flow Task

The next task is to add a **Data Flow task** to your package, the **Data Flow task** provides the functionality for transforming, cleaning, and modifying data.

Data Flow Task can be added under **Control Flow** tab.

<ol style="list-style-type: none"> 1. Make sure the Control Flow tab is selected. 2. Open the SSIS Toolbox 3. Drag a Data Flow Task from the SSIS Toolbox and drop it on the Control Flow design surface. 	
<ol style="list-style-type: none"> 4. Right click the Data Flow Task, select Rename to change the task name to Merge. 	

IV. Build Data Flow

The **Data Flow tab** is where you **specify the details** of any Data Flow task that you've added on the Control Flow tab

- **Data Flow Sources** are ways that data gets into the system.
- **Data Flow Transformations** allow you alter and manipulate the data in various ways.
- **Data Flow Destinations** are the places that you can send the transformed data.

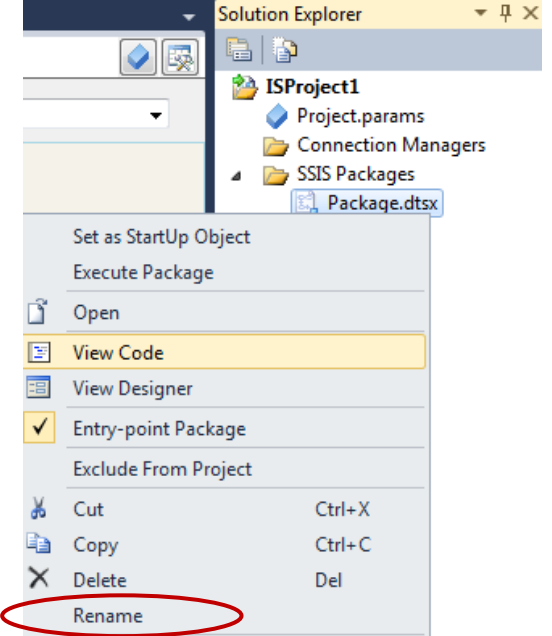
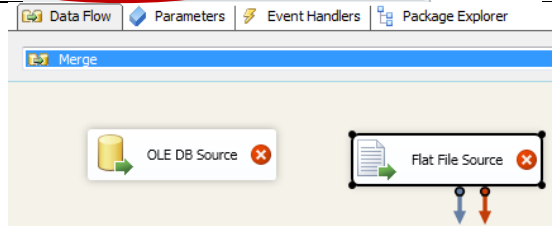
a) An Example of Transformations

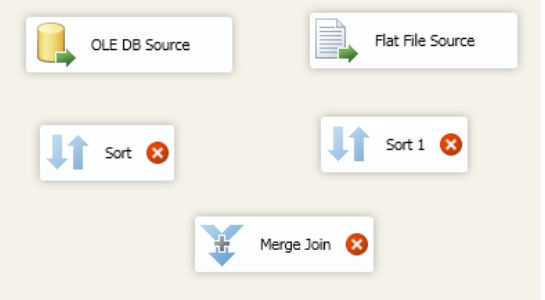
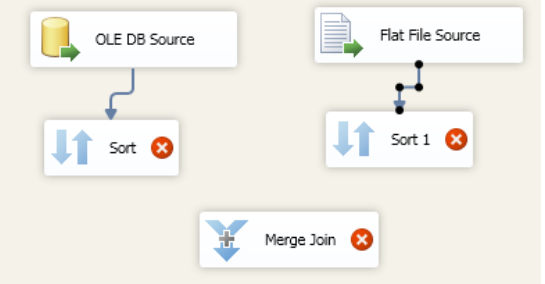
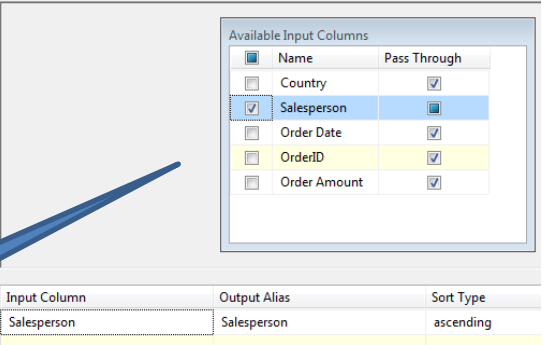
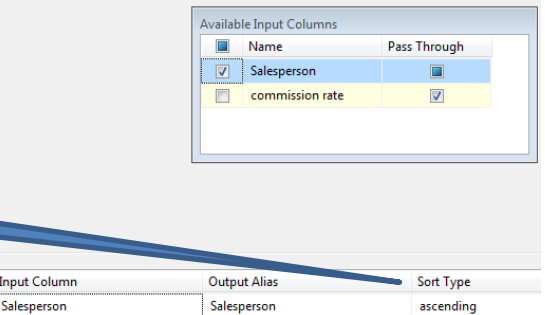
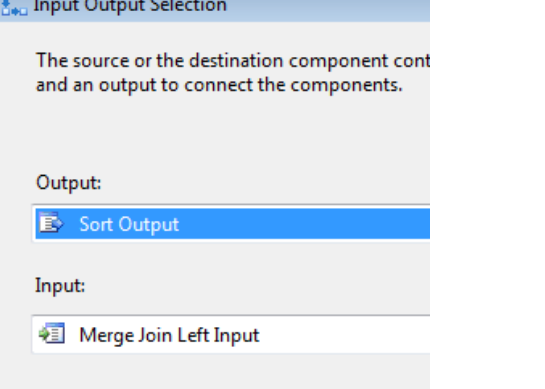
Data transformation is, in general, a process that takes input data and converts it into another form as output. Let's try **Merge Join** transformation first.

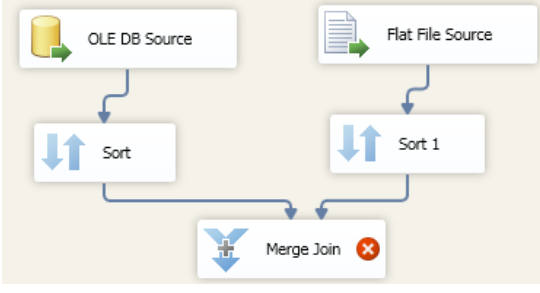
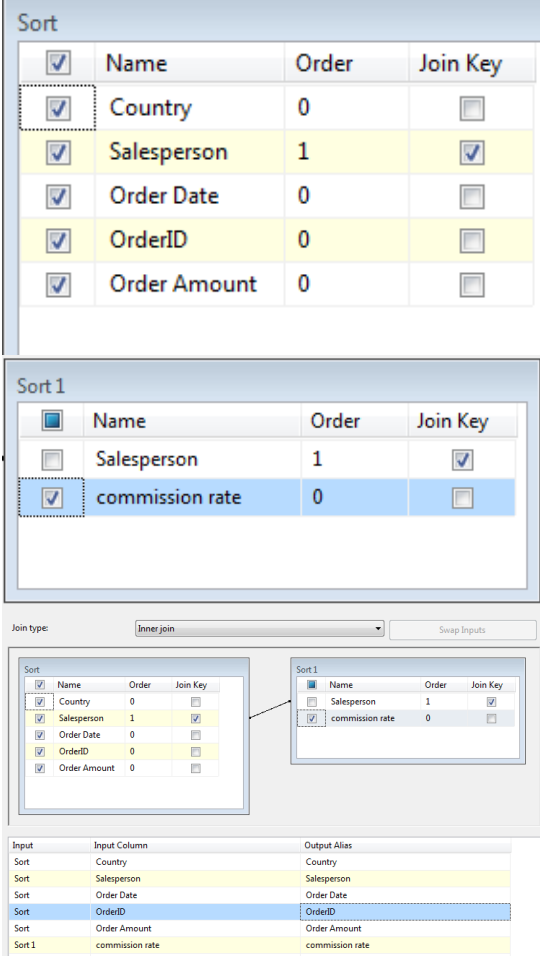
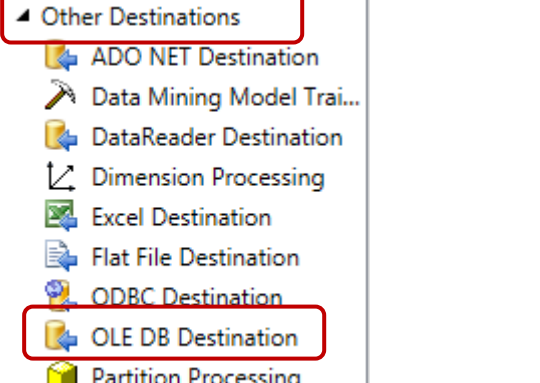
Merge Join is used to merge two sorted data sets into a single dataset by executing inner join, left outer join etc. Merge Join transformations require sorted data for their inputs. It means both data flows must be sorted by the columns to be used as the join condition. Merge join transformation combines data depending on matching keys or string name.

The tasks you need to perform are:

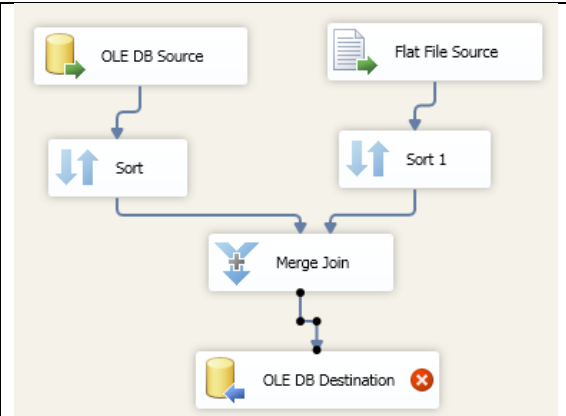
- 1) **Extracting** the data from tbl_sales table and the flat file (commission.txt)
- 2) **Transforming:** Sort the records by Salesperson field in both sources, merge the commission rate to the tbl_sales by using the common field (Salesperson)
- 3) **Loading** the transformed data to a new table named **tbl_merge**

<ol style="list-style-type: none"> 1. In Solution Explorer, rename the package Package.dtsx to Merge.dtsx by right clicking on it. 2. The result will be like this: 	
<ol style="list-style-type: none"> 3. Select the Data Flow tab in the Package Designer. 4. In the SSIS Toolbox, expand Other Sources, and then drag an OLE DB Source and a Flat File Source onto the design surface of the Data Flow tab. 5. Double click the OLE DB Source. Choose tbl_sales as the source table and press OK. 6. Double click the Flat File Source and select commission as the Flat file connection manager. Then press OK. 	 <p>OLE DB Source settings:</p> <p>OLE DB connection manager: localhost\SalesOrder</p> <p>Data access mode: Table or view</p> <p>Name of the table or the view: [dbo].[tbl_sales]</p> <p>Flat file Source settings:</p> <p>Connection Manager: commission</p> <p>Columns: </p> <p>Error Output: </p> <p><input type="checkbox"/> Retain null values from the source</p>

<p>7. Drag the following Transformations from SSIS toolbox (under Common section):</p> <ul style="list-style-type: none"> a) Sort b) Sort c) Merge Join 	
<p>8. Click on OLE DB Source, drag the blue arrow from its bottom and drop it on top of the first Sort Transformation.</p> <p>9. Click on Flat File Source, drag the blue arrow from its bottom and drop it on top of the second Sort Transformation.</p>	
<p>10. Double click the first Sort Transformation, Specify the column (Salesperson) to sort, and set the sort type (ascending). Then press OK.</p> <p>11. Double click the second Sort Transformation, Sort the records by Salesperson in ascending order. Then press OK.</p> <div style="text-align: center; margin-top: 20px;"> <div style="border: 1px solid blue; background-color: #4a7ebb; color: white; padding: 5px; display: inline-block; margin: 10px;">Salesperson</div> <div style="border: 1px solid blue; background-color: #4a7ebb; color: white; padding: 5px; display: inline-block; margin: 10px;">ascending</div> </div>	<p>First sort transformation:</p>  <p>Second sort transformation:</p> 
<p>12. Connect the first Sort Transformation with the Merge Join Transformation and select Merge Join Left Input for the input.</p> <p>13. Connect the second Sort Transformation with the Merge Join Transformation</p>	

																																																										
<p>14. Double click on the Merge Join Transformation, check <i>all the fields</i> in the first table (on the left), and <i>commission rate</i> field in the second table (on the right). Press OK.</p>	 <p>Sort</p> <table border="1"> <thead> <tr> <th><input checked="" type="checkbox"/></th> <th>Name</th> <th>Order</th> <th>Join Key</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td> <td>Country</td> <td>0</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>Salesperson</td> <td>1</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>Order Date</td> <td>0</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>OrderID</td> <td>0</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>Order Amount</td> <td>0</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>Sort 1</p> <table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Name</th> <th>Order</th> <th>Join Key</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Salesperson</td> <td>1</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>commission rate</td> <td>0</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>Join type: Inner join</p> <p>Input:</p> <table border="1"> <thead> <tr> <th>Input</th> <th>Input Column</th> <th>Output Alias</th> </tr> </thead> <tbody> <tr> <td>Sort</td> <td>Country</td> <td>Country</td> </tr> <tr> <td>Sort</td> <td>Salesperson</td> <td>Salesperson</td> </tr> <tr> <td>Sort</td> <td>Order Date</td> <td>Order Date</td> </tr> <tr> <td>Sort</td> <td>OrderID</td> <td>OrderID</td> </tr> <tr> <td>Sort</td> <td>Order Amount</td> <td>Order Amount</td> </tr> <tr> <td>Sort 1</td> <td>commission rate</td> <td>commission rate</td> </tr> </tbody> </table>	<input checked="" type="checkbox"/>	Name	Order	Join Key	<input checked="" type="checkbox"/>	Country	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Salesperson	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Order Date	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	OrderID	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Order Amount	0	<input type="checkbox"/>	<input type="checkbox"/>	Name	Order	Join Key	<input type="checkbox"/>	Salesperson	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	commission rate	0	<input type="checkbox"/>	Input	Input Column	Output Alias	Sort	Country	Country	Sort	Salesperson	Salesperson	Sort	Order Date	Order Date	Sort	OrderID	OrderID	Sort	Order Amount	Order Amount	Sort 1	commission rate	commission rate
<input checked="" type="checkbox"/>	Name	Order	Join Key																																																							
<input checked="" type="checkbox"/>	Country	0	<input type="checkbox"/>																																																							
<input checked="" type="checkbox"/>	Salesperson	1	<input checked="" type="checkbox"/>																																																							
<input checked="" type="checkbox"/>	Order Date	0	<input type="checkbox"/>																																																							
<input checked="" type="checkbox"/>	OrderID	0	<input type="checkbox"/>																																																							
<input checked="" type="checkbox"/>	Order Amount	0	<input type="checkbox"/>																																																							
<input type="checkbox"/>	Name	Order	Join Key																																																							
<input type="checkbox"/>	Salesperson	1	<input checked="" type="checkbox"/>																																																							
<input checked="" type="checkbox"/>	commission rate	0	<input type="checkbox"/>																																																							
Input	Input Column	Output Alias																																																								
Sort	Country	Country																																																								
Sort	Salesperson	Salesperson																																																								
Sort	Order Date	Order Date																																																								
Sort	OrderID	OrderID																																																								
Sort	Order Amount	Order Amount																																																								
Sort 1	commission rate	commission rate																																																								
<p>15. In the SSIS Toolbox, expand Other Destinations, and drag an OLE DB Destination onto the design surface.</p>	 <p>Other Destinations</p> <ul style="list-style-type: none"> ADO NET Destination Data Mining Model Trai... DataReader Destination Dimension Processing Excel Destination Flat File Destination ODBC Destination OLE DB Destination Partition Processing 																																																									

16. Click on the **Merge Join** Transformation, drag the **blue arrow** from its bottom and drop it on top of the **OLE DB Destination**.



17. **Double click** on the **OLE DB Destination**, create a new table to store the merged data, press **New** button for the option “Name of the table or the view”.

18. **Rename** the new table to **tbl_merge**. Then press **OK**.

19. Select the **Mappings** page of the dialog box (on the left hand side) to make sure that the mapping is correct.

20. Press **OK** to continue.

21. Select **File** → **Save All** to save the Package.

OLE DB connection manager:
 localhost\SalesOrder
 Data access mode:
 Table or view - fast load
 Name of the table or the view:
 New...
☐ Keep identity ☒ Table lock
☐ Keep nulls ☒ Check constraints
 Rows per batch:
 Maximum insert commit size: 2147483647

tbl_merge

Create Table

```
CREATE TABLE [tbl_merge] (
  [Country] nvarchar(255),
  [Salesperson] nvarchar(255),
  [Order Date] datetime,
  [OrderID] float,
  [Order Amount] float,
  [commission rate] varchar(50)
)
```

Connection Manager
 Mappings
 Error Output

Available Input Columns

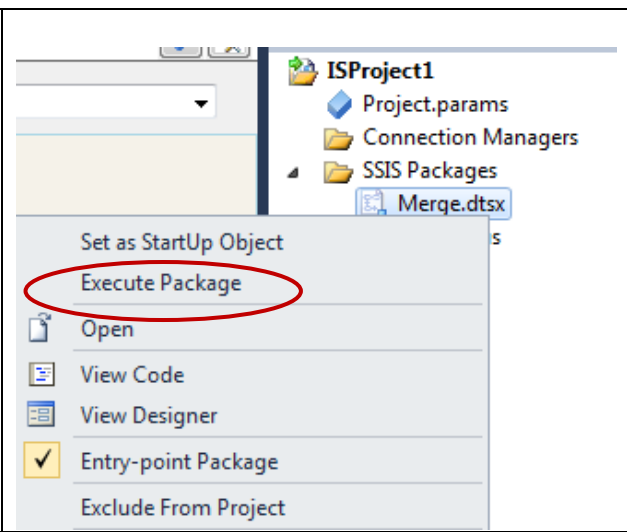
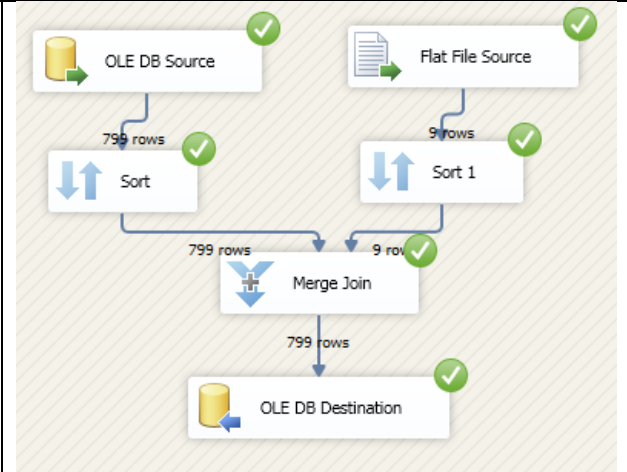
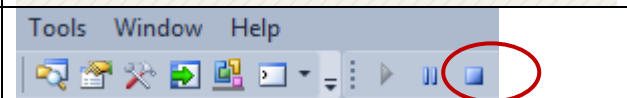
Name
Country
Salesperson
Order Date
OrderID
Order Amount
commission rate

Input Column

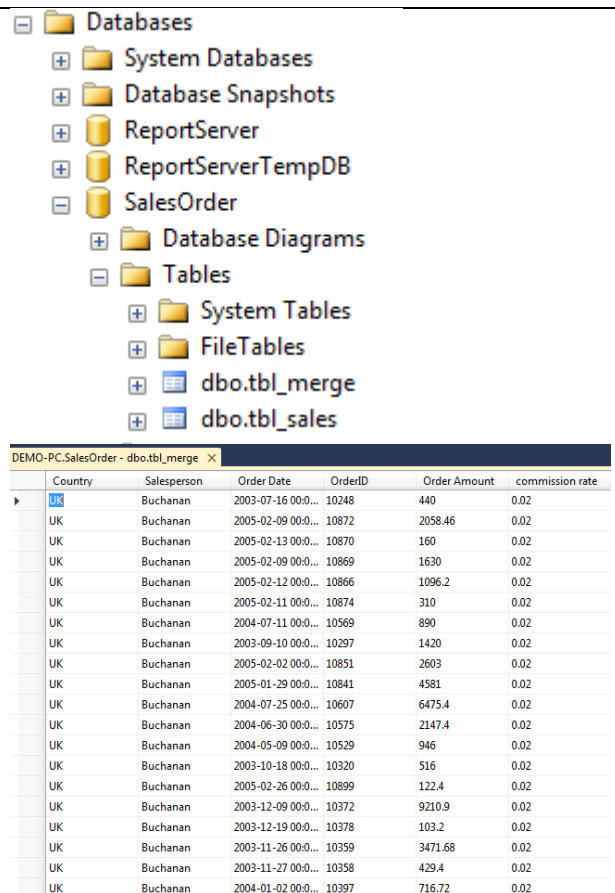
Country
Salesperson
Order Date
OrderID
Order Amount
commission rate

V. Running a package

You can run the final package to perform the merge join.

<p>1. Right click the package Merge.dtsx in Solution Explorer and select Execute Package.</p>	 <p>The screenshot shows the 'Solution Explorer' window with the project 'ISProject1' expanded. Under 'SSIS Packages', the file 'Merge.dtsx' is selected. A right-click context menu is open, and the 'Execute Package' option is highlighted with a red circle. Other options in the menu include 'Set as StartUp Object', 'Open', 'View Code', 'View Designer', 'Entry-point Package' (checked), and 'Exclude From Project'.</p>
<p>2. The result will be similar to:</p>	 <p>The screenshot displays the 'Data Flow Task' diagram. It starts with two sources: 'OLE DB Source' (799 rows) and 'Flat File Source' (9 rows). Both sources feed into 'Sort' tasks ('Sort' and 'Sort 1'). The outputs of the sort tasks then feed into a 'Merge Join' task. The final output of the 'Merge Join' task is an 'OLE DB Destination' (799 rows). All tasks in the diagram have a green checkmark in the top right corner, indicating successful execution.</p>
<p>3. Click Stop Debugging button.</p>	 <p>The screenshot shows the 'Tools' window with the 'Stop Debugging' button circled in red. The button is represented by a blue square icon with a white 'X' inside.</p>

4. Open SSMS, a new table **tbl_merge** is created. It contains merged data with the field commission rate.



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SalesOrder' database structure, including 'Database Diagrams', 'Tables', 'System Tables', 'FileTables', 'dbo.tbl_merge', and 'dbo.tbl_sales'. The right pane shows a data table with the following columns: Country, Salesperson, Order Date, OrderID, Order Amount, and commission rate. The table contains 20 rows of data, all with 'UK' as the country and 'Buchanan' as the salesperson.

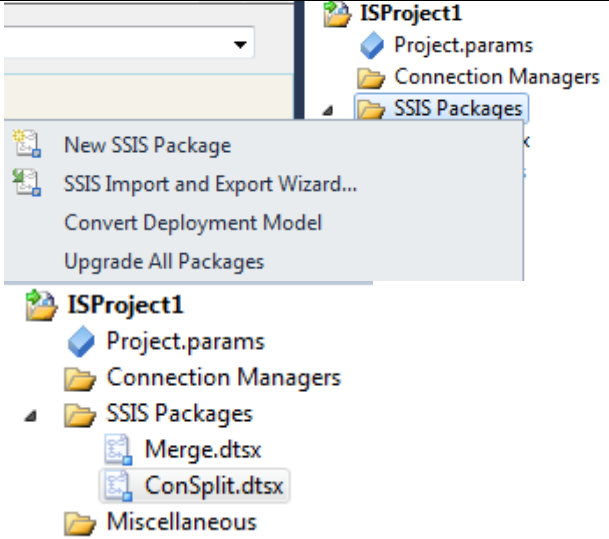
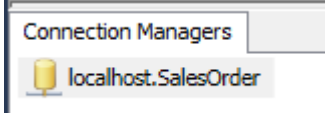
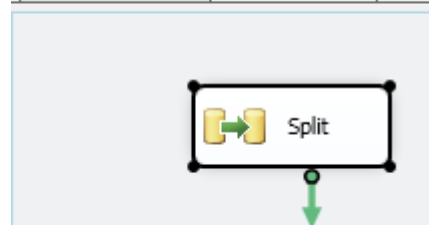
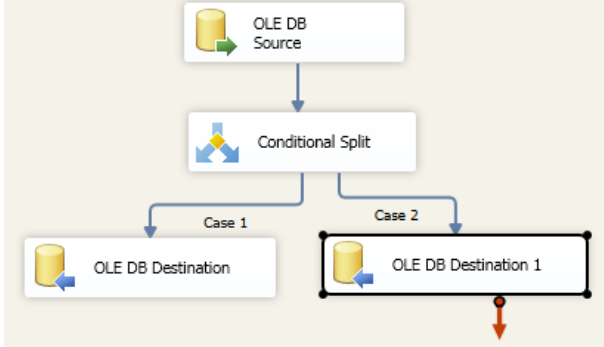
Country	Salesperson	Order Date	OrderID	Order Amount	commission rate
UK	Buchanan	2003-07-16 00:00:00	10248	440	0.02
UK	Buchanan	2005-02-09 00:00:00	10872	2058.46	0.02
UK	Buchanan	2005-02-13 00:00:00	10870	160	0.02
UK	Buchanan	2005-02-09 00:00:00	10869	1630	0.02
UK	Buchanan	2005-02-12 00:00:00	10866	1096.2	0.02
UK	Buchanan	2005-02-11 00:00:00	10874	310	0.02
UK	Buchanan	2004-07-11 00:00:00	10569	890	0.02
UK	Buchanan	2003-09-10 00:00:00	10297	1420	0.02
UK	Buchanan	2005-02-02 00:00:00	10851	2603	0.02
UK	Buchanan	2005-01-29 00:00:00	10841	4581	0.02
UK	Buchanan	2004-07-25 00:00:00	10607	6475.4	0.02
UK	Buchanan	2004-06-30 00:00:00	10575	2147.4	0.02
UK	Buchanan	2004-05-09 00:00:00	10529	946	0.02
UK	Buchanan	2003-10-18 00:00:00	10320	516	0.02
UK	Buchanan	2005-02-26 00:00:00	10899	122.4	0.02
UK	Buchanan	2003-12-09 00:00:00	10372	9210.9	0.02
UK	Buchanan	2003-12-19 00:00:00	10378	103.2	0.02
UK	Buchanan	2003-11-26 00:00:00	10359	3471.68	0.02
UK	Buchanan	2003-11-27 00:00:00	10358	429.4	0.02
UK	Buchanan	2004-01-02 00:00:00	10397	716.72	0.02

5. Open the **Visual Studio with SSDT** program, Select **File → Save All** to save the **Merge.dtsx**.

VI. Other Transformations

a) Conditional Split

Conditional Split can send data rows to different outputs or branches depending on the criteria/condition you have given. For example, You can split the sales order records based on different countries.

<p>1. Right click SSIS Packages in Solution Explorer, select New SSIS Package to create a new package and rename it to ConSplit.dtsx.</p>	
<p>2. Add an OLE DB Connection. (localhost.SalesOrder)</p>	
<p>3. Add a Data Flow Task in Control Flow tab and rename it to Split</p>	
<p>4. Go to Data Flow tab, perform the tasks below and connect the tasks as shown here.</p>	

- a) Define a **OLE DB Source** to extract the data from **tbl_sales** table
 - b) Add **Conditional Split** Transformations to split the records based on different countries

Case 1: Country == "UK"
Case 2: Country == "USA"
 [If the expression is invalid, the expression text will be highlighted in red color]
 - c) To add split data to database, we need to add two **OLE DB Destinations**
 - d) Connect the Conditional Split transformation with the first destination and change Output to **Case 1**. Repeat the step for another destination by choosing **Case 2** as Output.
 - e) Edit the two OLE DB destinations and create two new tables named **tbl_UK** and **tbl_USA** respectively. (make sure that the mappings are correct)
5. **Execute** the package **ConSplit.dtsx**. Use SSMS to verify the results. There will be
 - 215 records in **tbl_UK**
 - 584 records in **tbl_USA**
 6. **Stop** running the package.
 7. Select **File → Save All** to save the package. Close the package.

tbl_UK

OLE DB Source settings:

OLE DB connection manager:
localhost.SalesOrder

Data access mode:
Table or view

Name of the table or the view:
[dbo].[tbl_sales]

Conditional Split transformation settings:

Order	Output Name	Condition
1	Case 1	[Country]="UK"
2	Case 2	[Country]="USA"

Input Output Selection dialog box setting:

Input Output Selection

The source or the destination component contains multiple inputs or outputs and an output to connect the components.

Output:
Case 1

Input:
OLE DB Destination 1

Case 1

OK

OLE DB Destination: (tbl_UK)

OLE DB connection manager:
localhost

Data access mode:
Table or view - fast load

Name of the table or the view:
[tbl_UK]

New...

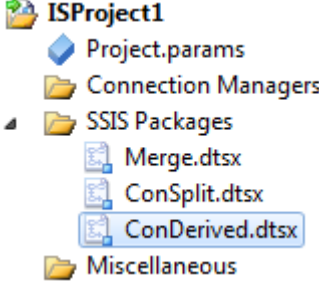
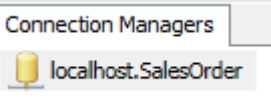
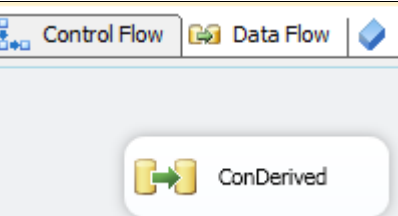
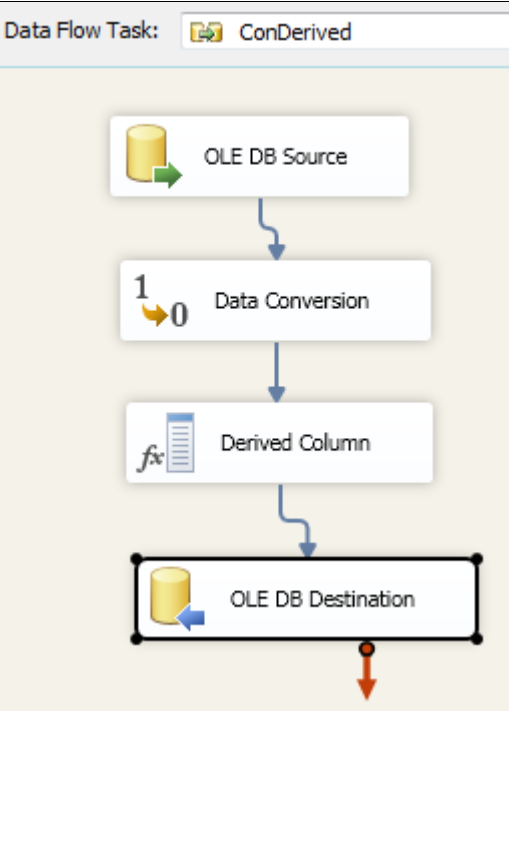
Create Table

```
CREATE TABLE [tbl_UK] (
  [Country] nvarchar(255),
  [Salesperson] nvarchar(255),
  [Order Date] datetime,
  [OrderID] float,
  [Order Amount] float
)
```


b) Data Conversion and Derived Column

Data conversion transformation enables us to convert columns from one data type to another. The converted data can either replace the existing column or be added as a new column.

Derived Column transformation creates a new column that is derived from the output of another column. This transformation provides you two options; either you can create a new column as a derived column or replace the existing column with a new derived column

1. Create a new package using Visual Studio with SSDT and rename it to ConDerived.dtsx in Solution Explorer .	
2. Add an OLE DB Connection . (localhost.SalesOrder)	
3. Add a Data Flow Task in Control Flow tab and rename it to ConDerived	
4. Go to Data Flow tab , perform the tasks below and connect the tasks like this:	

- a) Define a **OLB DB Source** to extract the data from **tbl_merge** table (with commission rate column) ***

OLD DB Source settings:

Name of the table or the view:

[dbo].[tbl_merge]

- b) Add a **Data Conversion** Transformation, select the field **commission rate**. Change **Output alias** to "Numeric Commission rate", and change the data type to **double-precision float [DT_R8]**

Data Conversion transformation settings:

Available Input Columns

☒ Name
☐ Country
☐ Salesperson
☐ Order Date
☐ OrderID
☐ Order Amount
☒ commission r...

Numeric
commission rate

Input Column	Output Alias	Data Type
commission rate	Numeric commission rate	double-precision float [DT_R8]

- c) Add a **Derived Column** Transformation, create a new derived column named **Commission Amount**, the expression is:

Derived Column transformation settings:

[Numeric commission rate] * [Order Amount]

Derived Column Name	Derived Column
Commission Amount	<add as new column>

[If the expression is invalid, the expression text will be highlighted in red]

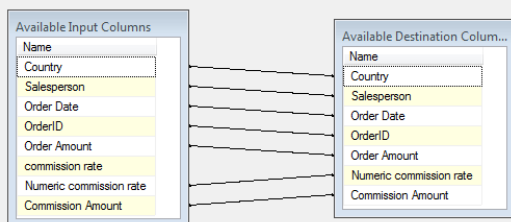
Expression
[Numeric commission rate] * [Order Amount]

- d) Load the output to a new table **tbl_conDerived** using **OLB DB Destination**, **remove the field [commission rate] varchar(50)** and keep the numeric commission rate.
(make sure that the mapping is correct)

OLD DB Destination: (tbl_conDerived)

Create Table

```
CREATE TABLE [tbl_conDerived] (
  [Country] nvarchar(255),
  [Salesperson] nvarchar(255),
  [Order Date] datetime,
  [OrderID] float,
  [Order Amount] float,
  [Numeric commission rate] float,
  [Commission Amount] float
)
```



Commission
Rate is removed

5. **Run** the package **ConDerived.dtsx**. Then **Stop** running the package.

The result is SSMS:

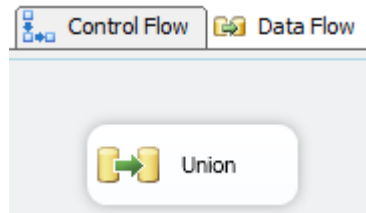
6. Use SSMS to verify the results. There will be 799 records in **tbl_conDerived** table.
- There are 7 columns
 - A calculated field "Commission Amount" is shown
7. Select **File** → **Save All** to save the package **ConDerived.dtsx**. Then **close** it.

	Country	Salesperson	Order Date	OrderID	Order Amo...	Numeric commission rate	Commission Amount
1	UK	Buchanan	2003-07-16 ...	10248	440	0.02	8.8
2	UK	Buchanan	2005-02-09 ...	10872	2058.46	0.02	41.1692
3	UK	Buchanan	2005-02-13 ...	10870	160	0.02	3.2
4	UK	Buchanan	2005-02-09 ...	10869	1630	0.02	32.6
5	UK	Buchanan	2005-02-12 ...	10866	1096.2	0.02	21.924
6	UK	Buchanan	2005-02-11 ...	10874	310	0.02	6.2
7	UK	Buchanan	2004-07-11 ...	10569	890	0.02	17.8
8	UK	Buchanan	2003-09-10 ...	10297	1420	0.02	28.4
9	UK	Buchanan	2005-02-02 ...	10851	2603	0.02	52.06
10	UK	Buchanan	2005-01-29 ...	10841	4581	0.02	91.62
11	UK	Buchanan	2004-07-25 ...	10607	6475.4	0.02	129.508
12	UK	Buchanan	2004-06-30 ...	10575	2147.4	0.02	42.948
13	UK	Buchanan	2004-05-09 ...	10529	946	0.02	18.92
14	UK	Buchanan	2003-10-18 ...	10320	516	0.02	10.32
15	UK	Buchanan	2005-02-26 ...	10899	122.4	0.02	2.448

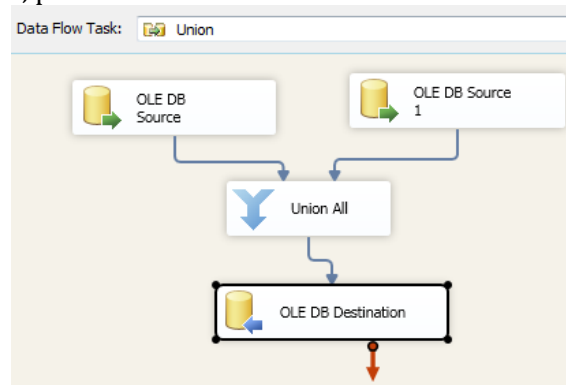
VII. Exercise 1

Union All allows you to combine *multiple* inputs and produce one output. Its add inputs to transformation output one after the other and doesn't sort the data nor remove duplicates.

1. Create a new package and rename it to **Union.dtsx**.
2. Add an **OLE DB Connection**. (localhost.SalesOrder)
3. Add a **Data Flow Task** in **Control Flow** tab and rename it to **Union**



4. Go to **Data Flow** tab, perform the tasks below and connect the tasks like this:



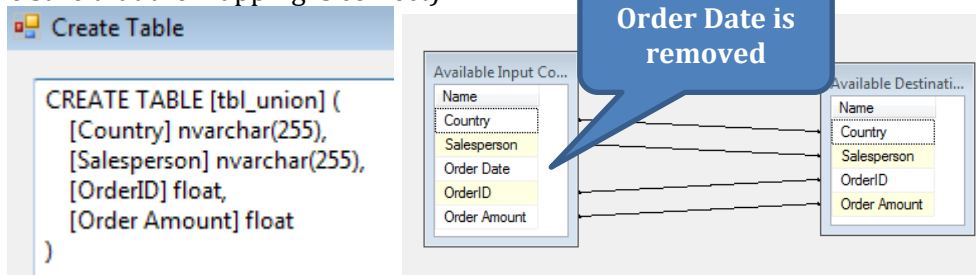
5. Add two **OLE DB Sources** to extract the data from **tbl_UK** and **tbl_USA** tables respectively.
6. Add **Union All** Transformation to combine the splitted records into one.

Union All Transformation Editor

Configure the properties used to merge multiple inputs into one output by creating mappings between columns.

Output Column Name	Union All Input 1	Union All Input 2
Country	Country	Country
Salesperson	Salesperson	Salesperson
Order Date	Order Date	Order Date
OrderID	OrderID	OrderID
Order Amount	Order Amount	Order Amount

7. Add an **OLE DB Destination**, connect the **Union All** transformation with the OLE DB Destination.
8. Add a new table **tbl_union** to store the transformed data, **remove** the **Order Date** field. (make sure that the mapping is correct)



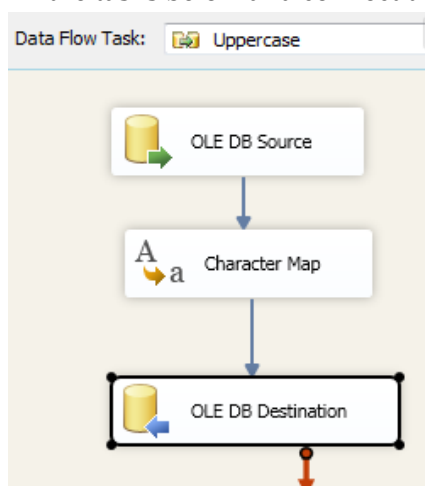
9. **Run** the package **Union.dtsx**. Then **Stop** running the package.
10. Use SSMS to verify the results. There will be 799 records in **tbl_union** (with 4 columns)

	Country	Salesperson	OrderID	Order Amount
1	USA	Peacock	10250	1552.6
2	USA	Leverling	10251	654.06
3	USA	Peacock	10252	3597.9
4	USA	Leverling	10253	1444.8
5	USA	Leverling	10256	517.8
6	USA	Peacock	10257	1119.9
7	USA	Davolio	10258	1614.88
8	USA	Peacock	10259	100.8
9	USA	Peacock	10260	1504.65
10	USA	Peacock	10261	448
11	USA	Callahan	10262	584
12	USA	Fuller	10265	1176

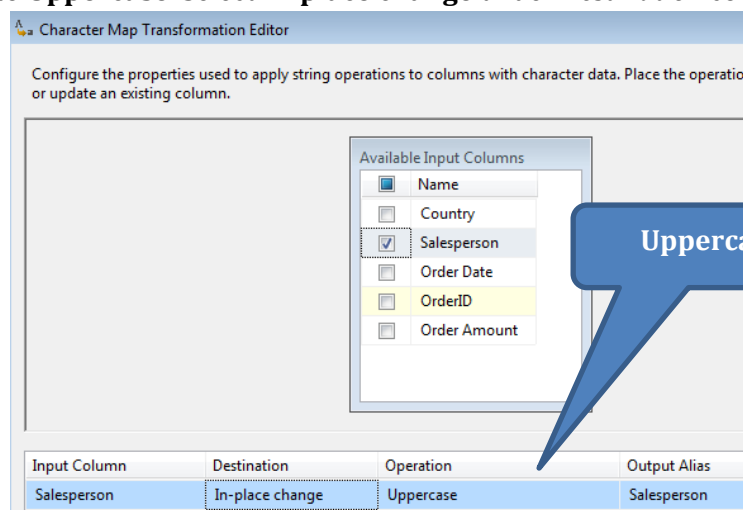
11. Select **File** → **Save All** to save the package. Then close the package.

VIII. Exercise 2

1. Create a new package and rename it to **Uppercase.dtsx** in **Solution Explorer**.
2. Add an **OLE DB Connection**. (localhost.SalesOrder)
3. Add a **Data Flow Task** in **Control Flow** tab and rename the task to **Uppercase**.
4. Go to **Data Flow** tab, perform the tasks below and connect the tasks like this:



5. Define a **OLE DB Source** to extract the data from **tbl_sales**
6. Add a **Character Map** Transformation (Under **Other Transforms** section) to change the **Salesperson** to **Uppercase**. Select **In-place change** under *Destination* column.



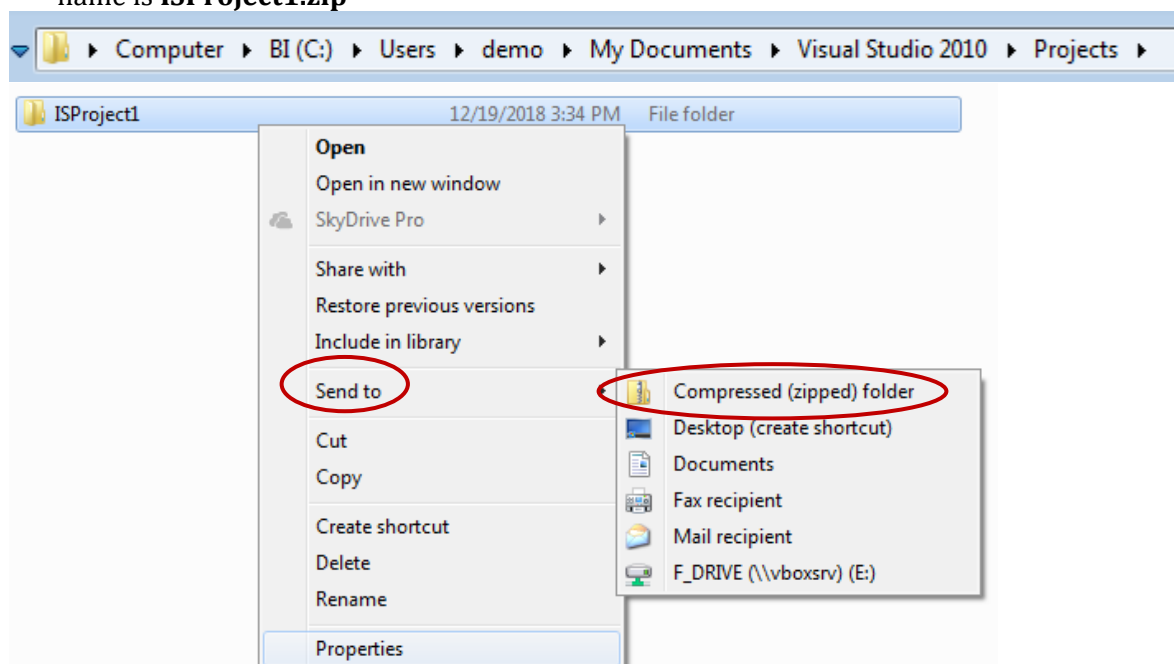
7. Load the output to an **OLE DB destination** and create a new table named **tbl_uppercase** (make sure that the mappings are correct) Note: If you do not create a new table to store the transformed data, the data will be appended in the same table tbl_sales and the no. of records will be doubled.
8. **Run** the package **Uppercase.dtsx**. **Stop** running the package
9. Use SSMS to verify the results. There will be 799 records in **tbl_uppercase**

	Country	Salesperson	Order Date	OrderID	Order Amount
1	UK	BUCHANAN	2003-07-16 00:00:00.000	10248	440
2	UK	SUYAMA	2003-07-10 00:00:00.000	10249	1863.4
3	USA	PEACOCK	2003-07-12 00:00:00.000	10250	1552.6
4	USA	LEVERLING	2003-07-15 00:00:00.000	10251	654.06
5	USA	PEACOCK	2003-07-11 00:00:00.000	10252	3597.9
6	USA	LEVERLING	2003-07-16 00:00:00.000	10253	1444.8

10. Select **File → Save All** to save the package **Uppercase.dtsx**. Close the **Visual Studio**.

IX. Answer Submission

1. **Zip** your Integration Services projects (**ISProject1** folder with **ISProject1.sln**) that you created in C:\Users\demo\Documents\Visual Studio 2010\Projects. The default file name is **ISProject1.zip**



2. Submit the zip file **ISProject1.zip** to the site <http://buelearning.hkbu.edu.hk/>