Jaihind Comprehensive Educational Institute`s

JAIHIND COLLEGE OF ENGINEERING

Tal- Junnar, Dist- Pune, Pin Code -410511

DTE Code: EN6609, SPPU Code: CEGP015730

Computer Laboratory IV [417535]

LABORATORY MANUAL

DEPARMTENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE ENGINEERING

Institute Vision

• To enrich the role of nation building by imparting the qualitative technical education.

Institute Mission

• Impart technical knowledge through prescribed curriculum of university.

• Inculcate ethical and moral values in students for environmental and sustainable development.

• Equip the aspirants through co-curricular and extra- curricular activities to excel in career.

Department Vision

• To provide current technical education and train competitive engineering professionals in Artificial Intelligence and Data Science with a commitment to fulfilling industry requirements.

Department Mission

• To foster students with latest technologies in the field of Artificial Intelligence and Data Science.

• To provide skill-based education to master the students in problem solving and analytical skills in the area of Artificial Intelligence and Data Science.

• To develop employability skills among students in the fields of Artificial Intelligence, Data Science.

Program Educational Objectives (PEOs)

PEO1 work in the domain of Artificial Intelligence and Data Science to design ability of a computer system.

PEO2 apply analytical skills, decision making skills, leadership skills and critical thinking skills to develop Artificial Intelligence and Data Science based solutions for business problems.

PEO3 demonstrate proficiency in applying Artificial Intelligence and Data Science methodologies to solve complex real world problems across various domains such as statistics, machine learning, data analytics and Artificial Intelligence.

Program Outcomes (POs)

PO1 Engineering knowledge

Apply the knowledge of mathematics, science, Engineering fundamentals, and an Engineering specialization to the solution of complex Engineering problems.

PO2 Problem analysis

Identify, formulate, review research literature and analyze complex Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.

PO3 Design / Development of Solutions

Design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and Environmental considerations.

PO4 Conduct Investigations of Complex Problems

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 Modern Tool Usage

Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations.

PO6 The Engineer and Society

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practices.

PO7 Environment and Sustainability

Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 Ethics

Apply ethical principles and commit to professional ethics and responsibilities and norms of Engineering practice.

PO9 Individual and Team Work

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 Communication Skills

Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 Project Management and Finance

Demonstrate knowledge and understanding of Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.

PO12 Life-long Learning

Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Computer Laboratory – IV

Course Code	Course Name	Teaching Scheme(Hrs./Week)	Credits
417535	Computer Laboratory IV Buisness Intelligence [417532B]	2	2

Course Objectives:

- To introduce the concepts and components of Buisness Intelligence (BI)

Course Outcomes:

After completion of the course, learners should be able to-

CO1: Apply basic principles of elective subjects to problem solving and modeling

CO2: Use tools and techniques in area of software development to build mini projects

CO3: Design and develop applications on subjects of their choice

CO4: Implement and manage deployment, administration & security

Table Of Contents

Lab Assignment No.	01
Title	Import Data from different Sources such as (Excel, Sql Server, Oracle etc.) and load in targeted system.
Roll No.	
Class	BE AI & DS
Date Of Completion	
Subject	Computer Laboratory IV[417535]
Assessment Marks	
Assessor's Sign	

Assignment No. 01

Title : Import the legacy data from different sources such as (Excel, Sql Server, Oracle etc.) and load in the target system. (You can download sample databases such as Adventure works,Northwind, foodmart etc.)

Objective:

- To introduce the concepts and components of Business Intelligence (BI)
- To understand how to **i**mport the legacy data from different sources and load in the target system.
- To Understand concepts of legacy system.

Outcomes:

- Students will be able to explain concepts and components of Business Intelligence (BI)
- Students will be able to **i**mport the legacy data from different sources and load in the target system.
- Students will be able to understand concepts of legacy system.

Software & Hardware Requirement:

- 64-bit Open source Linux or its derivative
- Additional tools like powerBI, SQL Server

Prerequisite:

1. Basics of dataset extensions.
2. Concept of data import.

Theory :

01. What is Legacy Data?

Legacy data, according to Business Dictionary, is "information maintained in an old or out-of-date format or computer system that is consequently challenging to access or handle."

02. Sources of Legacy Data

Where does legacy data come from? Virtually everywhere. Figure 1 indicates that there are many sources from which you may obtain legacy data. This includes existing databases, often relational, although non-RDBs such as hierarchical, network, object, XML, object/relational databases, and NoSQL databases. Files,

such as XML documents or "flat files" such as configuration files and comma-delimited text files, are also common sources of legacy data. Software, including legacy applications that have been wrapped (perhaps via CORBA) and legacy services such as web services or CICS transactions, can also provide access to existing information. The point to be made is that there is often far more to gaining access to legacy data than simply writing an SQL query against an existing relational database.

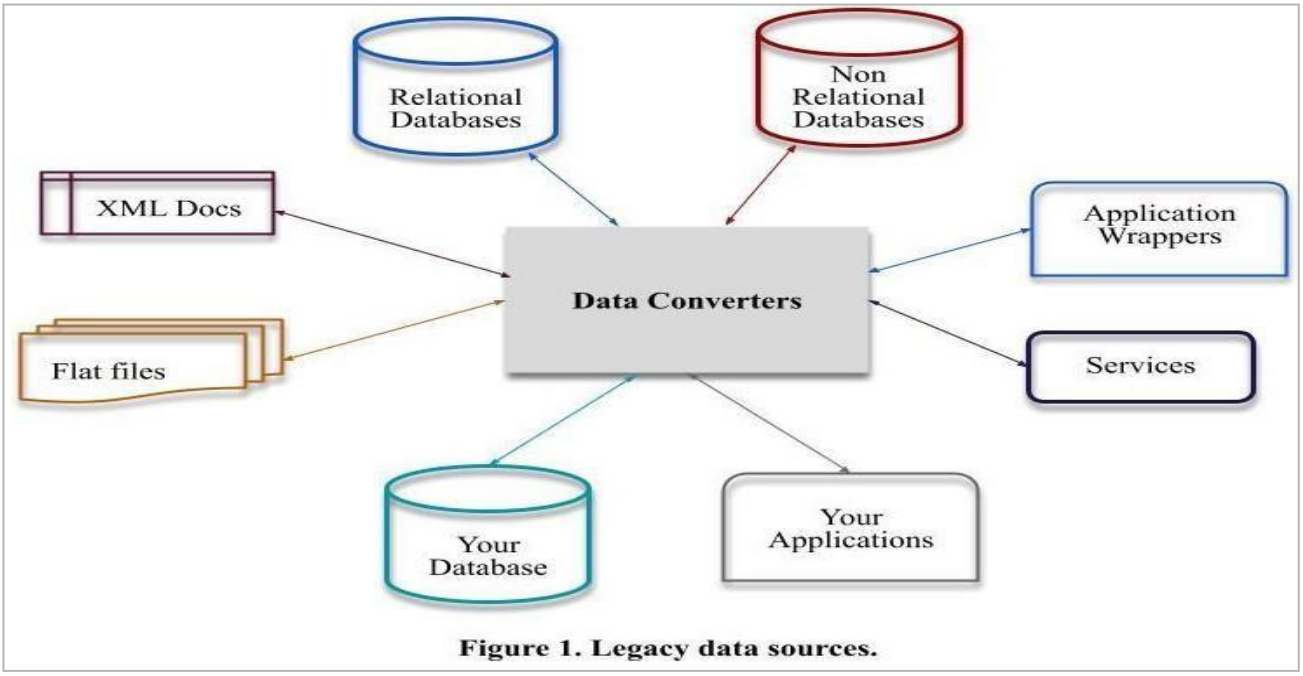
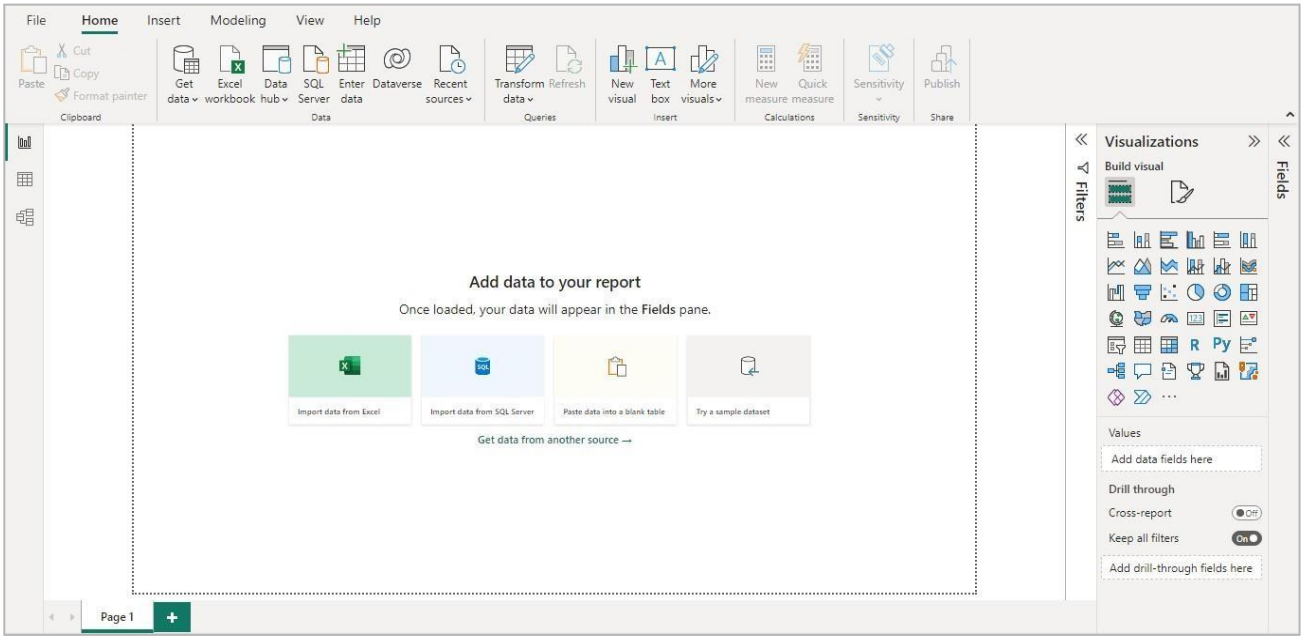


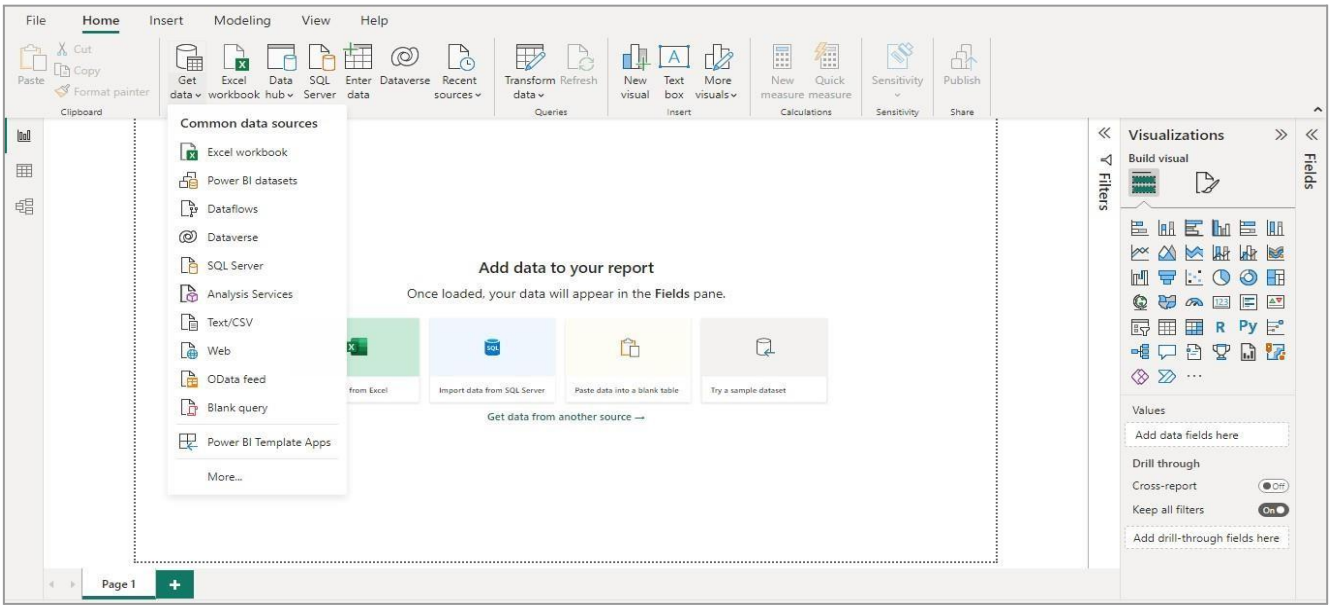
Figure 1. Legacy data sources.

03. How to import legacy data step by step.

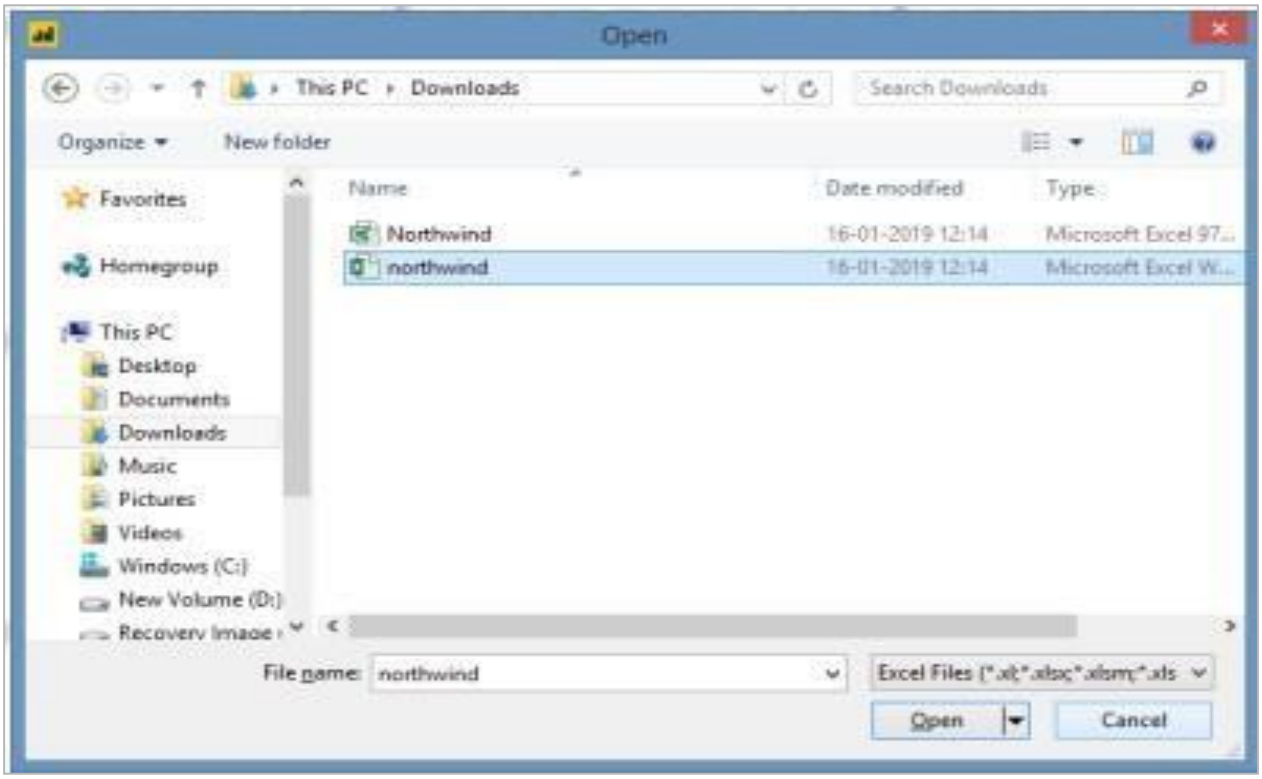
Step 1: Open Power BI



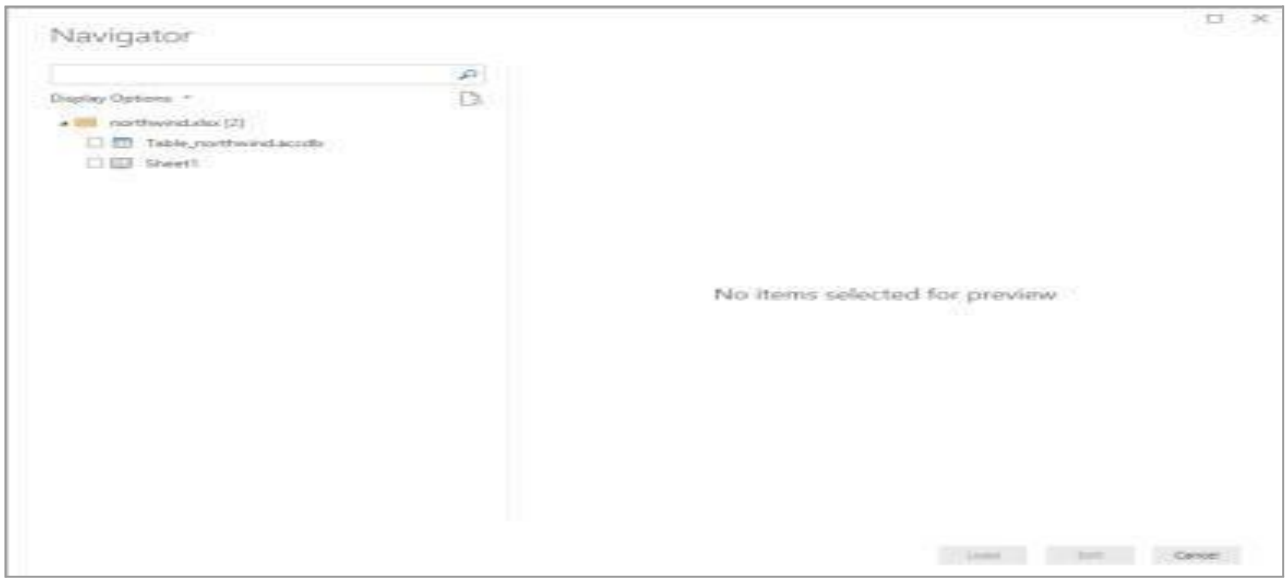
Step 2 : Click on Get data following list will be displayed → select Excel

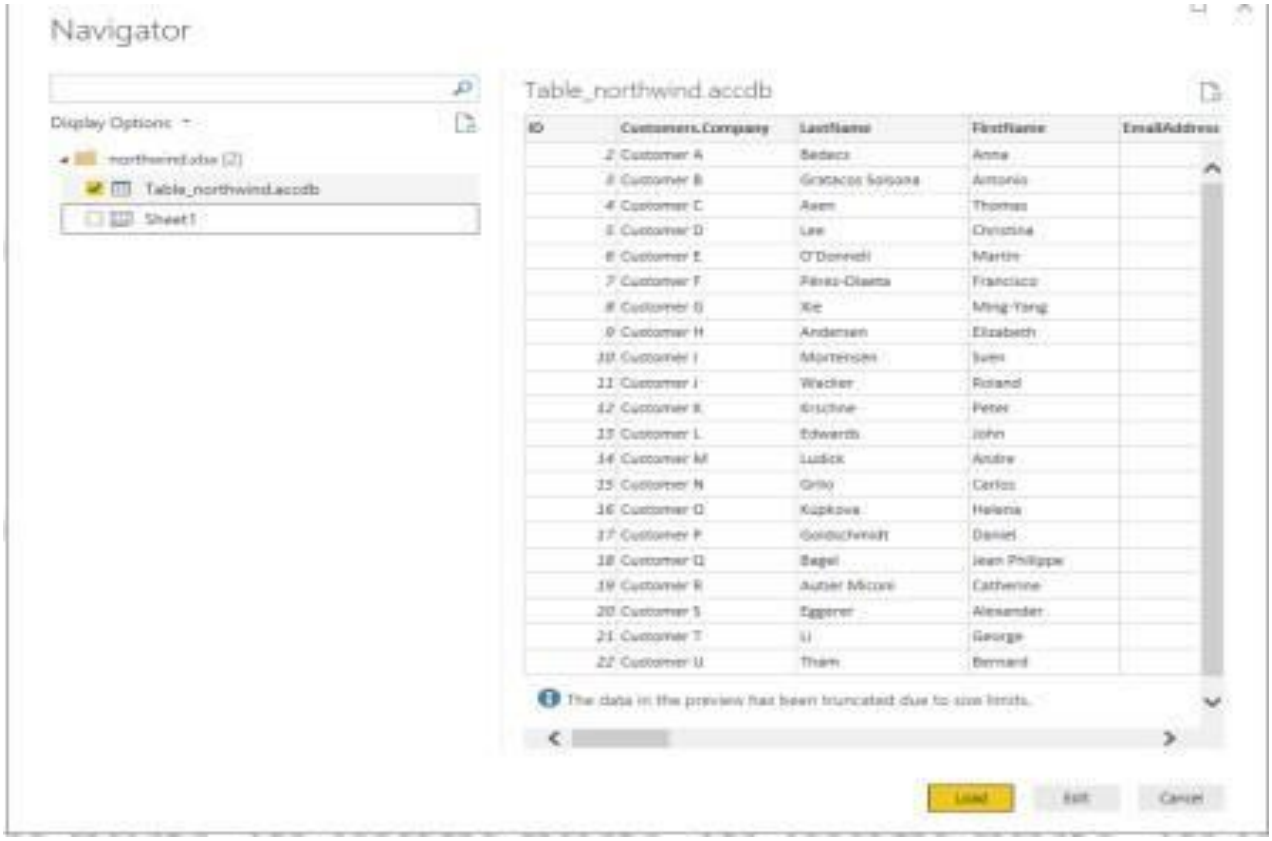


Step 3: Select required file and click on Open, Navigator screen appears

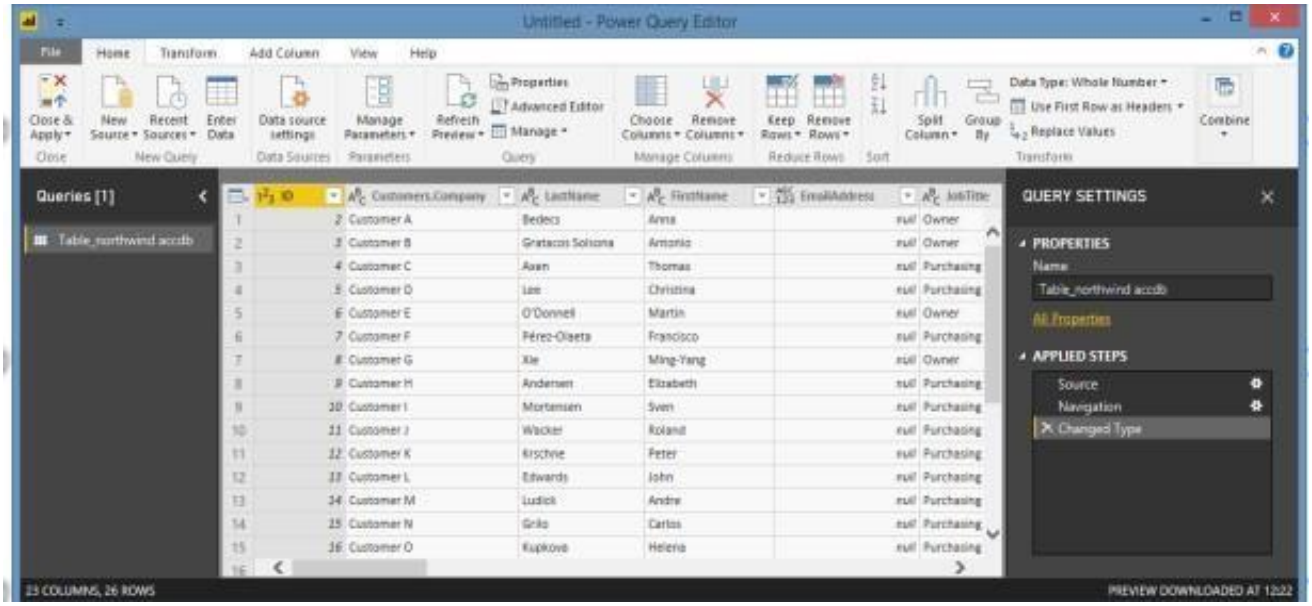


Step 4: Select file and click on edit

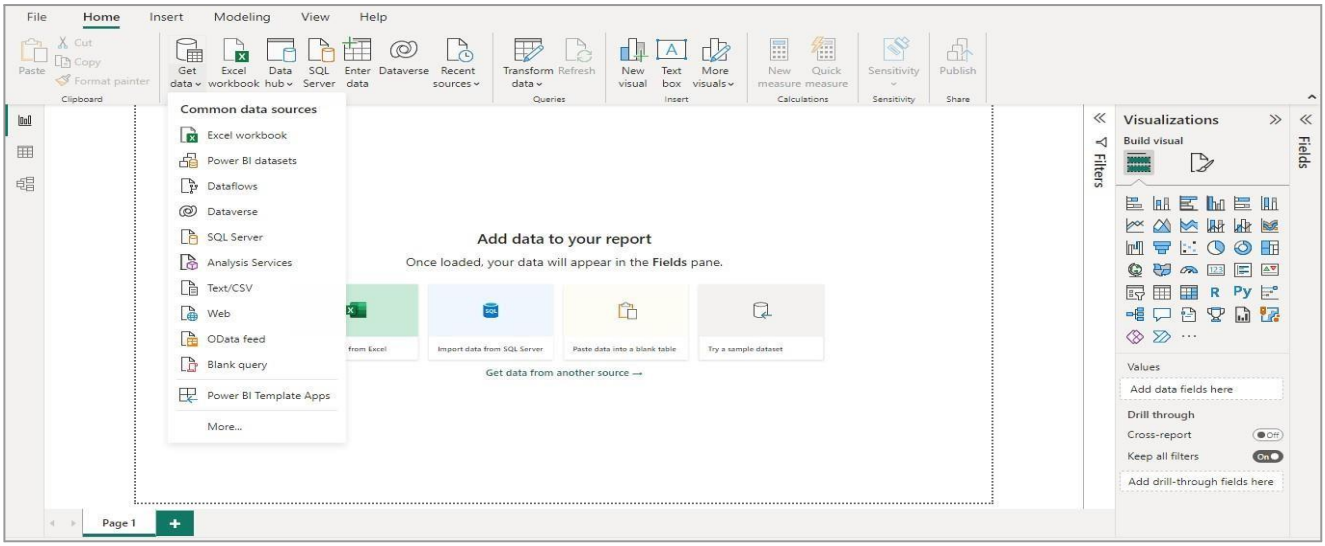




Step 5: Power query editor appears



Step 6: Again, go to Get Data and select OData feed

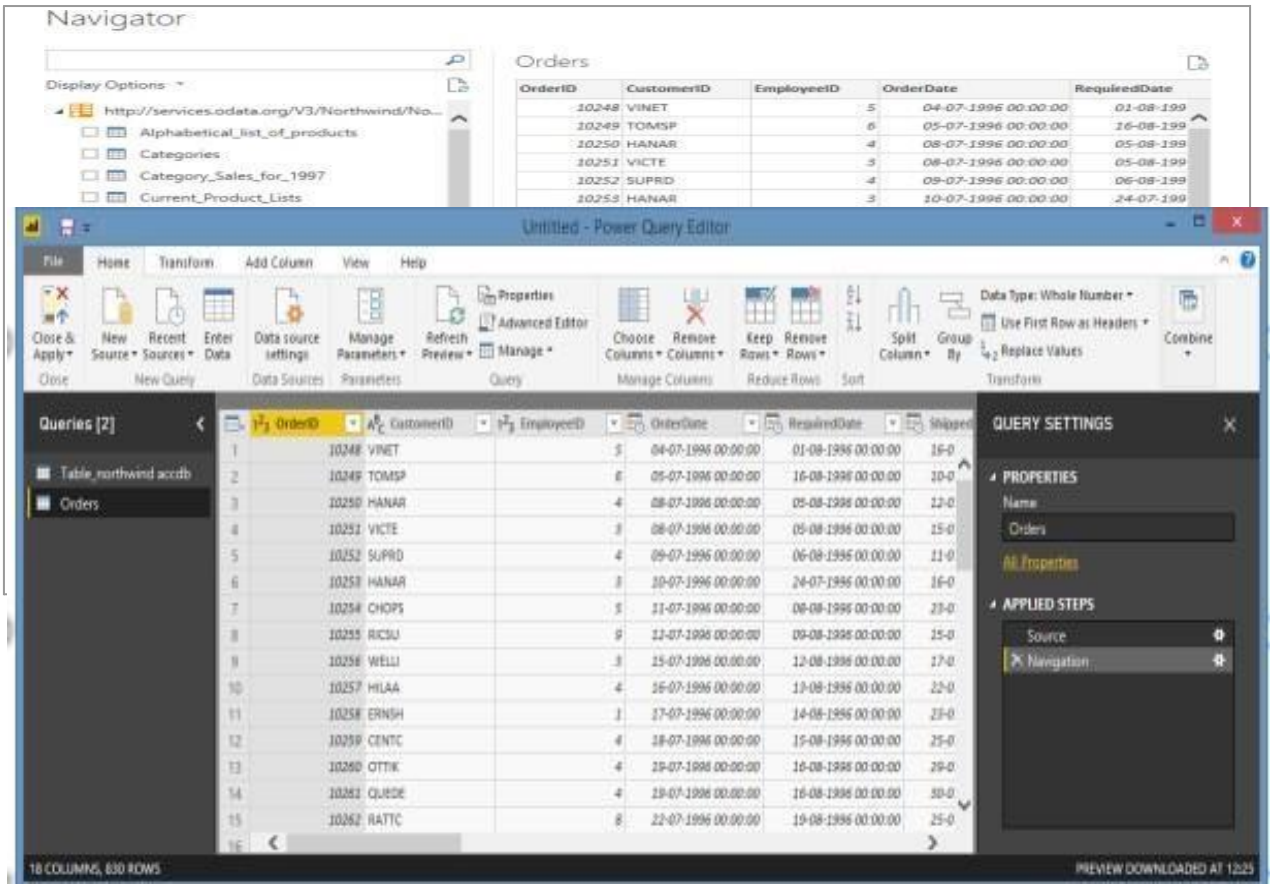


Step 7: Paste url as http://services.odata.org/V3/Northwind/Northwind.svc/ Click on ok

Step 8: Select orders table And click on edit

Note: If you just want to see preview you can just click on table name without clicking on checkbox

Click on edit to view table



Conclusion : In this way we import the Legacy datasets using the Power BI Tool.

Lab Assignment No.	02
Title	Data Visualization from Extraction Transformation and Loading (ETL) Process
Roll No.	
Class	BE AI & DS
Date Of Completion	
Subject	Computer Laboratory IV[417535]
Assessment Marks	
Assessor's Sign	

Assignment No. 02

Title: Data Visualization from Extraction Transformation and Loading (ETL) Process.

Course Objective:

- To understand the process of data extraction, transformation, and loading (ETL).
- To learn how to clean, transform, and visualize data using Python libraries like pandas, matplotlib, and seaborn.
- To develop skills in data handling, including missing values, data normalization, and data visualization.

Course Outcome:

- Students will be able to Extract data from CSV sources.
- Students will be able to Perform data cleaning and transformation, including handling missing data and normalizing numerical columns.
- Students will be able to Load the transformed data into a new file or database.

Software and Hardware Requirements :

Python (3.x), Jupyter Notebook, Python library(Pandas, Seaborn, Matplotlib).

Processor , RAM , Disk Space , Internet Connection.

Prerequisite :

- Basic Python programming and data handling with **pandas**.
- Understanding of **matplotlib** and **seaborn** for data visualization.
- Familiarity with reading/writing **CSV** files and basic statistical concepts.

Theory :

What is ETL (Extract, Transform, Load)?

ETL is a process in data warehousing that involves three main tasks:

1. **Extraction**: Collecting raw data from different sources (databases, files, APIs).
2. **Transformation**: Cleaning, filtering, and transforming the extracted data into a suitable format for analysis (e.g., handling missing values, normalizing, merging datasets).
3. **Loading**: Storing the transformed data into a new database or file, ready for analysis or reporting.

Why ETL is Important?

ETL is essential for preparing raw data for analysis and reporting. Data extracted from different sources often requires cleaning and transformation to make it useful for decision-

making. Visualizing transformed data allows businesses to derive actionable insights and make informed decisions.

Data Visualization:

Data visualization helps in representing data in graphical formats like charts, graphs, and plots. It helps identify patterns, trends, and outliers, and supports data-driven decision-making. Popular types of visualizations include:

- **Line Plots**: Useful for showing trends over time.
- **Bar Charts**: Effective for comparing quantities across different categories.
- **Histograms**: Used to visualize the distribution of numerical data.
- **Box Plots**: Help to visualize the spread and identify outliers in the data.

Python Libraries for Data Visualization:

- **matplotlib**: A foundational library for creating static, animated, and interactive visualizations.
- **seaborn**: Built on top of `matplotlib`, it provides a high-level interface for drawing attractive and informative statistical graphics.
- **pandas**: While primarily for data manipulation, pandas also includes basic plotting capabilities, which integrate with `matplotlib` for enhanced visualization.

Steps in the Practical:

1. **Extract Data**: Use pandas to read data from a CSV file.
2. **Transform Data**: Perform necessary cleaning steps like handling missing values, converting data types, and normalizing numerical columns.
3. **Load Data**: Save the cleaned and transformed data to a new CSV file.
4. **Visualize Data**: Create different types of visualizations to analyze trends, distribution, and relationships in the data.

Source Code :

Step 1: Extract Data

```
import pandas as pd

# Extract data from a CSV file
df = pd.read_csv(r"C:\Users\saira\Downloads\sales_data.csv")

# Display the first few rows
print(df.head())
```

Step 2: Transform Data

```
# Handling missing values
df.dropna(inplace=True)  # Remove missing values

# Convert date column to datetime format (if applicable)
df['date_column'] = pd.to_datetime(df['date_column'])
```

```
# Normalize a numeric column (e.g., sales)
df['normalized_sales'] = (df['sales'] - df['sales'].min()) /
(df['sales'].max() - df['sales'].min())

# Display transformed data
print(df.head())
```

Step 3: Load Data

```
# Save the cleaned data to a new CSV file
df.to_csv('cleaned_data.csv', index=False)
print("Data successfully loaded into 'cleaned_data.csv'")
```

Step 4: Data Visualization

```
import matplotlib.pyplot as plt
import seaborn as sns

# Line plot for trend analysis
plt.figure(figsize=(10,5))
sns.lineplot(x='date_column', y='sales', data=df)
plt.title('Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.xticks(rotation=45)
plt.show()

# Bar chart for categorical data
plt.figure(figsize=(10,5))
sns.barplot(x='category', y='sales', data=df)
plt.title('Sales by Category')
plt.xlabel('Category')
plt.ylabel('Sales')
plt.show()

# Histogram for distribution of sales
plt.figure(figsize=(10,5))
sns.histplot(df['sales'], bins=20, kde=True)
plt.title('Sales Distribution')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```

Conclusion : In this way we learned how to perform the ETL (Extract, Transform, Load) process using Python, including data extraction, cleaning, and transformation. They gained hands-on experience in visualizing data trends and distributions using matplotlib and seaborn.

Lab Assignment No.	03
Title	Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.
Roll No.	
Class	BE AI & DS
Date Of Completion	
Subject	Computer Laboratory IV[417535]
Assessment Marks	
Assessor's Sign	

Assignment No : 03

Title : Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server.

Objective:

- To introduce the concepts and components of Business Intelligence (BI)
- To understand the concept of ETL Process.
- To learn (ETL) process to construct the database in the Sql server.

Outcomes:

- Students will be able to explain concepts and components of Business Intelligence (BI)
- Students will be able to explain understand the concept of ETL Process.
- Students will be able to use (ETL) process to construct the database in the Sql server.

Software & Hardware Requirement:

- 64-bit Open source Linux or its derivative
- Additional tools like powerBI, SQL Server

Prerequisite:

1. Basics of ETL Tools.
2. Concept of Sql Server.

Theory :

What is ETL?

The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for **Extraction, Transformation and Loading**.

The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.

To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes. ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.

1. Extraction

- Identify the Data Sources: The first step in the ETL process is to identify

the data sources. This may include files, databases, or other data repositories.

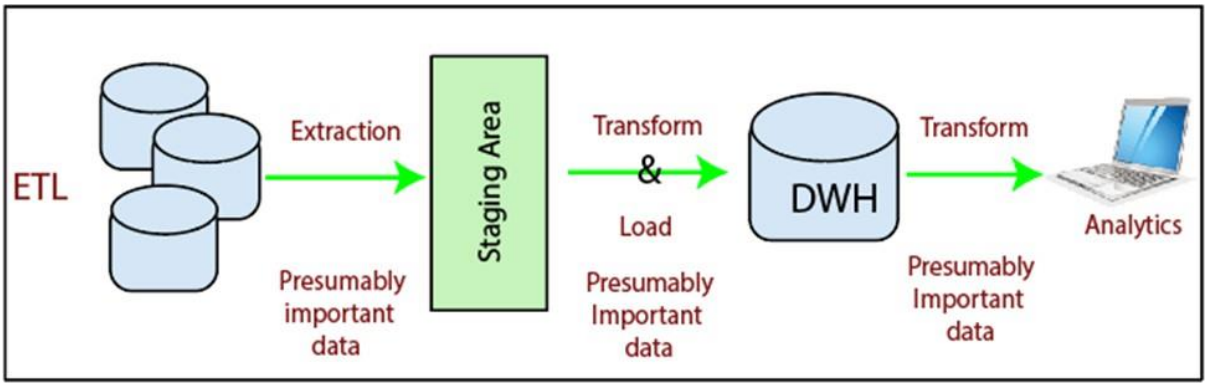
- Extract the Data: Once the data sources are identified, we need to extract the data from them. This may involve writing queries to extract the relevant data or using tools such as SSIS to extract data from files or databases.
- Validate the Data: After extracting the data, it's important to validate it to ensure that it's accurate and complete. This may involve performing data profiling or data quality checks.

2. Transformation

- Clean and Transform the Data: The next step in the ETL process is to clean and transform the data. This may involve removing duplicates, fixing invalid data, or converting data types. We can use tools such as SSIS or SQL scripts to perform these transformations.
- Map the Data: Once the data is cleaned and transformed, we need to map the data to the appropriate tables and columns in the database. This may involve creating a data mapping document or using a tool such as SSIS to perform the mapping.

3. Loading

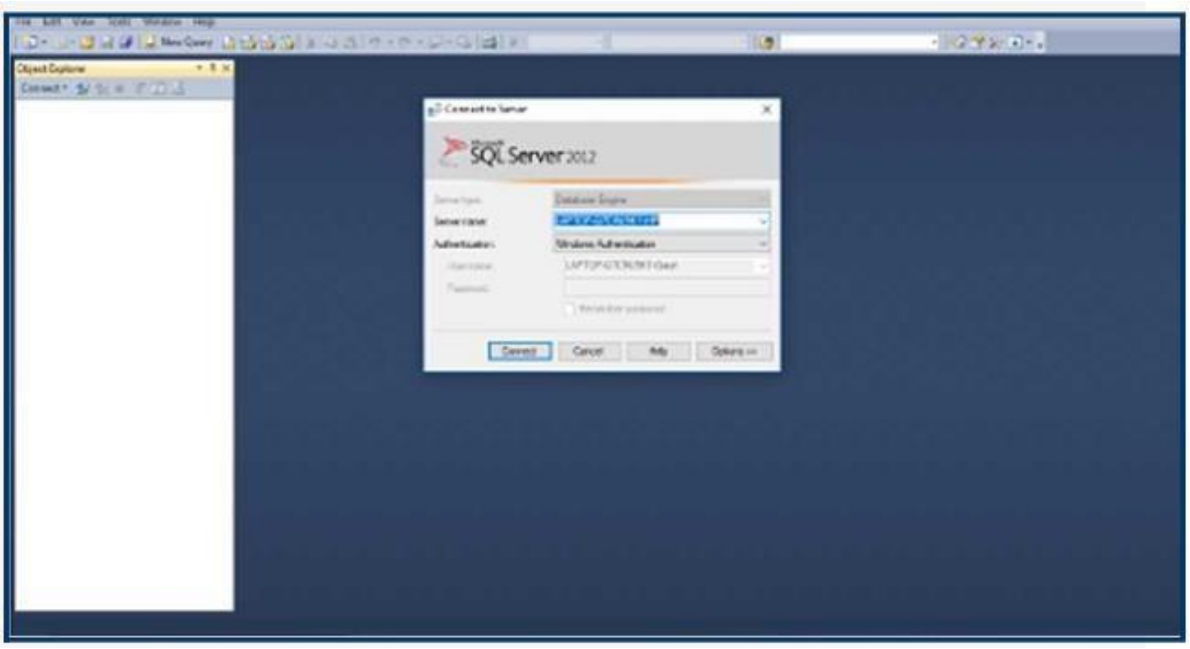
- Create the Database: Before loading the data, we need to create the database and the appropriate tables. This can be done using SQL Server Management Studio or a SQL script.
- Load the Data: Once the database and tables are created, we can load the data into the database. This may involve using tools such as SSIS or writing SQL scripts to insert the data into the appropriate tables.
- Validate the Data: After loading the data, it's important to validate it to ensure that it was loaded correctly. This may involve performing data profiling or data quality checks to ensure that the data is accurate and complete.



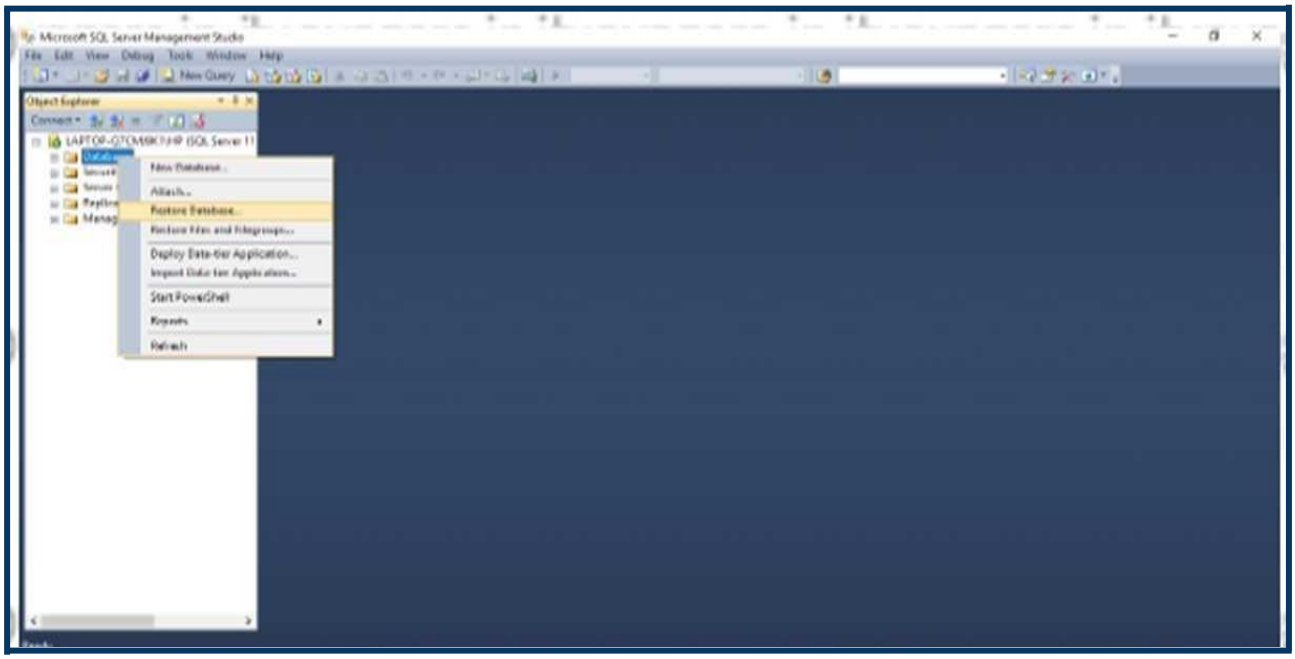
Perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL server.

Software requirements: SQL SERVER 2012 FULL VERSION (SQLServer2012SPl-FullSlipstream-ENU-x86)

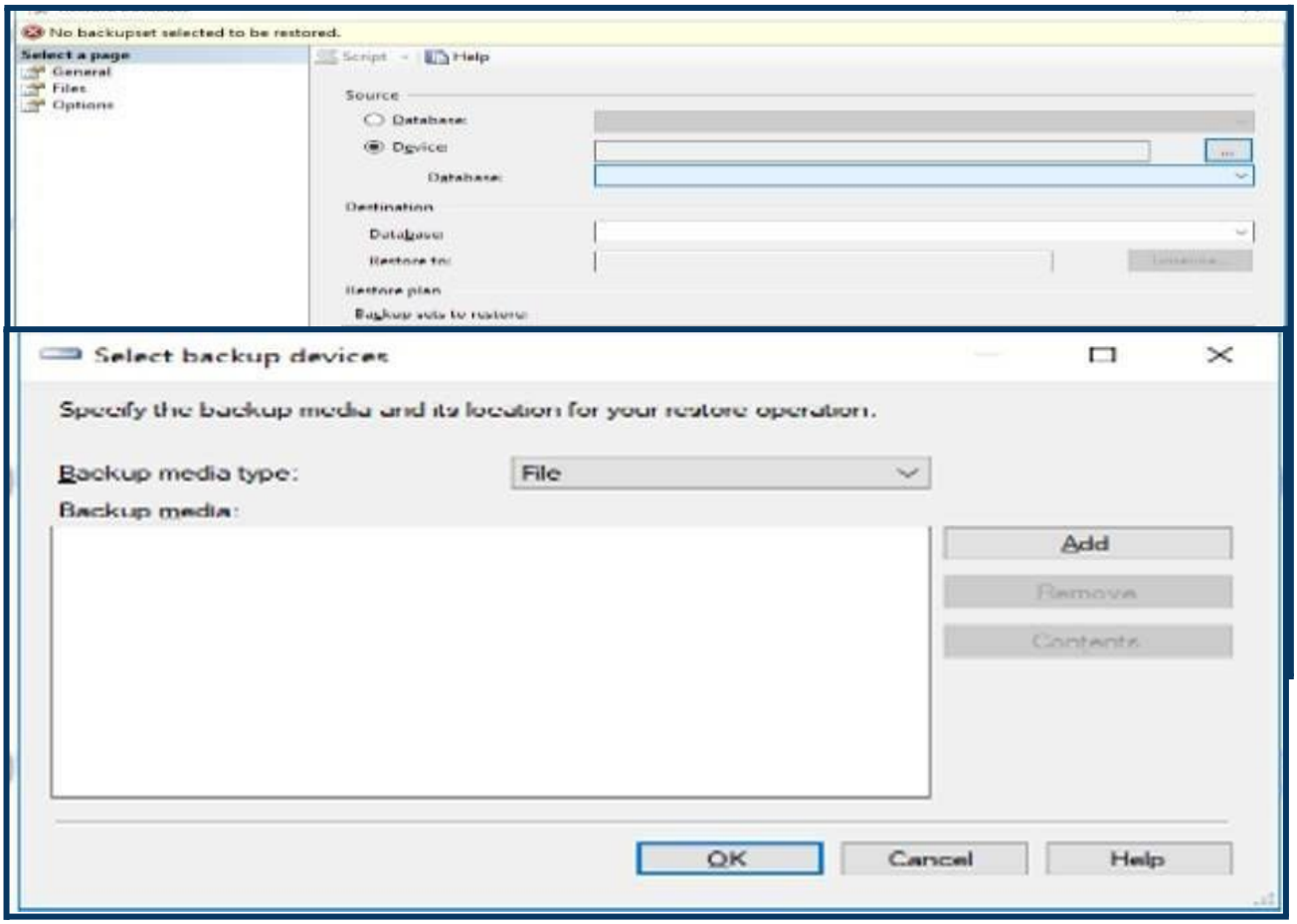
Step 1: Open SQL Server Management Studio to restore backup file



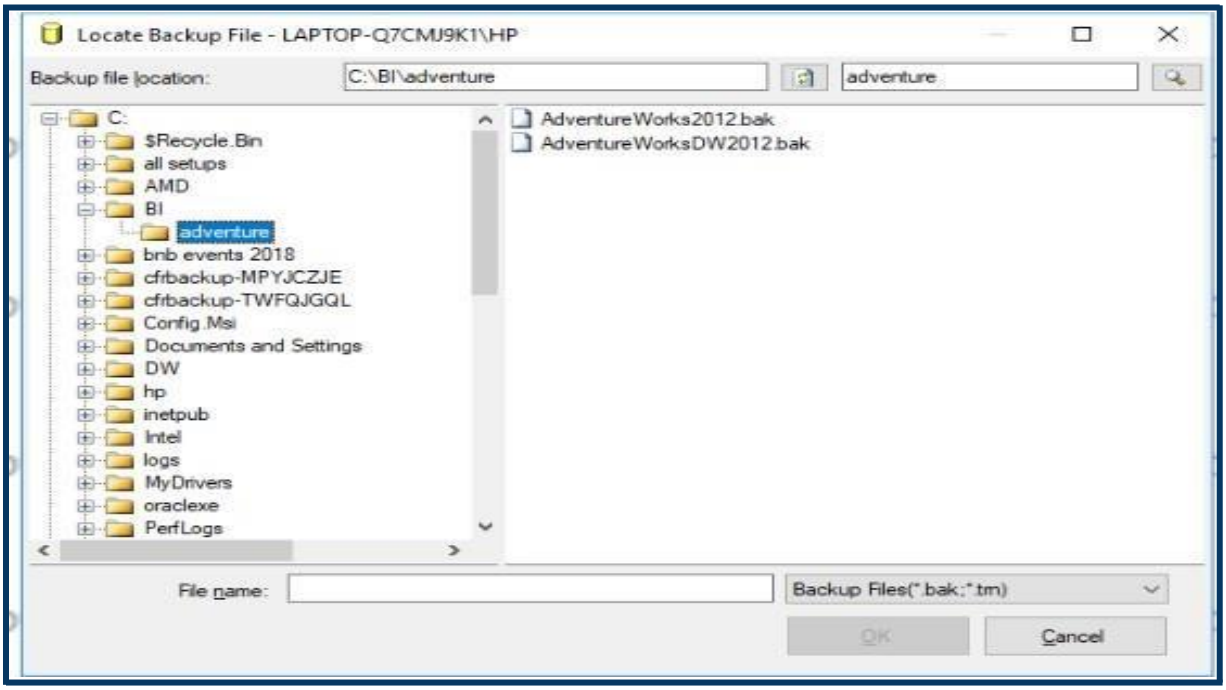
Step 2: Right click on Databases Restore Database



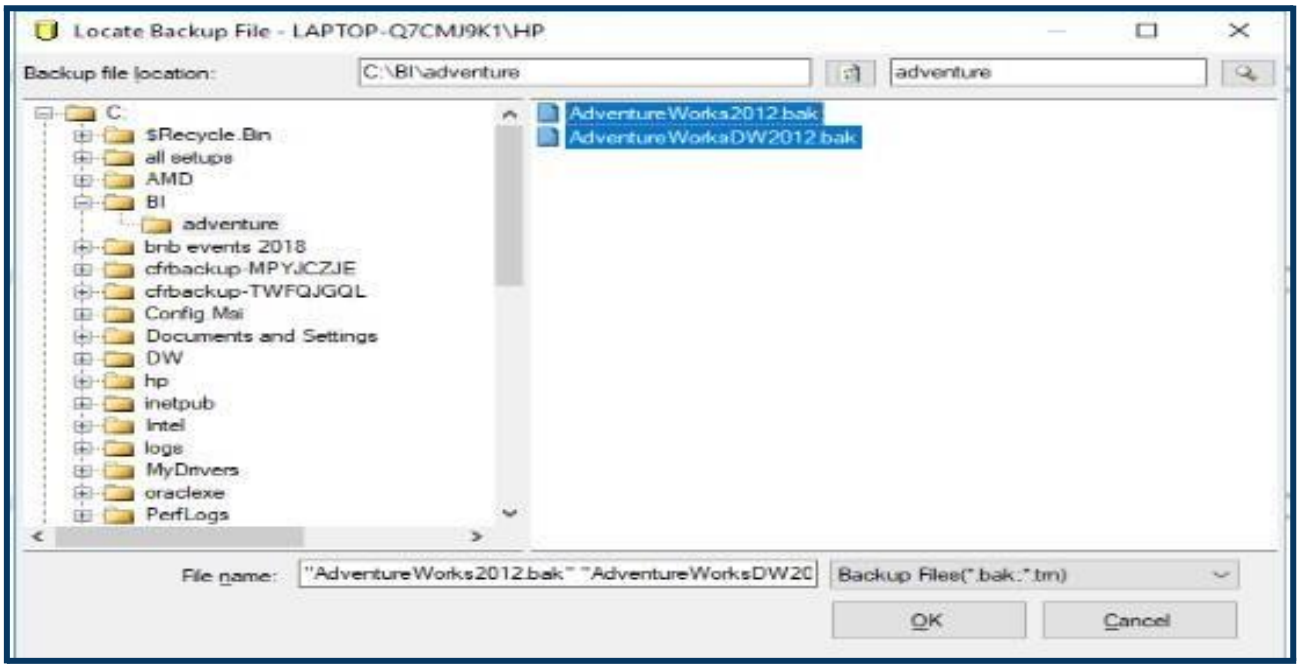
Step 3: Select Device click on ... icon towards end of device box



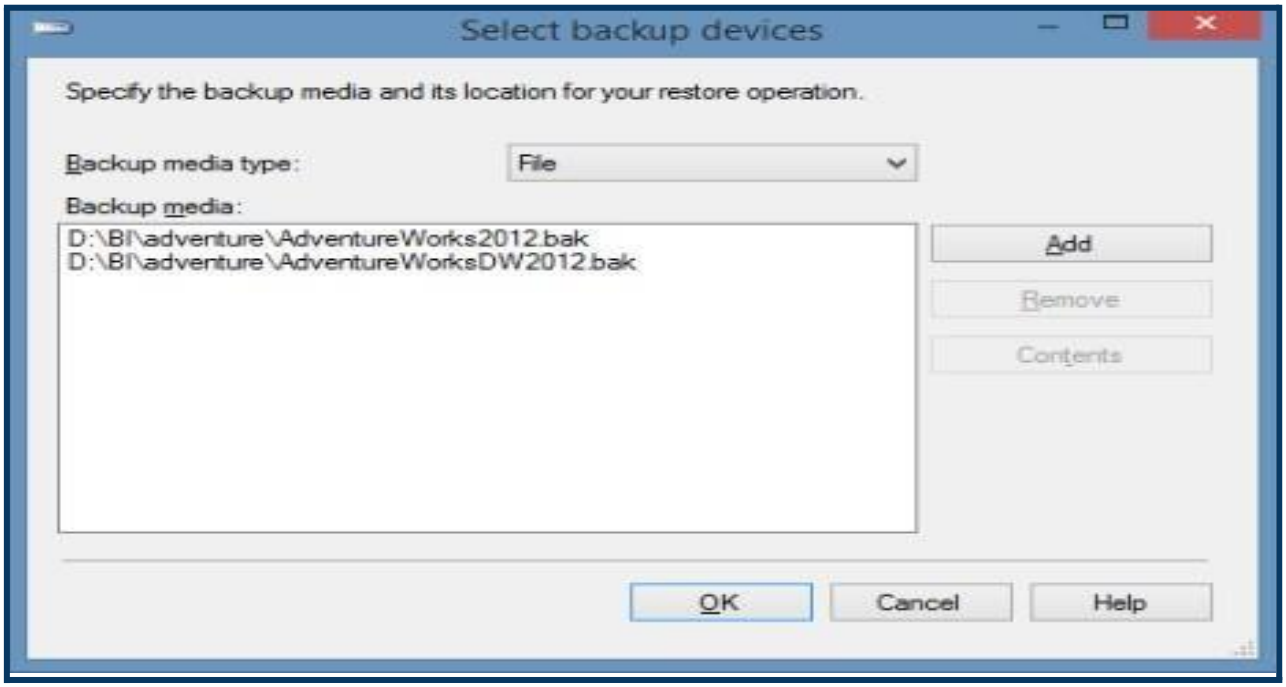
Step 4: Click on Add Select path of backup files

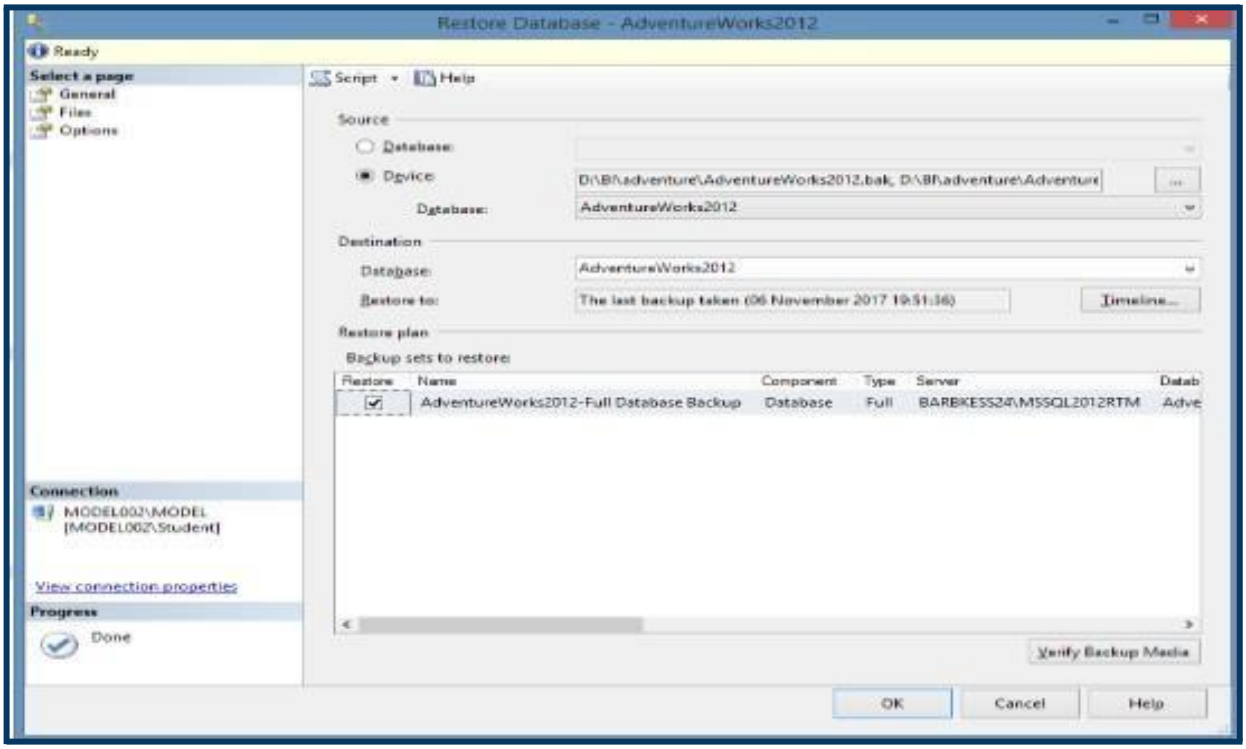


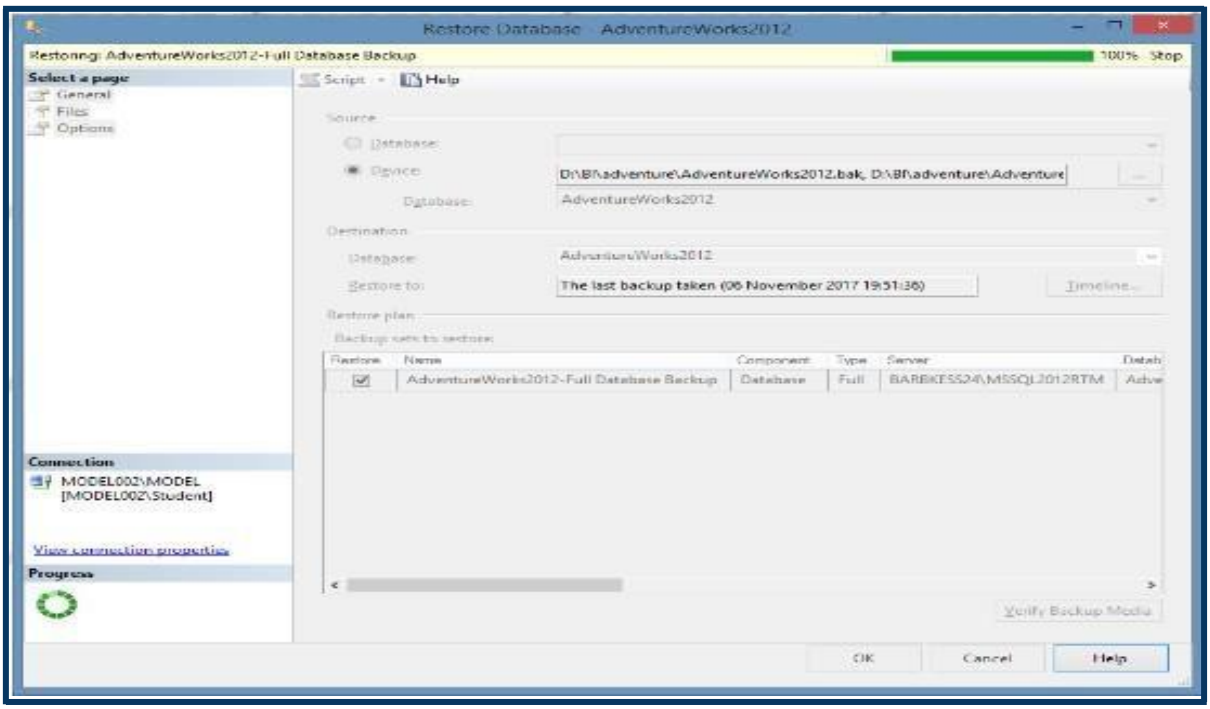
Step 5: Select both files at a time

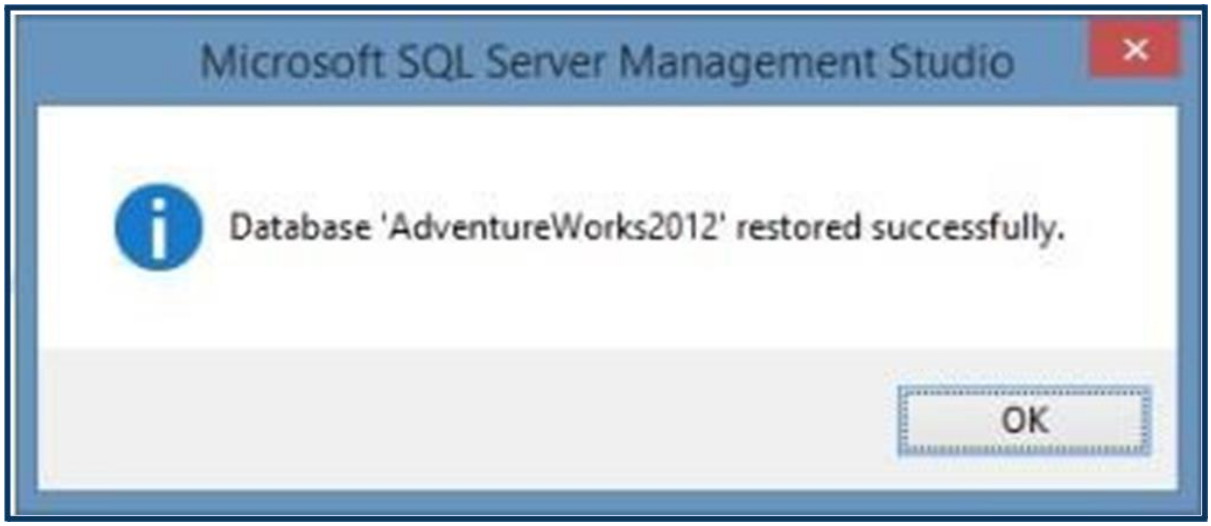


Step 6 : Click ok and in select backup devices window Add both files of Adventure Works

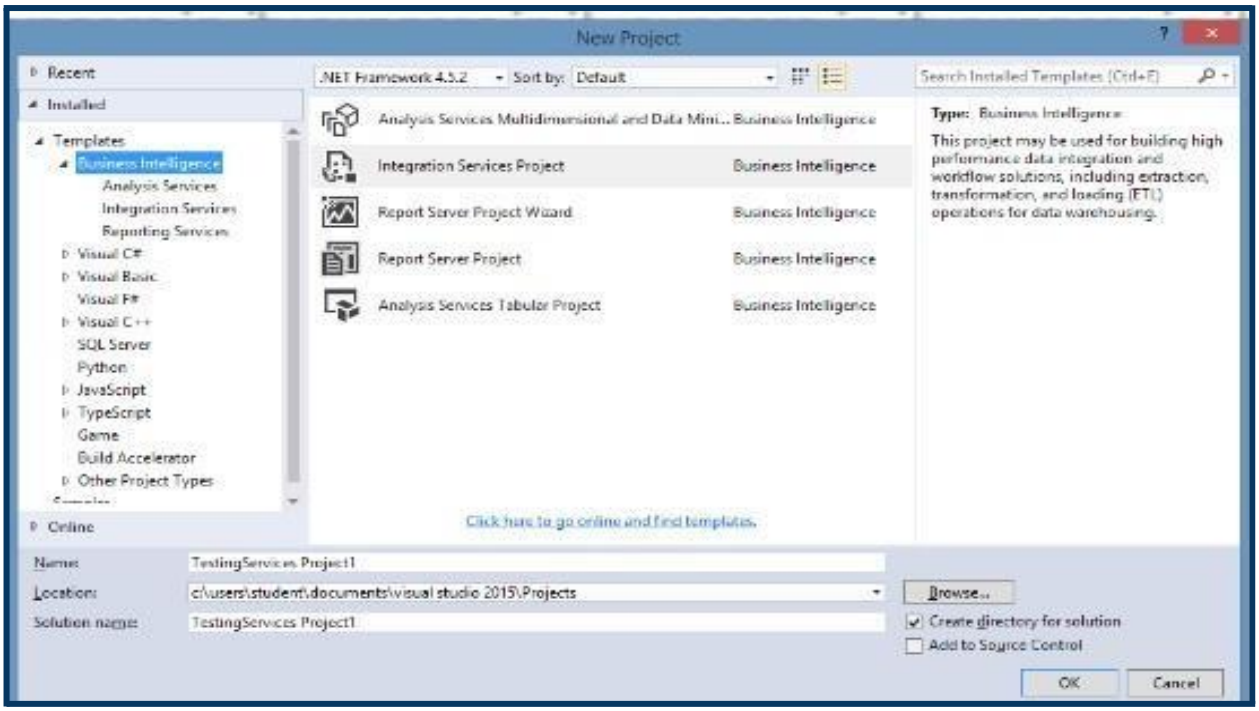


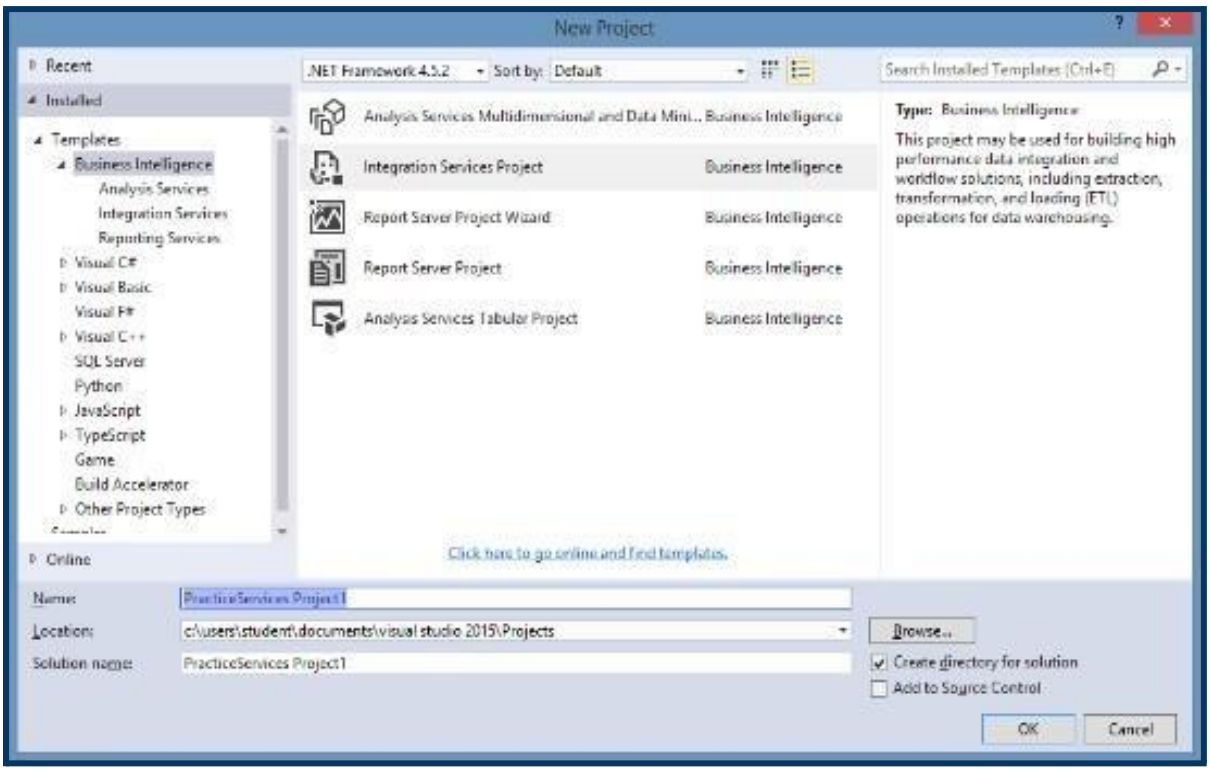


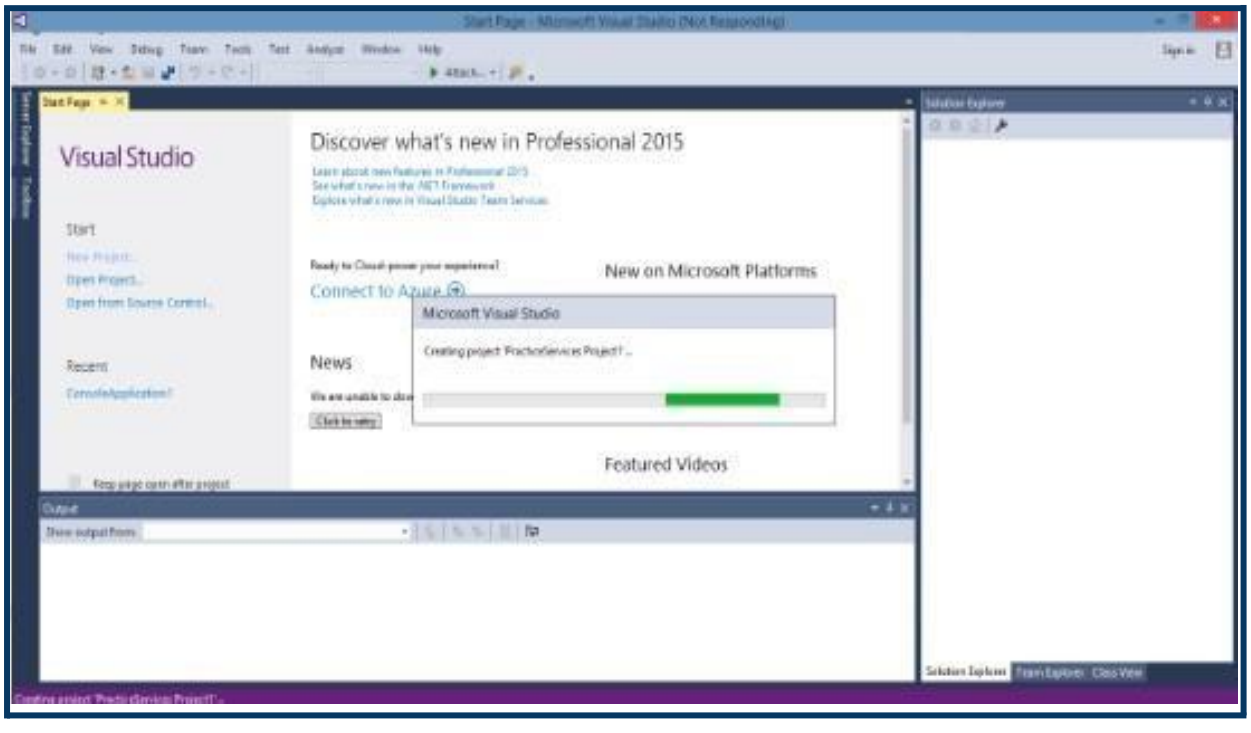

Restore Database - AdventureWorks2012
Restoring AdventureWorks2012-Full Database Backup
100%
Stop
Select a page
General
Files
Options
Script
Help
Source
Database:
Device:
D:\BI\adventure\AdventureWorks2012.bak; D:\BI\adventure\Adventure
Database:
AdventureWorks2012
Destination
Database:
AdventureWorks2012
Restore to:
The last backup taken (06 November 2017 19:51:36)
Timeline...
Restore plan
Backup sets to restore:
Restore
Name
Component
Type
Server
Datab
AdventureWorks2012-Full Database Backup
Database
Full
BARBKESS24\MSSQL2012RTM
Adve
Verify Backup Media
Connection
MODEL002\MODEL
[MODEL002\Student]
View connection properties
Progress
OK
Cancel
Help

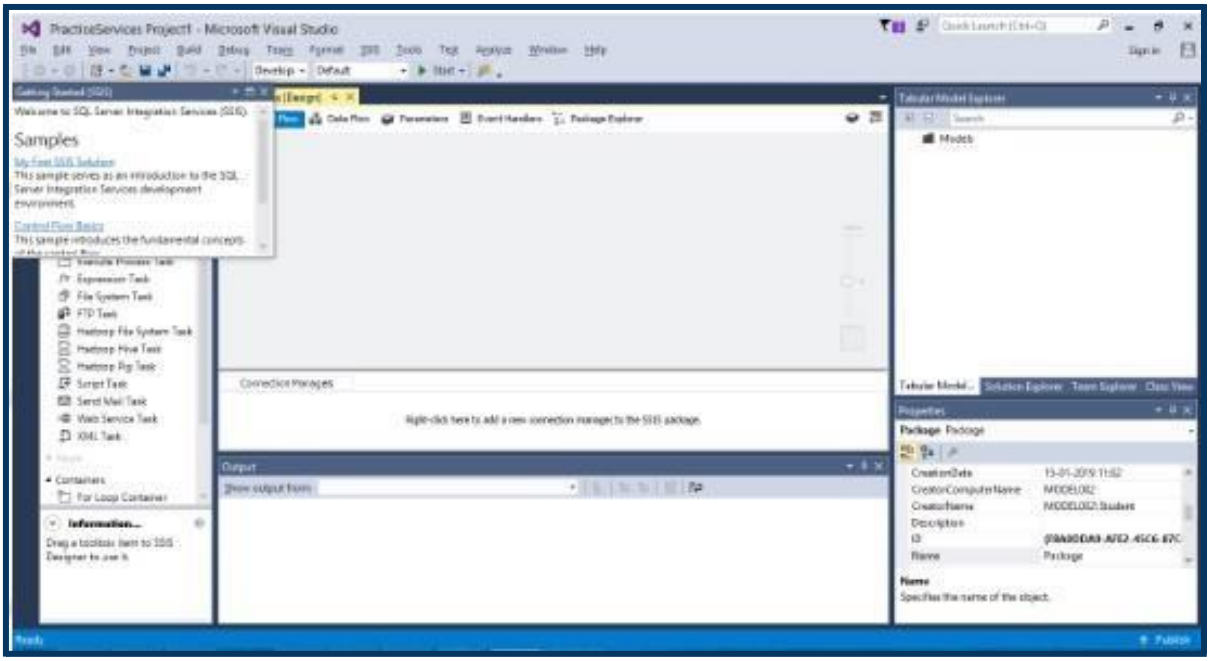

Microsoft SQL Server Management Studio
Database 'AdventureWorks2012' restored successfully.
OK

Step 7: Open SQL Server Data Tools, Select File New Project Business Intelligence Integration Services Project & give appropriate project name.



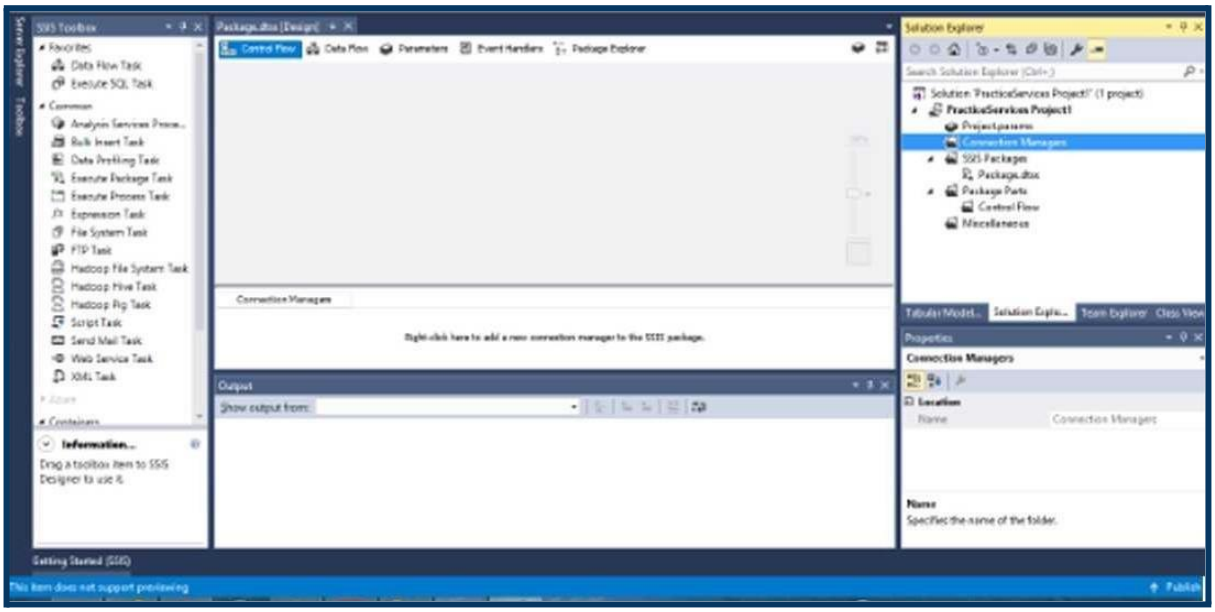

New Project
Recent
Installed
Templates
Business Intelligence
Analysis Services
Integration Services
Reporting Services
Visual C#
Visual Basic
Visual F#
Visual C++
SQL Server
Python
JavaScript
TypeScript
Game
Build Accelerator
Other Project Types
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
Analysis Services Multidimensional and Data Mini... Business Intelligence
Integration Services Project Business Intelligence
Report Server Project Wizard Business Intelligence
Report Server Project Business Intelligence
Analysis Services Tabular Project Business Intelligence
Type: Business Intelligence
This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.
Click here to go online and find templates.
Name:
Location: c:\users\student\documents\visual studio 2015\Projects
Solution name: PracticeServices Project1
Browse...
Create directory for solution
Add to Source Control
OK
Cancel


Visual Studio
Discover what's new in Professional 2015
Start
Recent
New on Microsoft Platforms
Connect to Azure
News
Microsoft Visual Studio
Featured Videos

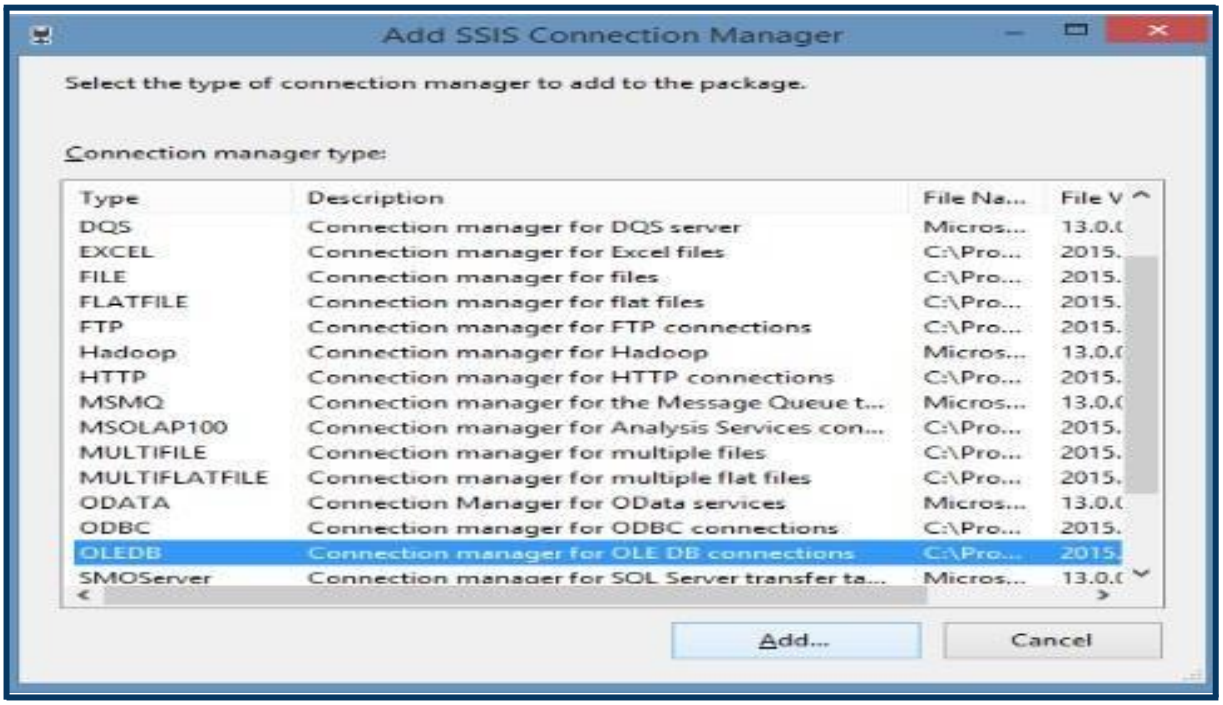


Step 8: Right click on Connection Managers in solution explorer and click on New Connection Manager.

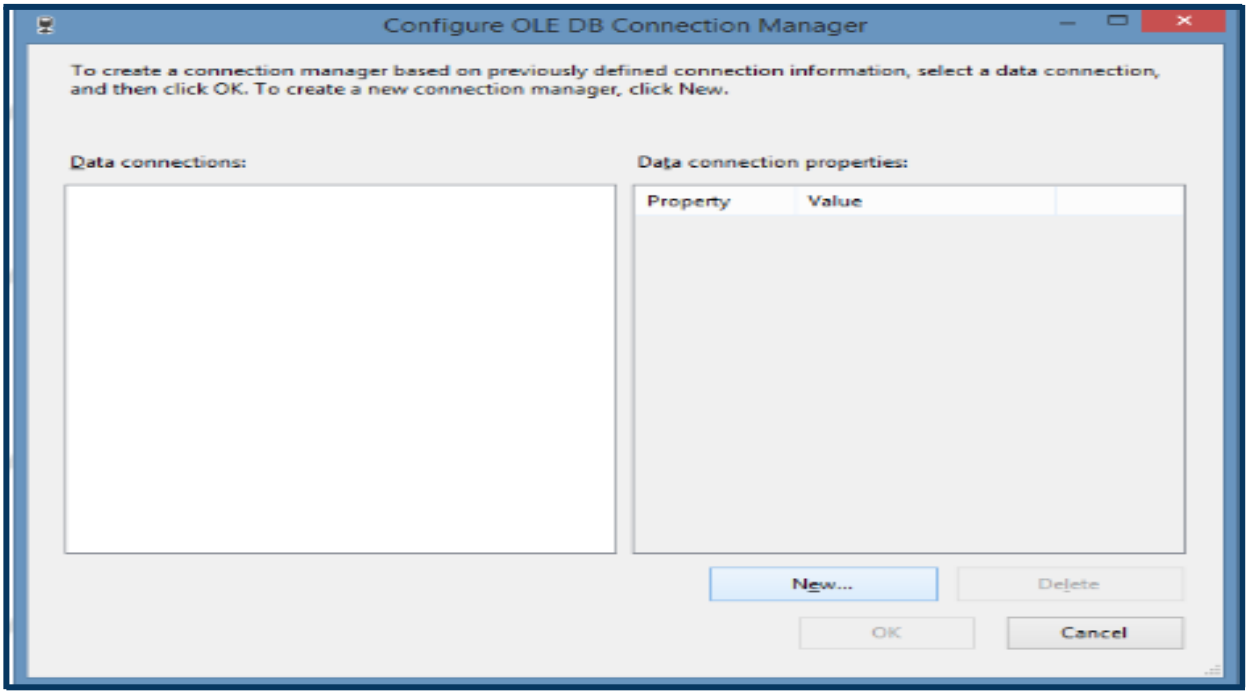
Add the SSIS connection manager window.



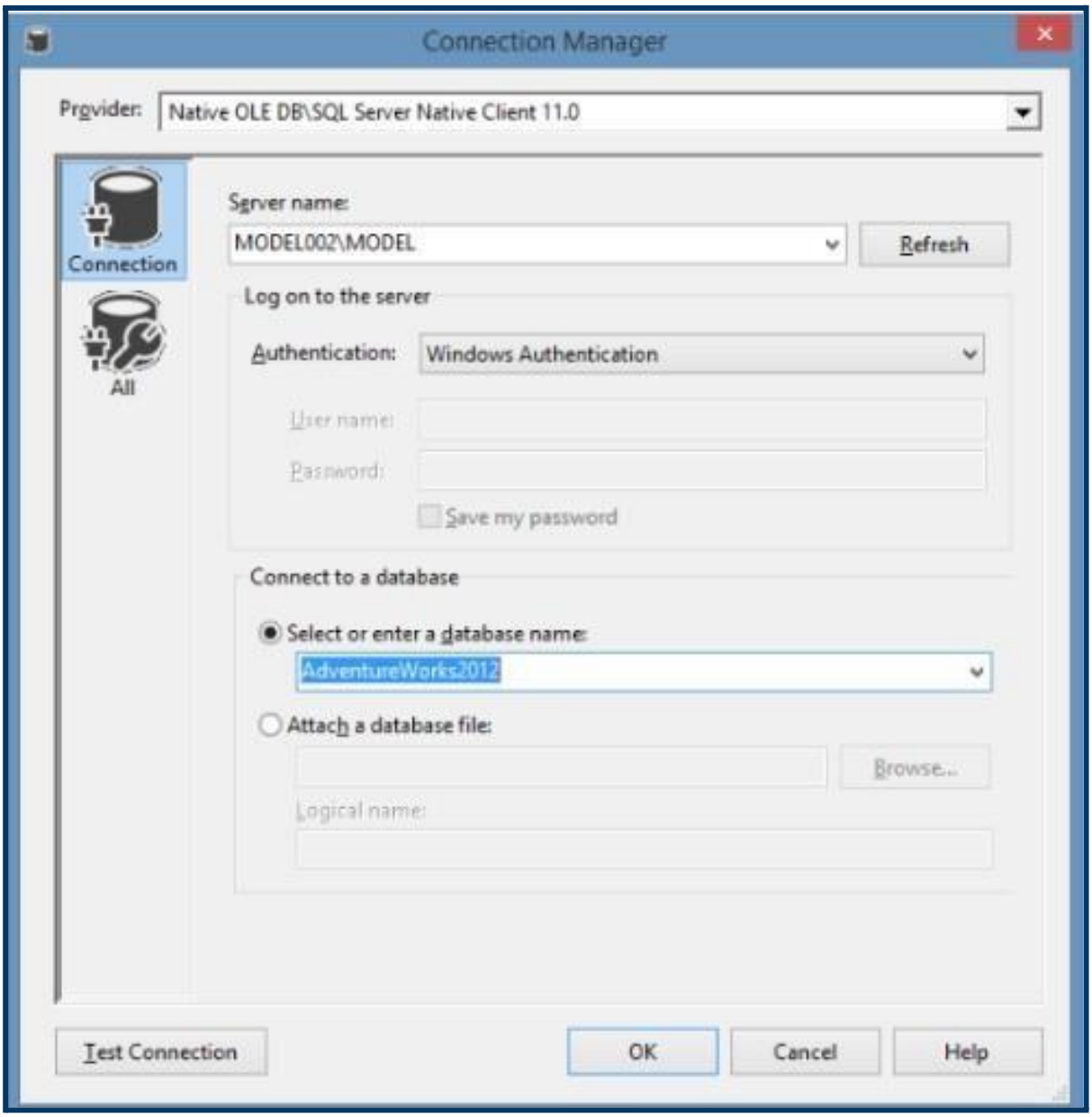
Step 9: Select OLEDB Connection Manager and Click on Add



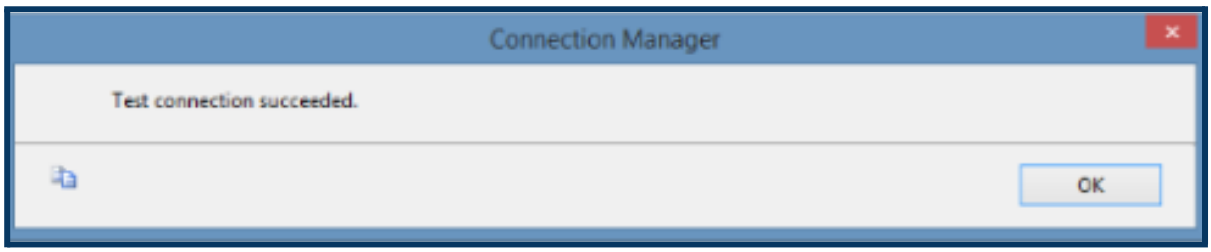
Step 10: Configure OLE DB Connection Manager window appears Click on New



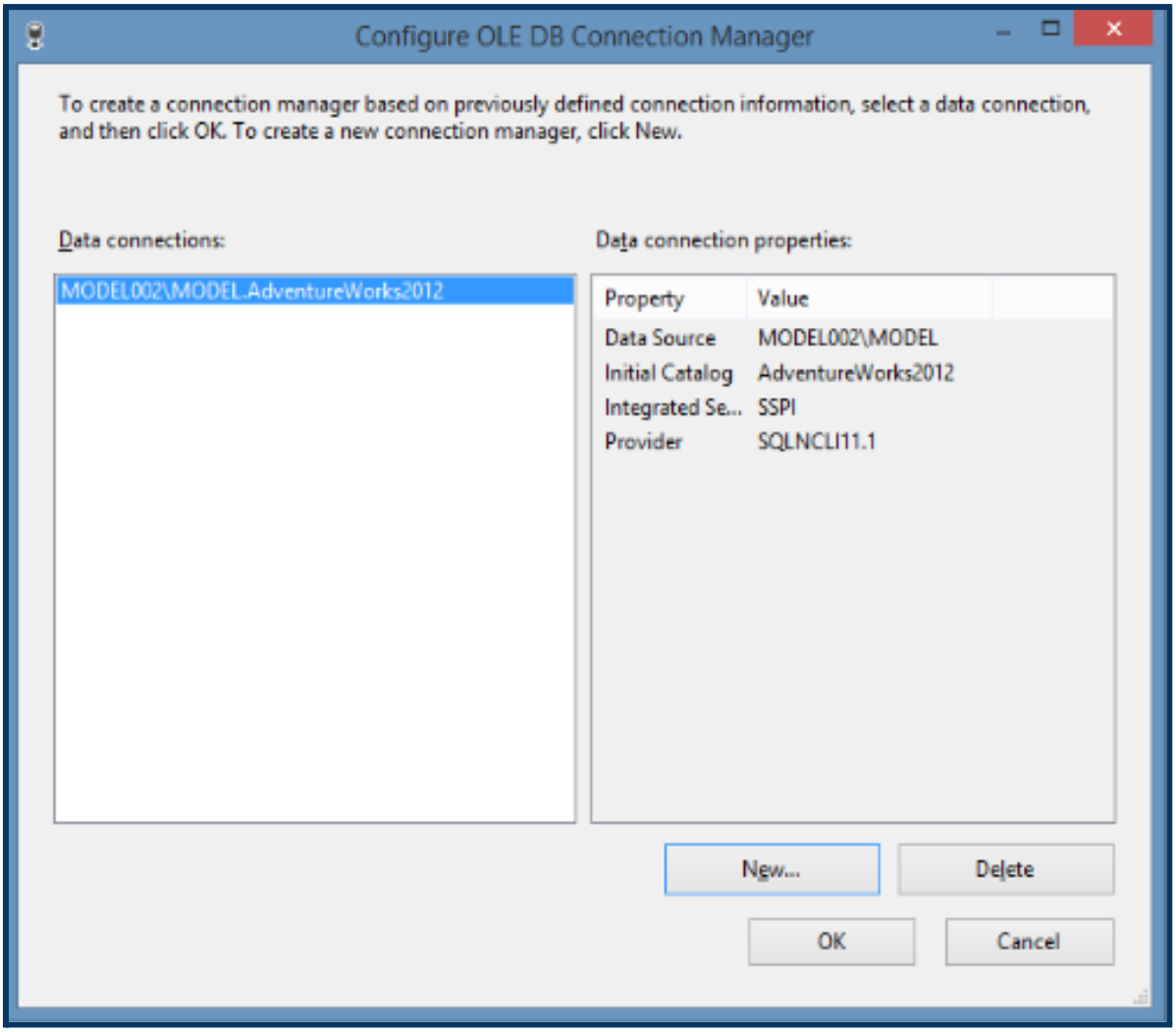
Step 11: Select Server name(as per your machine) from drop down and database name and click on Test connection.



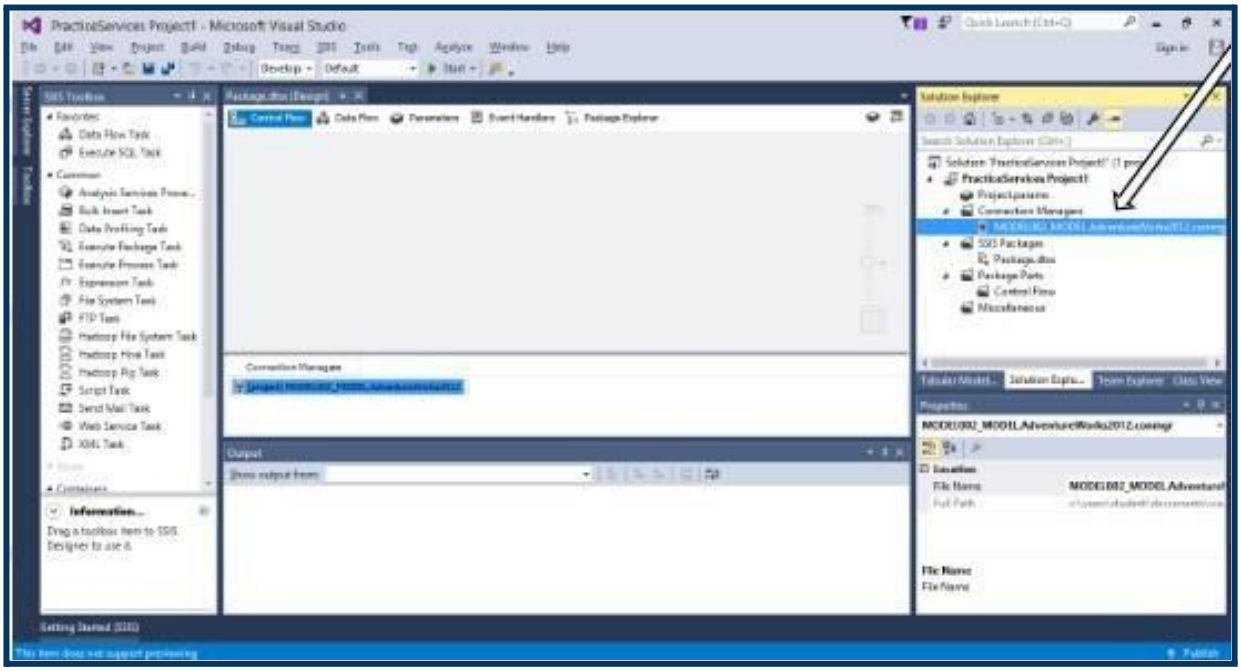
If the test connection succeeded, click on OK.



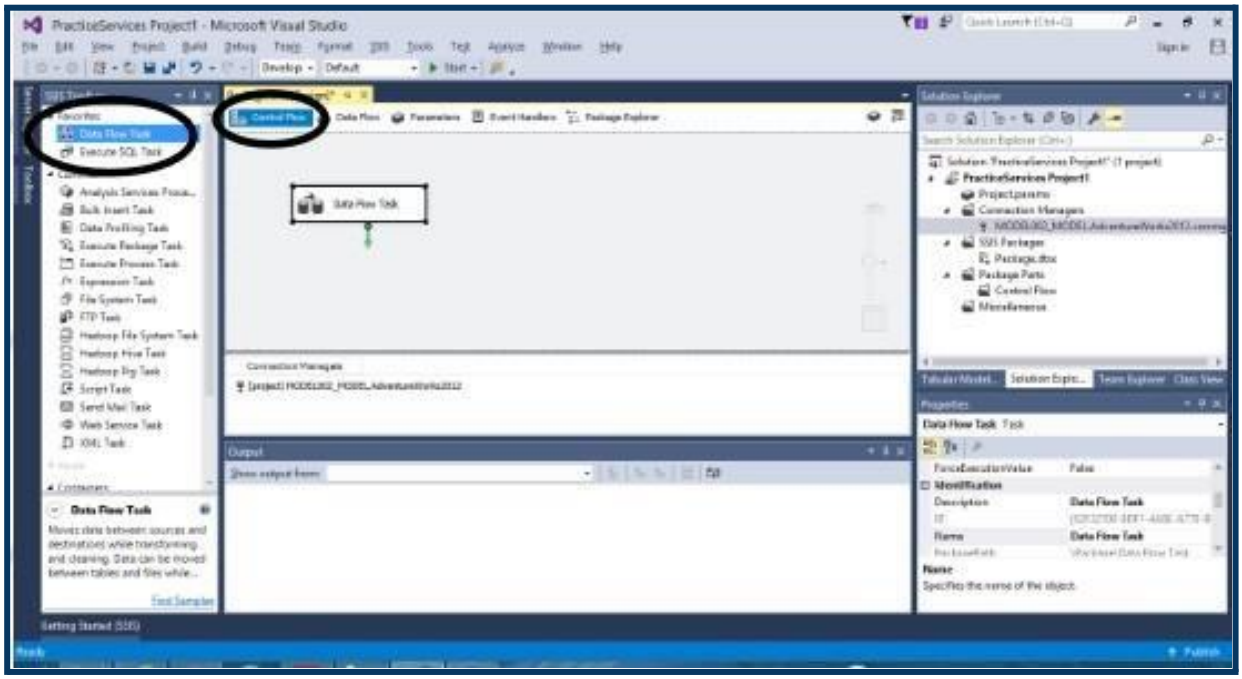
Step 12: Click on OK



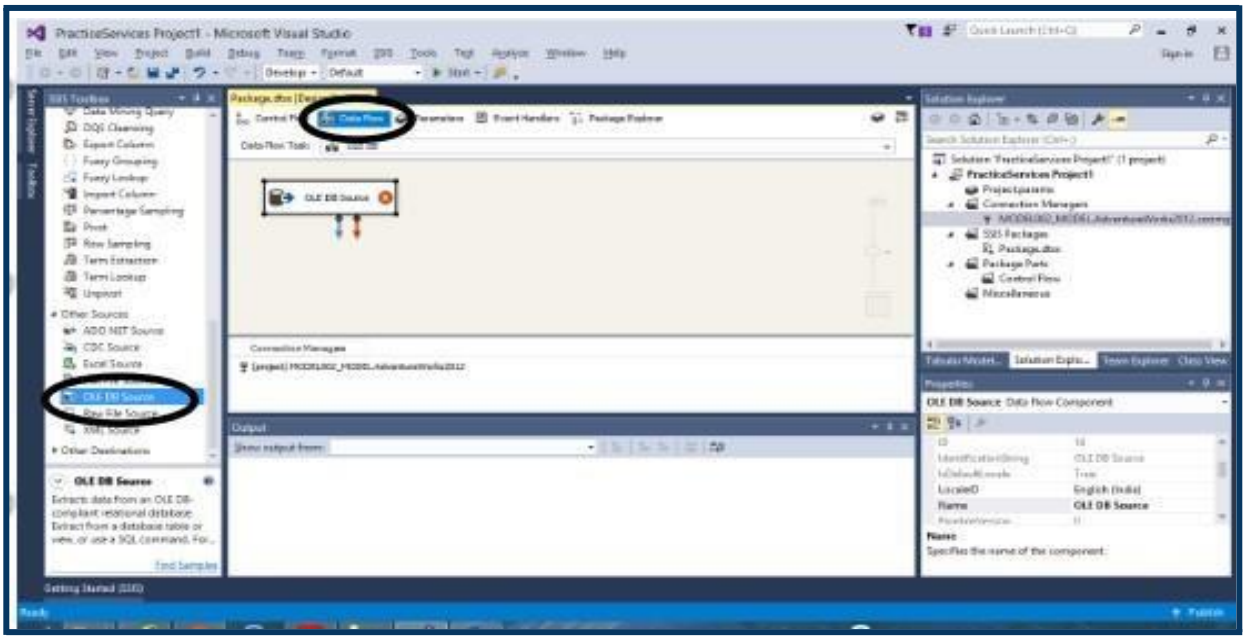
Connection is added to connection manager



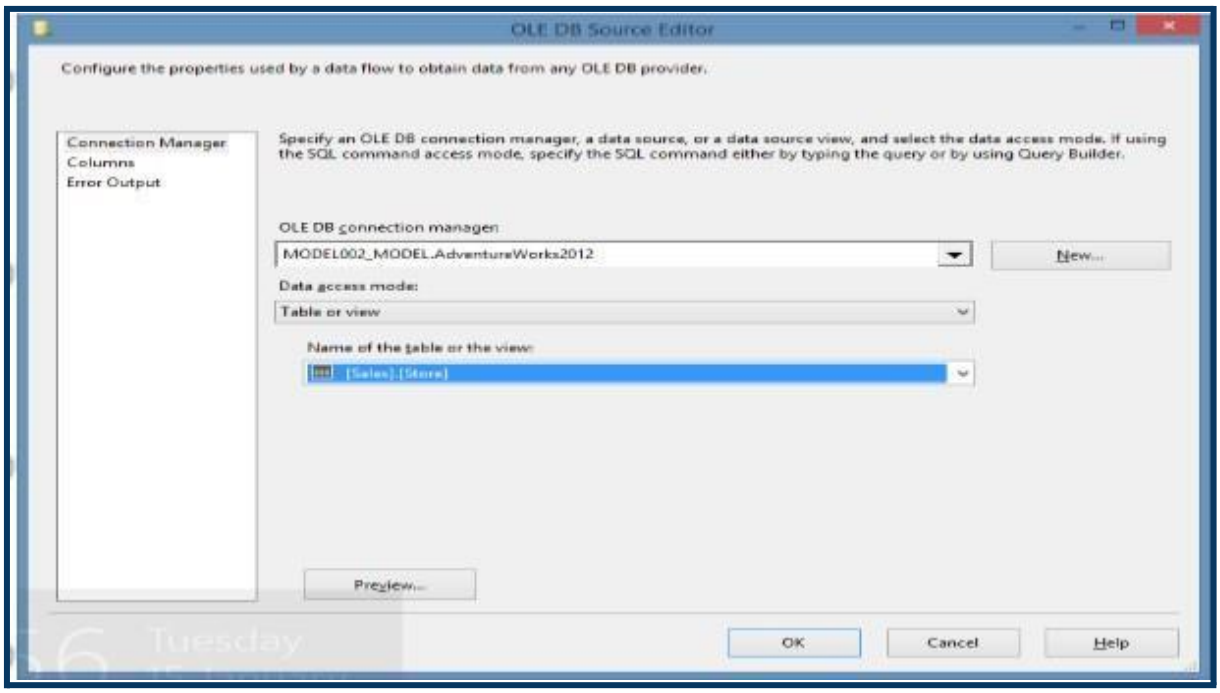
Step 13: Drag and drop Data Flow Task in Control Flow tab



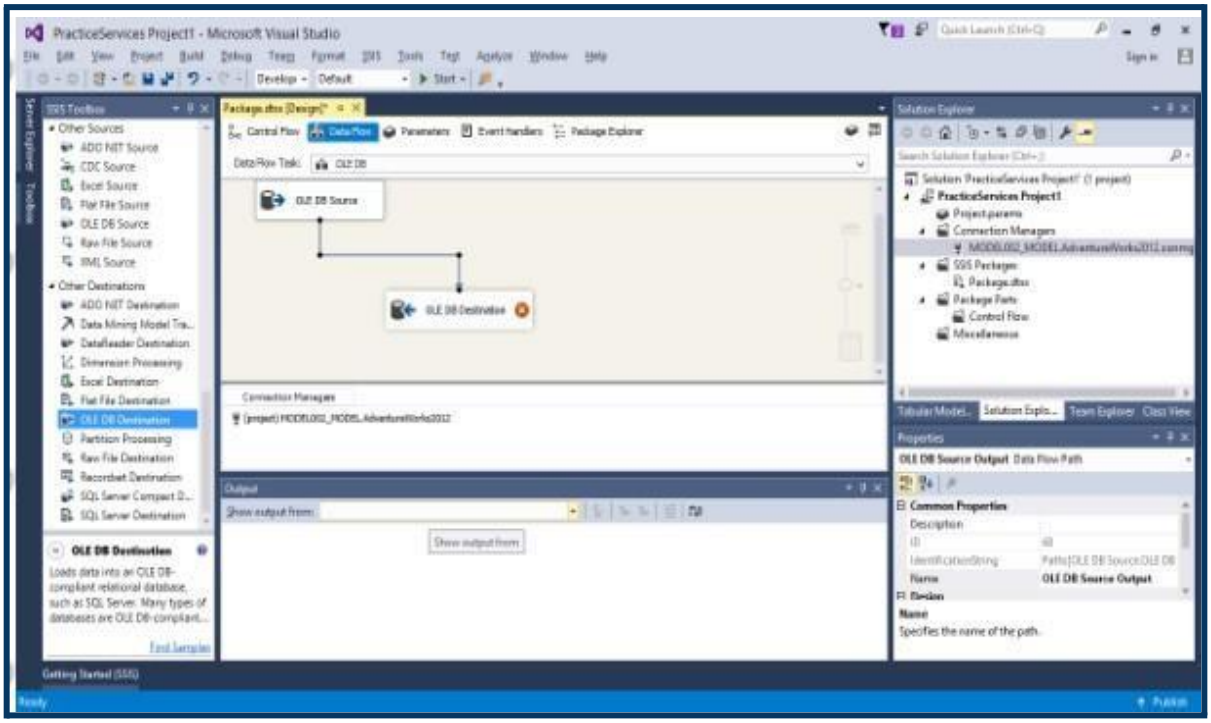
Step 14: Drag OLE DB Source from Other Sources and drop into Data Flow tab



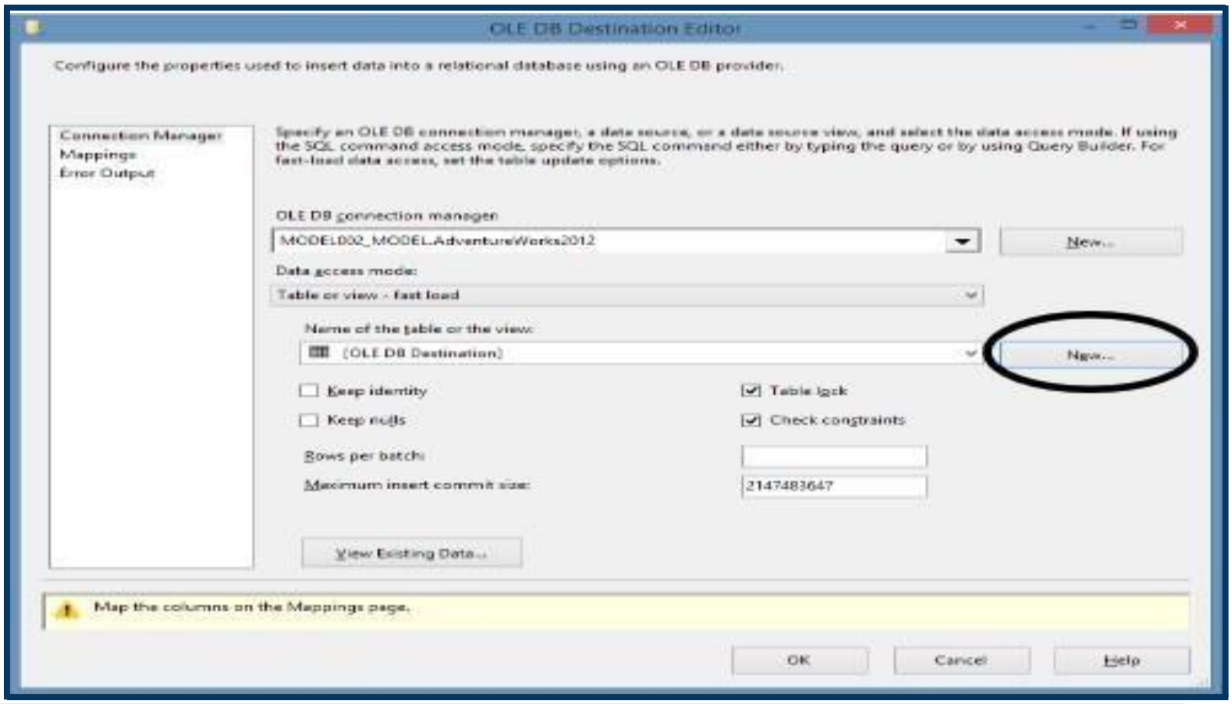
Step 15: Double click on OLE DB source -> OLE DB Source Editor appears-> click on New to add connection manager.
Select [Sales].[Store] table from drop down ok

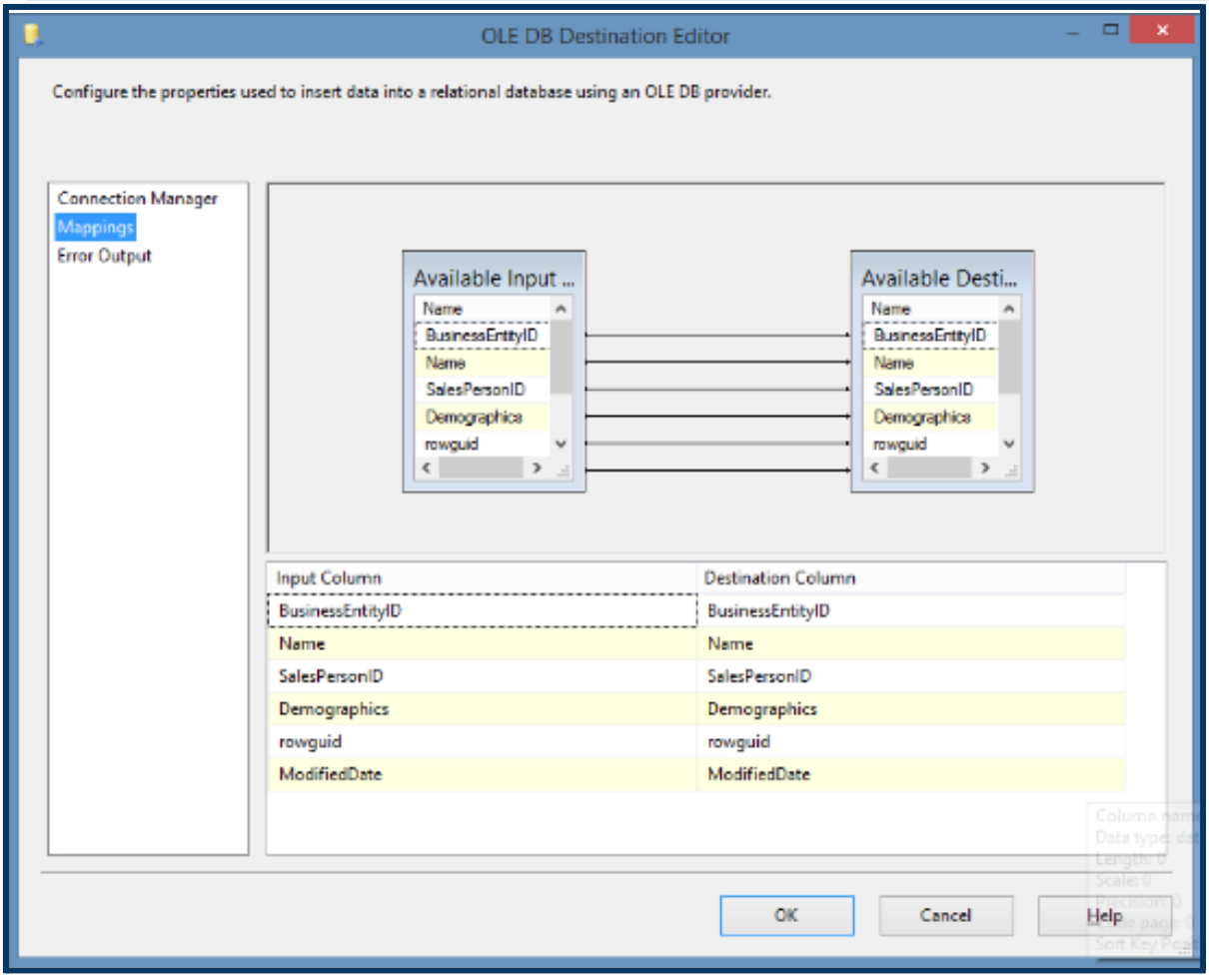


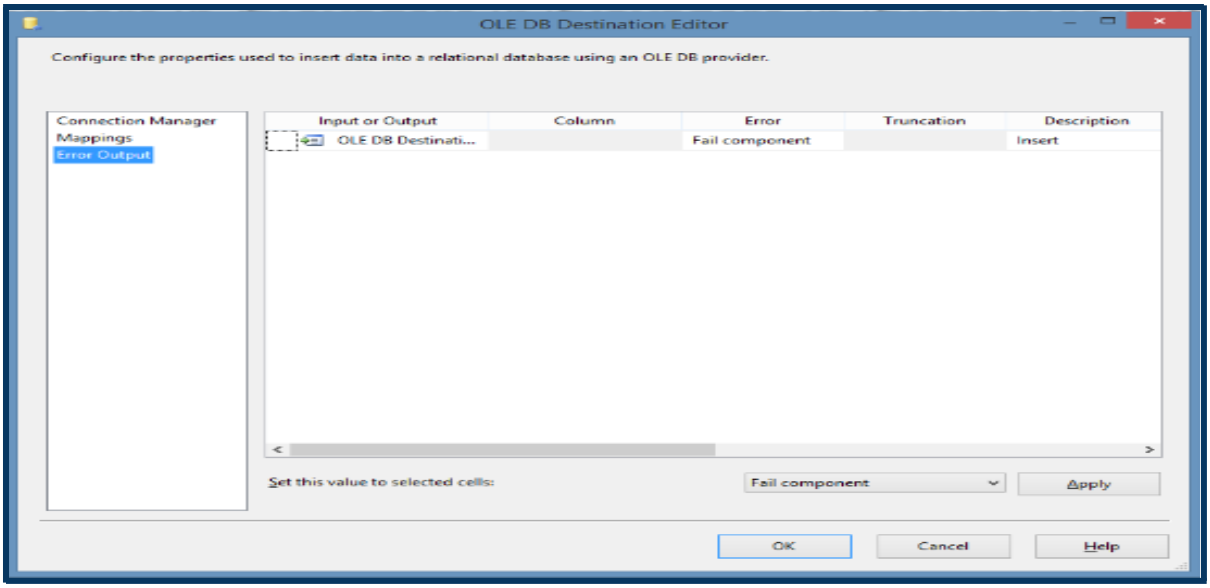
Step 16: Drag ole db destination in data flow tab and connect both



Step 17: Double click on OLE DB destination, Click on New to run the query to get [OLE DB Destination] in Name of the table or the view.

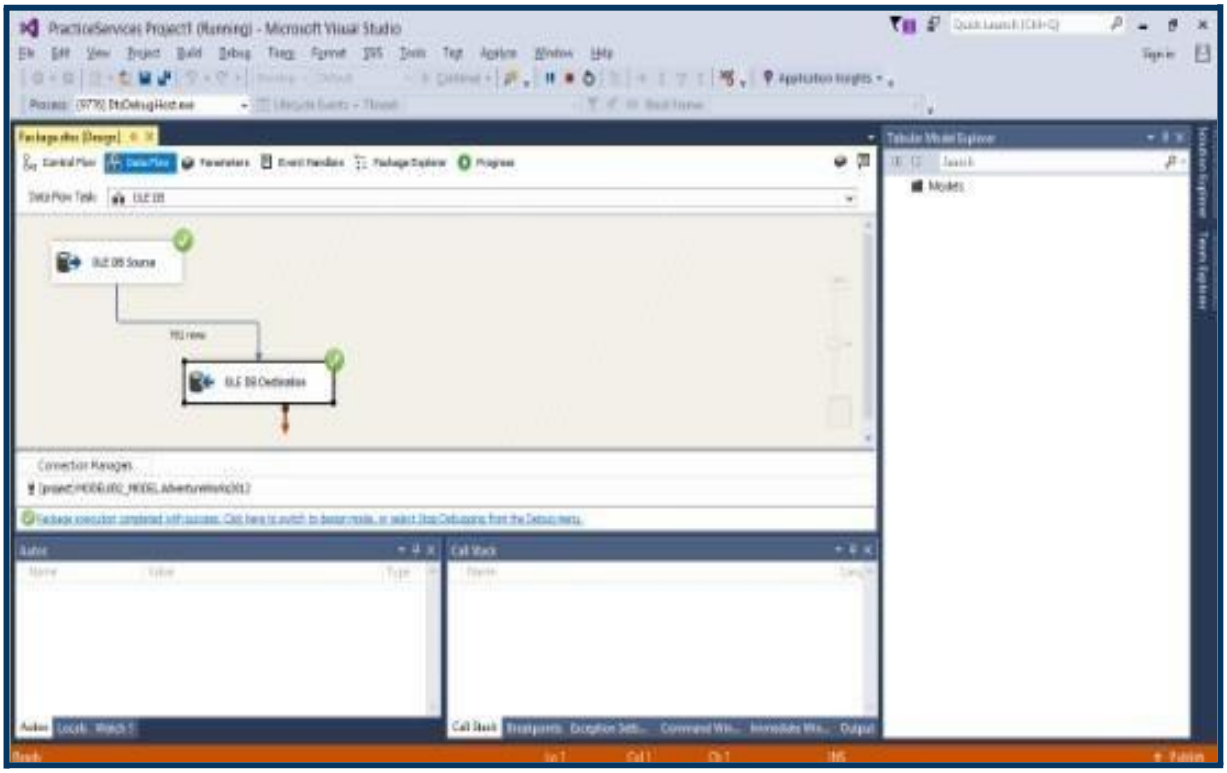




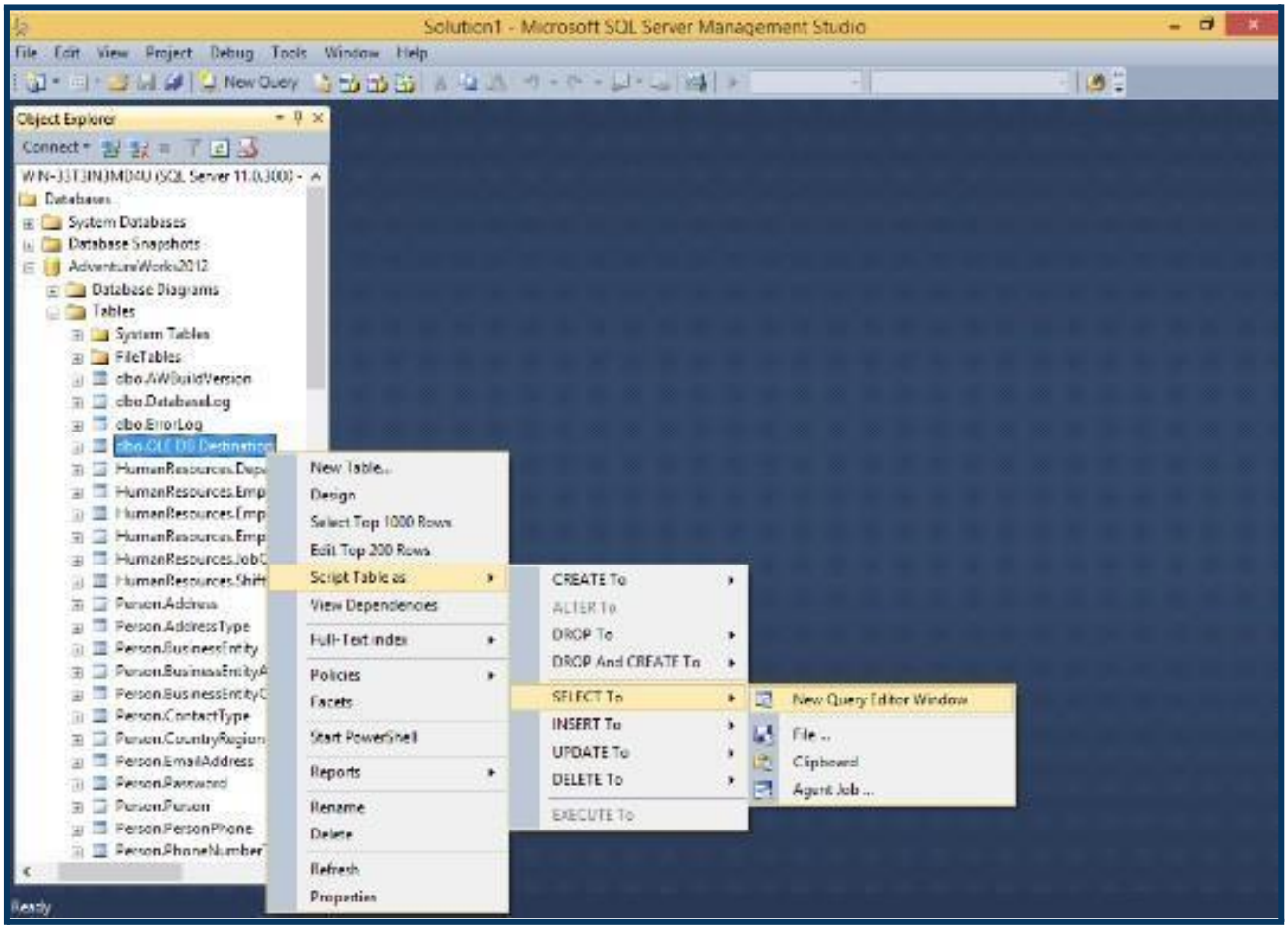


Click on OK.

Step 18: Click on Start



Step 19: Go to SQL Server Management Studio
In database tab Adventureworks Right click on [dbo].[OLE DB Destination] Script Table as SELECT To New Query Editor Window

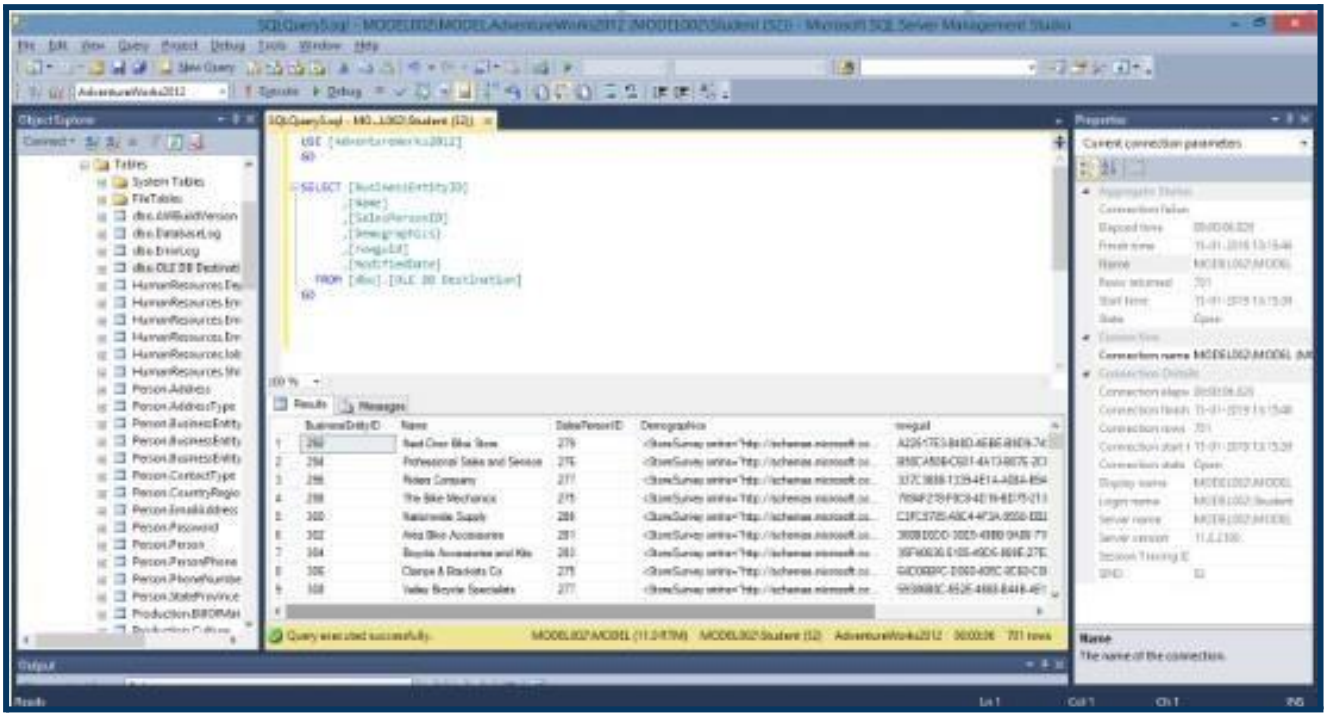


Step 20: Execute the following query to

get output. USE

[AdventureWorks2012] GO
SELECT [BusinessEntityID]
,[Name]
,[SalesPersonID]
,[Demographics]
,[rowguid]
,[ModifiedDate]

FROM [dbo].[OLE DB Destination] GO



Conclusion : In this way we can perform the ETL process to construct a database in SQL Server.

Lab Assignment No.	04
Title	Perform the data classification algorithm using any Classification algorithm
Roll No.	
Class	BE AI & DS
Date Of Completion	
Subject	Computer Laboratory IV[417535]
Assessment Marks	
Assessor's Sign	

Assignment No : 04

Title : Perform the data classification algorithm using any Classification algorithm.

Course Objective:

- To understand the concept of **classification** in machine learning.
- To implement a classification algorithm using Python.
- To evaluate the performance of a classification model using accuracy, precision, recall, and F1-score.

Course Outcome:

- Students will be able to Understand the working of classification algorithms.
- Students will be able to Preprocess and split data for training and testing.
- Students will be able to Train a classification model using **Logistic Regression, Decision Tree, Random Forest, or any other algorithm**.
- Students will be able to Evaluate model performance using appropriate metrics.

Prerequisite:

- Basic Python programming knowledge.
- Understanding of **pandas** for data handling.
- Familiarity with **scikit-learn** for implementing machine learning models.

Software and Hardware Requirements:

Python (3.x), Jupyter Notebook ,pandas, numpy for data manipulation, scikit-learn for classification algorithms, matplotlib, seaborn for visualization

Processor, RAM, Disk Space.

Theory:

What is Classification?

Classification is a supervised machine learning technique where the model learns from labeled data to categorize new instances into predefined classes. Examples include:

- Email spam detection (Spam/Not Spam)
- Disease diagnosis (Diabetic/Non-Diabetic)
- Credit risk analysis (High/Low risk)

Common Classification Algorithms:

1. **Logistic Regression** – Best for binary classification problems.
2. **Decision Tree** – Works well with structured data but prone to overfitting.
3. **Random Forest** – An ensemble technique that reduces overfitting.
4. **Support Vector Machine (SVM)** – Best for complex boundary separation.
5. **K-Nearest Neighbors (KNN)** – Classifies based on nearest neighbors.

What is the Random Forest Algorithm?

Random Forest is an **ensemble learning technique** that builds multiple decision trees and merges them for a more accurate and stable prediction. It works on the principle of **bagging (Bootstrap Aggregating)**, where:

1. **Multiple Decision Trees** are trained on different subsets of data.
2. Each tree makes a prediction, and the majority vote (for classification) is considered as the final output.

Why Random Forest?

- **Handles Overfitting**: Since multiple trees are trained, individual biases are reduced.
- **Works well with Missing Data**: Random Forest can handle missing values better than a single decision tree.
- **Can Handle Large Datasets**: Performs well on high-dimensional data.

Working of Random Forest Algorithm

1. **Bootstrap Sampling**: The dataset is randomly divided into multiple subsets.
2. **Decision Tree Training**: Each subset is used to train a separate decision tree.
3. **Feature Selection**: Random selection of features at each tree node enhances diversity.
4. **Majority Voting**: For classification, the most frequent class label across all trees is chosen as the final prediction.

Advantages of Random Forest:

- **Higher accuracy** than a single decision tree.
- **Robust to noise and outliers.**
- **Reduces the risk of overfitting.**

Limitations:

- Computationally intensive when the number of trees is very large.
- Less interpretable than a single decision tree.

Source Code :

1. Import Necessary Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

2. Load Dataset

```
# Load the Iris dataset
iris = sns.load_dataset("iris")
iris.head()
```

3. Data Preprocessing

```
# Encode target labels as numbers
iris['species'] = iris['species'].astype('category').cat.codes  # 'setosa'
-> 0, 'versicolor' -> 1, 'virginica' -> 2

# Define features (X) and target (y)
X = iris.drop(columns=['species'])
y = iris['species']

# Split data into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
```

4. Feature Scaling

```
# Standardizing the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

5. Train the Random Forest Classifier

```
# Initialize and train the Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)  # 100
trees in the forest
rf_model.fit(X_train, y_train)
```

6. Make Predictions

```
# Predict on test data
y_pred = rf_model.predict(X_test)
```

7. Evaluate Model Performance

```
# Accuracy Score
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Classification Report
print("Classification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt="d",
xticklabels=['Setosa', 'Versicolor', 'Virginica'], yticklabels=['Setosa',
'Versicolor', 'Virginica'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Conclusion : In this way, we implemented the Random Forest classification algorithm to efficiently classify data by combining multiple decision trees. We observed how ensemble learning improves accuracy, reduces overfitting, and handles large datasets effectively

Lab Assignment No.	05
Title	Perform the data clustering algorithm using any Clustering algorithm
Roll No.	
Class	BE AI & DS
Date Of Completion	
Subject	Computer Laboratory IV[417535]
Assessment Marks	
Assessor's Sign	

Assignment No : 05

Title : Perform the data clustering algorithm using any Clustering algorithm.

Course Objective:

- To understand the concept of **clustering** in machine learning.
- To implement a clustering algorithm using Python.
- To analyze and visualize clustered data.

Course Outcome:

- Students will be able to Understand how clustering groups similar data points.
- Students will be able to Implement the **K-Means clustering algorithm** on a dataset.
- Students will be able to Evaluate and visualize clusters using scatter plots and other techniques.

Prerequisite:

- Basic Python programming knowledge.
- Understanding of **pandas** for data handling.
- Familiarity with **matplotlib** and **seaborn** for visualization.
- Knowledge of unsupervised learning concepts.

Software & Hardware Requirements :

- **Software:** Python (3.x), Jupyter Notebook/VSCode/PyCharm, pandas, numpy, scikit-learn, matplotlib, seaborn.
- **Hardware:** Intel Core i3 or higher, 4GB+ RAM, 500MB+ disk space for dataset storage.

Theory:

Introduction to Clustering

Clustering is an **unsupervised machine learning technique** that groups similar data points into clusters based on their characteristics. Unlike classification, clustering does not require labeled data. It is widely used in various applications, such as:

- **Customer Segmentation** – Grouping customers based on purchasing behavior.
- **Image Compression** – Reducing image size by clustering similar pixels.
- **Anomaly Detection** – Identifying fraudulent transactions or unusual patterns.

What is K-Means Clustering?

K-Means is one of the most popular clustering algorithms that partitions data into **K clusters** by minimizing the distance between data points and their assigned cluster centroid.

Working of K-Means Algorithm

1. **Select the number of clusters (K).**
2. **Initialize K cluster centroids randomly.**
3. **Assign each data point to the nearest centroid.**
4. **Update centroids** by computing the mean of all points in each cluster.
5. **Repeat steps 3 and 4** until centroids stabilize (i.e., no further changes in cluster assignments).

Choosing the Optimal Number of Clusters (K)

The **Elbow Method** is commonly used to determine the ideal value of K. It plots the **Within-Cluster Sum of Squares (WCSS)** for different values of K and identifies the point where the decrease in WCSS slows down, forming an "elbow" in the graph.

Advantages of K-Means Clustering

- **Simple and fast** for large datasets.
- **Efficient** when clusters are well-separated.
- **Scalable** to high-dimensional data.

Limitations of K-Means Clustering

- **Sensitive to outliers** and noisy data.
- **Choosing K** requires prior knowledge or testing multiple values.
- **May fail** if clusters are of different sizes or densities.

Applications of K-Means Clustering

- Market segmentation for personalized marketing strategies.
- Image segmentation in computer vision.
- Document clustering for text mining and recommendation systems.

Source Code :

1. Import Necessary Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
```

2. Load Dataset

```
# Load the Iris dataset

iris = sns.load_dataset("iris")
print(iris.head())  # Display first 5 rows
```

3. Data Preprocessing

```
# Drop the categorical 'species' column
X = iris.drop(columns=['species'])

# Standardizing the features (scaling)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

4. Find Optimal K Using the Elbow Method

```
# Elbow Method to determine the best number of clusters
inertia = []
K_range = range(1, 11)  # Check for K=1 to K=10

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# Plot Elbow Curve
plt.figure(figsize=(8, 5))
plt.plot(K_range, inertia, marker='o', linestyle='--', color='b')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia (Within-Cluster Sum of Squares)')
plt.title('Elbow Method for Optimal K')
plt.show()
```

5. Train the K-Means Model

```
# Apply K-Means with 3 clusters
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
clusters = kmeans.fit_predict(X_scaled)

# Assign clusters to a new column in the dataset
iris['Cluster'] = clusters

# Debug check
print(iris.head())  # Verify the 'Cluster' column is added
```

6. Evaluate Clustering Using Silhouette Score

```
silhouette_avg = silhouette_score(X_scaled, clusters)
print(f'Silhouette Score: {silhouette_avg:.2f}')
```

7. Reduce to 2D Using PCA for Visualization

```
# Apply PCA to reduce dimensions from 4D to 2D
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Convert to DataFrame for visualization
pca_df = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
pca_df['Cluster'] = clusters  # Assign clusters
```

```
# Scatter plot of PCA components with clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x=pca_df['PC1'], y=pca_df['PC2'], hue=pca_df['Cluster'],
palette='viridis', s=100)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('K-Means Clustering (PCA Reduced)')
plt.legend(title='Cluster')
plt.show()
```

Conclusion : In this way, we implemented K-Means clustering to group similar data points into clusters. The Elbow Method helped determine the optimal number of clusters **(**K**)**, and visualization provided insights into cluster distribution.