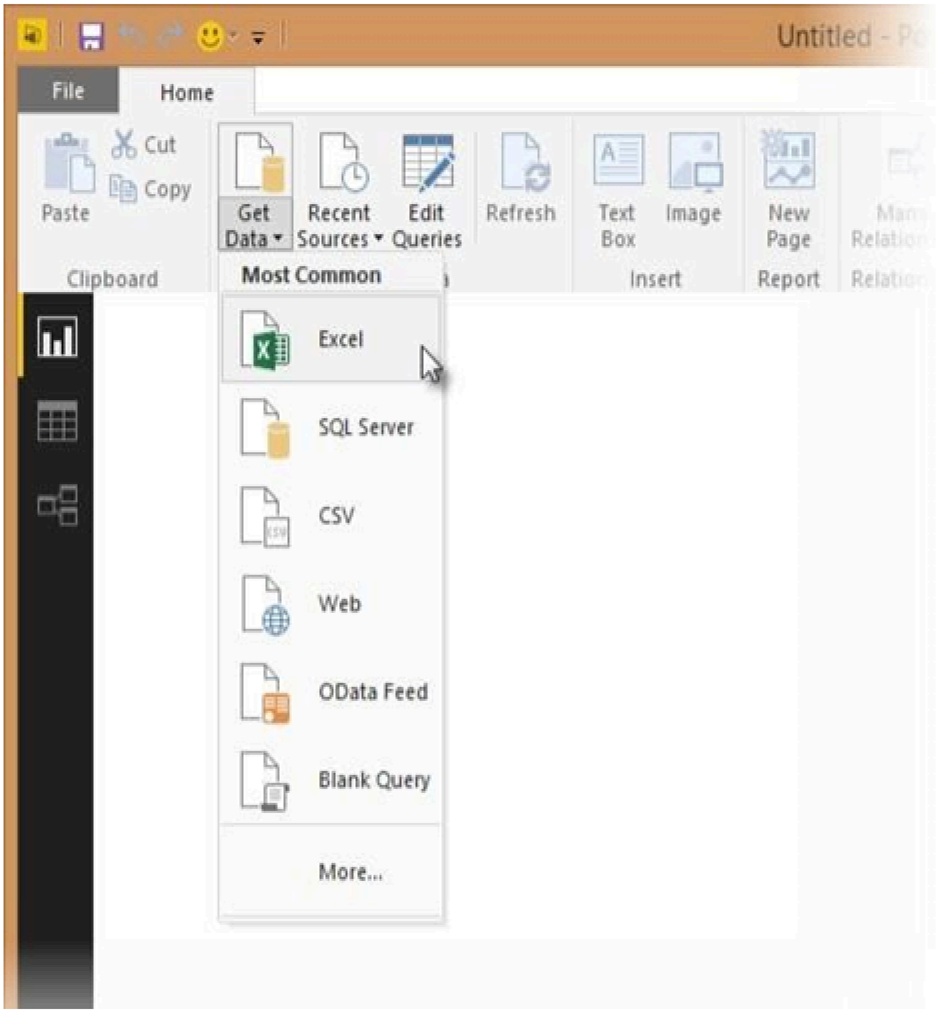
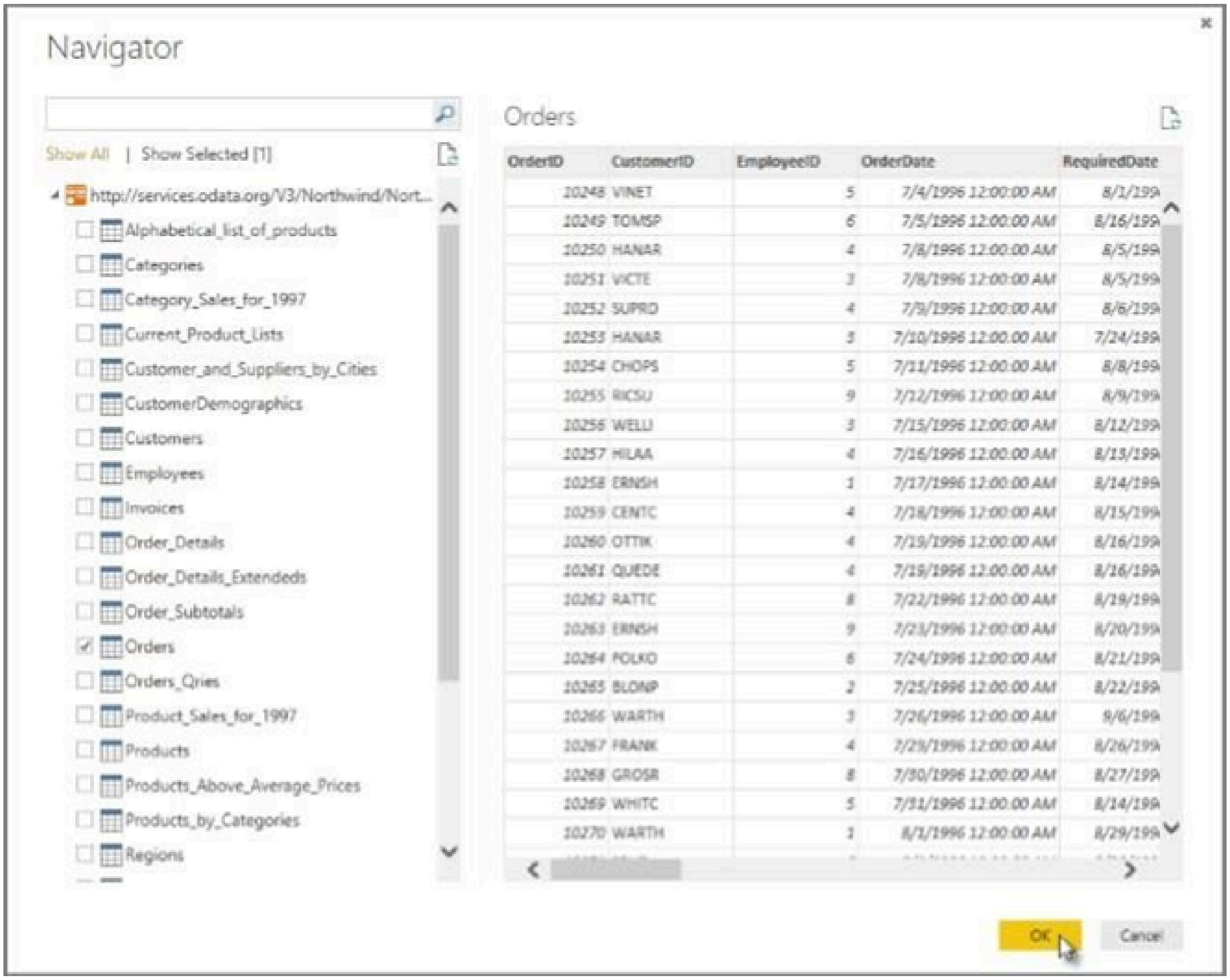
ASSIGNMENT 01:





Navigator

Show All | Show Selected (1)

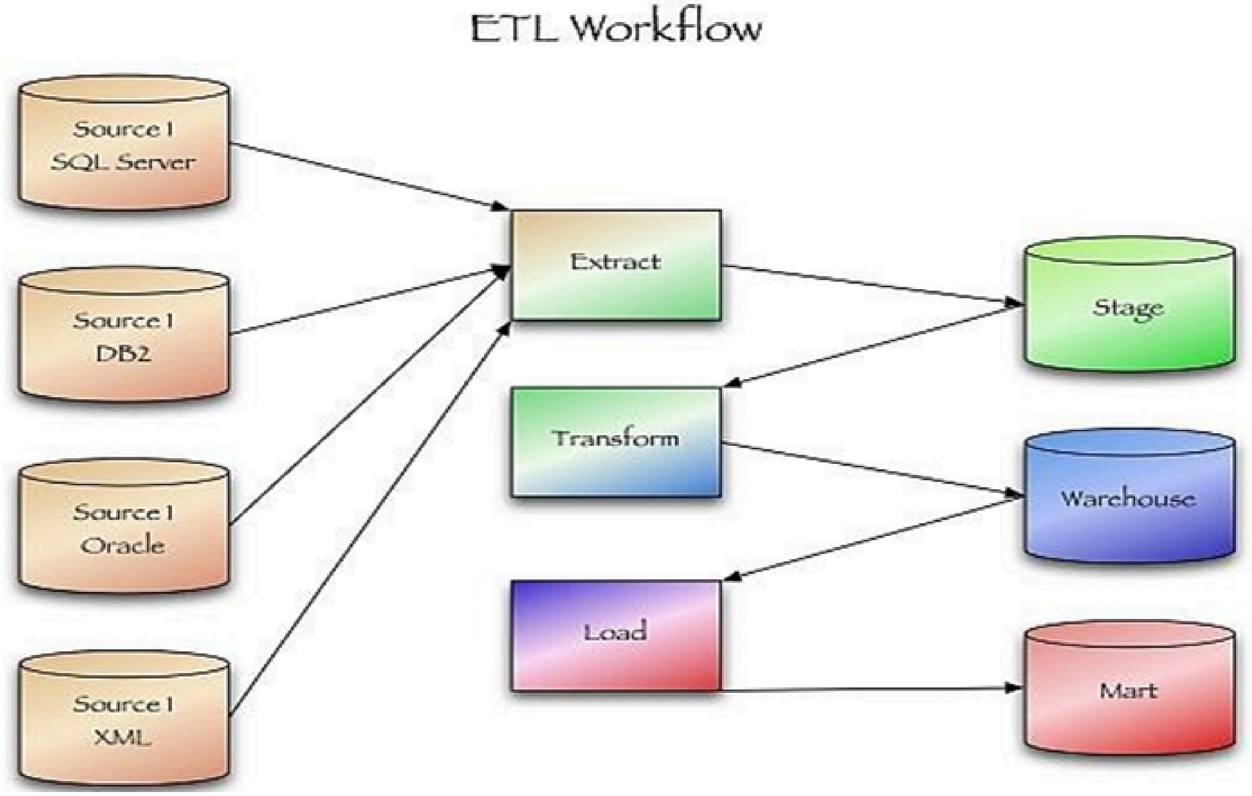
http://services.odata.org/V3/Northwind/Nort...

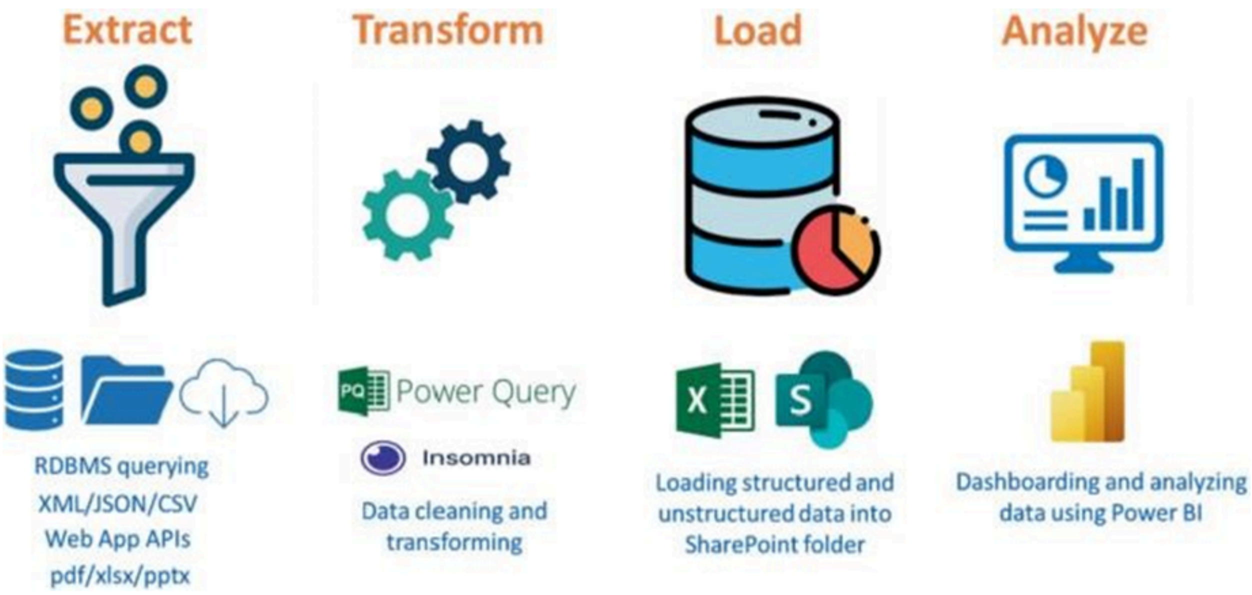
- □ Alphabetical_list_of_products
- □ Categories
- □ Category_Sales_for_1997
- □ Current_Product_Lists
- □ Customer_and_Suppliers_by_Cities
- □ CustomerDemographics
- □ Customers
- □ Employees
- □ Invoices
- □ Order_Details
- □ Order_Details_Extendeds
- □ Order_Subtotals
- ☑ Orders
- □ Orders_Qries
- □ Product_Sales_for_1997
- □ Products
- □ Products_Above_Average_Prices
- □ Products_by_Categories
- □ Regions

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/199
10249	TOMSP	6	7/5/1996 12:00:00 AM	8/16/199
10250	HANAR	4	7/8/1996 12:00:00 AM	8/5/199
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/199
10252	SUPRD	4	7/9/1996 12:00:00 AM	8/6/199
10253	HANAR	3	7/10/1996 12:00:00 AM	7/24/199
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/199
10255	RICSU	9	7/12/1996 12:00:00 AM	8/9/199
10256	WELLI	3	7/15/1996 12:00:00 AM	8/12/199
10257	HILAA	4	7/16/1996 12:00:00 AM	8/13/199
10258	ERNSH	1	7/17/1996 12:00:00 AM	8/14/199
10259	CENTC	4	7/18/1996 12:00:00 AM	8/15/199
10260	OTTIK	4	7/19/1996 12:00:00 AM	8/16/199
10261	QUEDE	4	7/19/1996 12:00:00 AM	8/16/199
10262	RATTC	8	7/22/1996 12:00:00 AM	8/19/199
10263	ERNSH	9	7/23/1996 12:00:00 AM	8/20/199
10264	FOLKO	6	7/24/1996 12:00:00 AM	8/21/199
10265	BLONP	2	7/25/1996 12:00:00 AM	8/22/199
10266	WARTH	3	7/26/1996 12:00:00 AM	9/6/199
10267	FRANK	4	7/29/1996 12:00:00 AM	8/26/199
10268	GROSR	8	7/30/1996 12:00:00 AM	8/27/199
10269	WHITC	5	7/31/1996 12:00:00 AM	8/14/199
10270	WARTH	1	8/1/1996 12:00:00 AM	8/29/199

ASSIGNMENT 02:





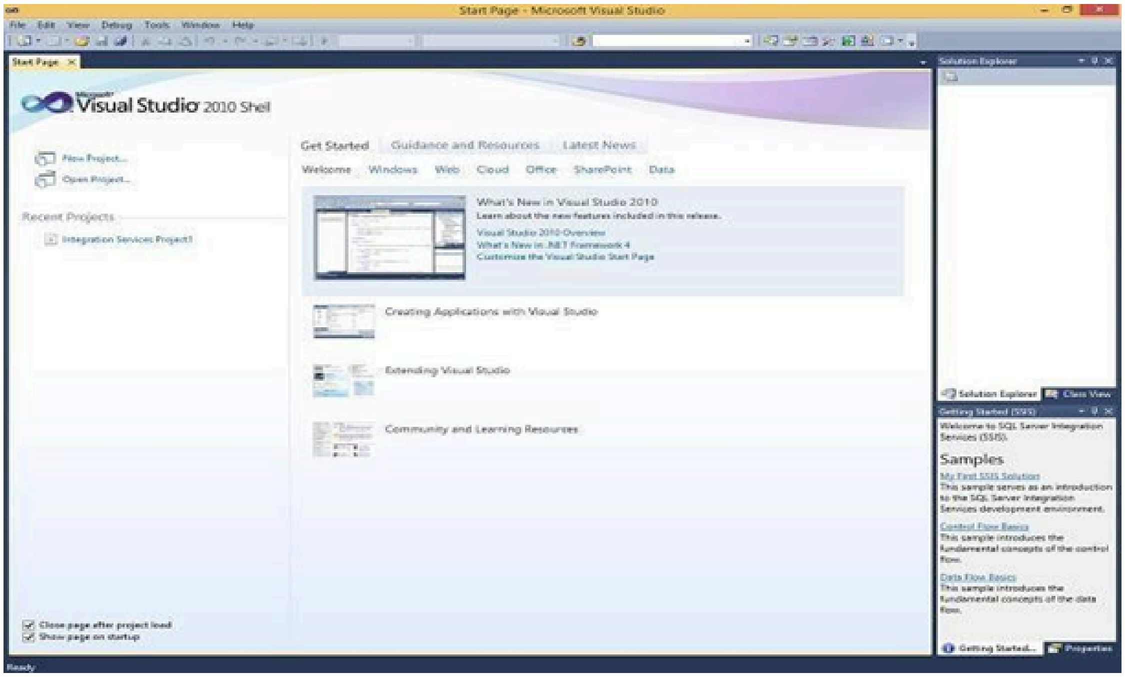
RDBMS querying
XML/JSON/CSV
Web App APIs
pdf/xlsx/pptx

Data cleaning and transforming

Loading structured and unstructured data into SharePoint folder

Dashboarding and analyzing data using Power BI

ASSIGNMENT 03:



Integration Services Project1

What's New in Visual Studio 2010

Learn about the new features included in this release.

Visual Studio 2010 Overview

What's New in .NET Framework 4

Customize the Visual Studio Start Page

Creating Applications with Visual Studio

Extending Visual Studio

Community and Learning Resources

Welcome to SQL Server Integration Services (SSIS).

Samples

My First SSIS Solution

This sample serves as an introduction to the SQL Server Integration Services development environment.

Control Flow Basics

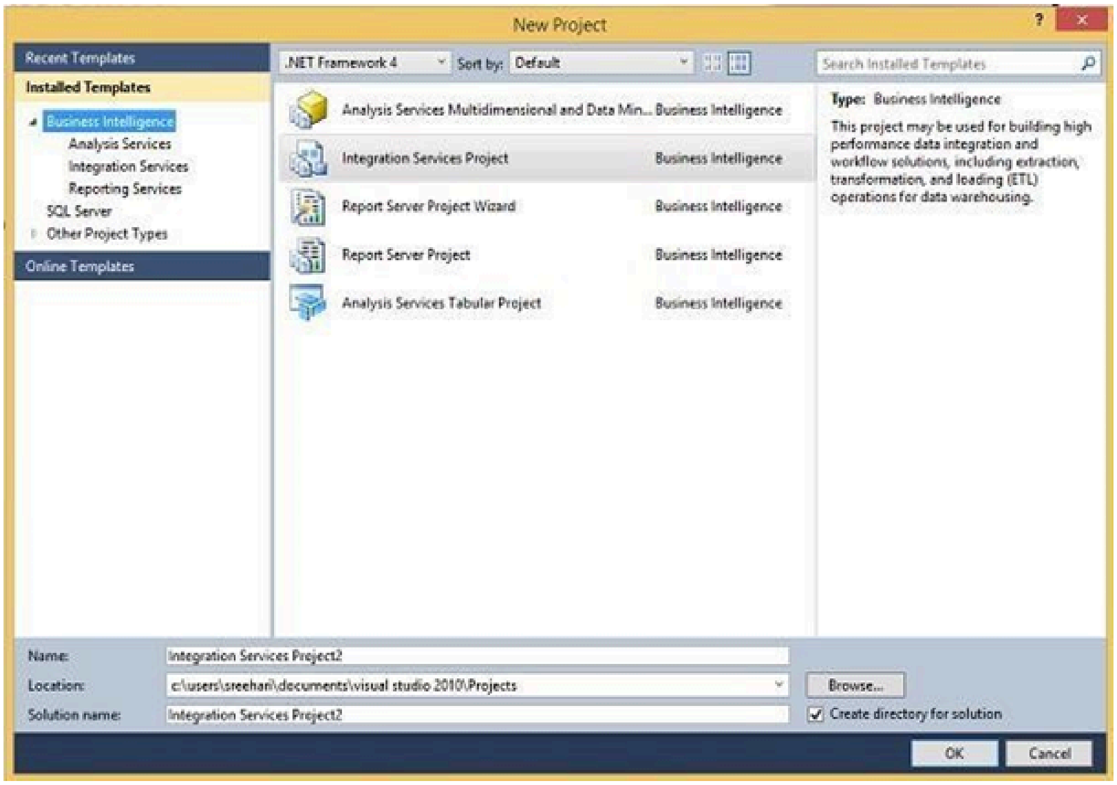
This sample introduces the fundamental concepts of the control flow.

Data Flow Basics

This sample introduces the fundamental concepts of the data flow.

☑ Close page after project load

☑ Show page on startup



.NET Framework 4   Sort by: Default

Analysis Services Multidimensional and Data Min...	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

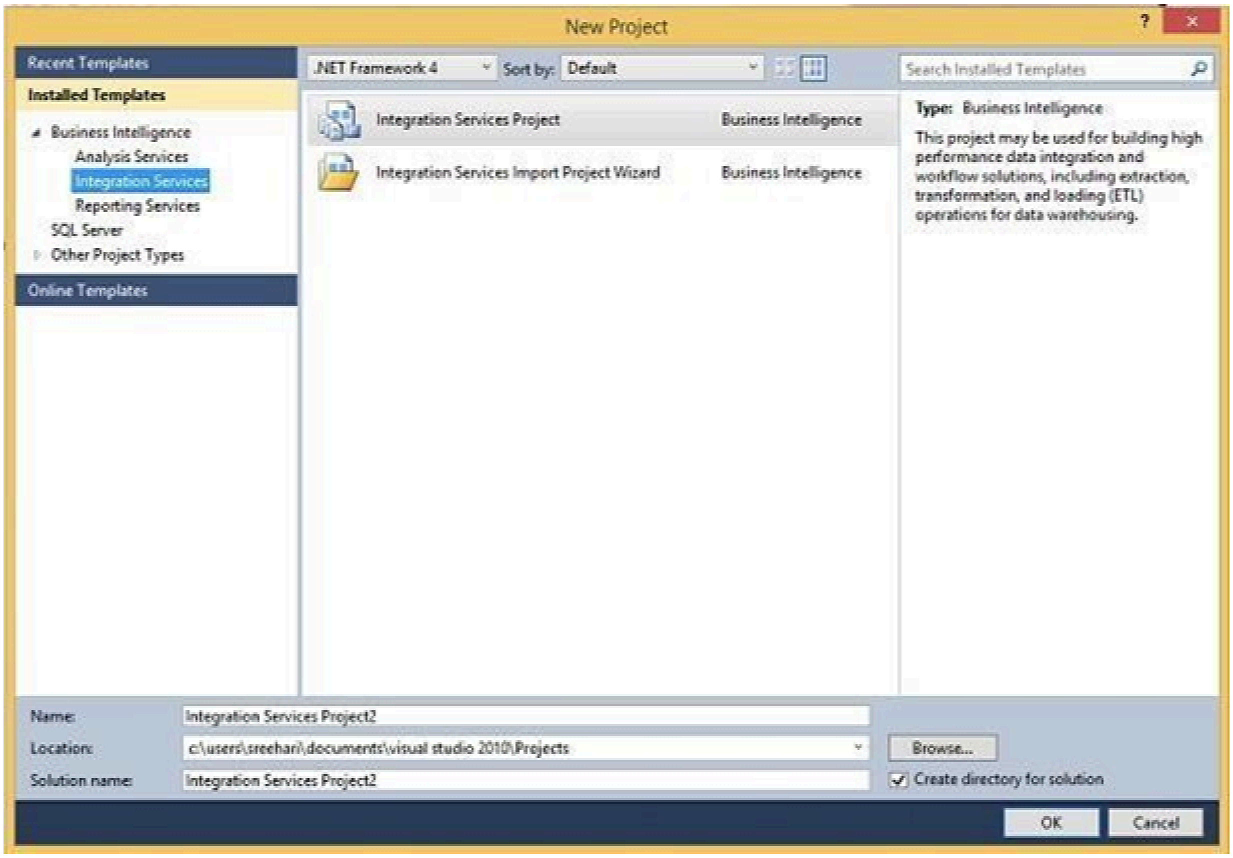
This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution



Integration Services Project	Business Intelligence
Integration Services Import Project Wizard	Business Intelligence

Type: Business Intelligence

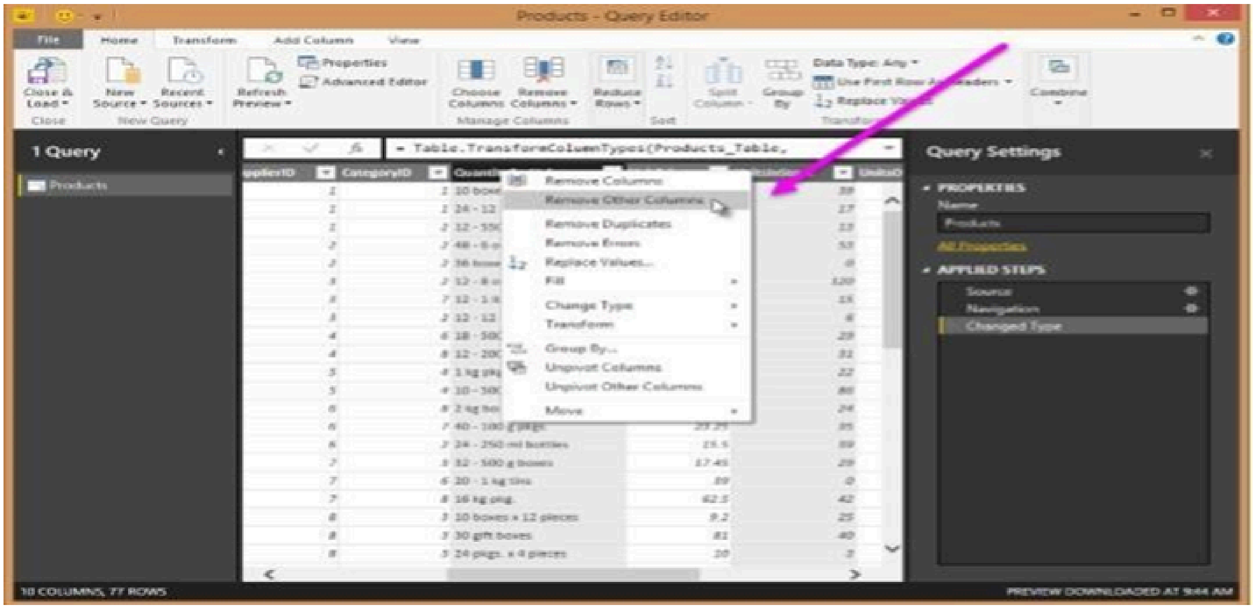
This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution



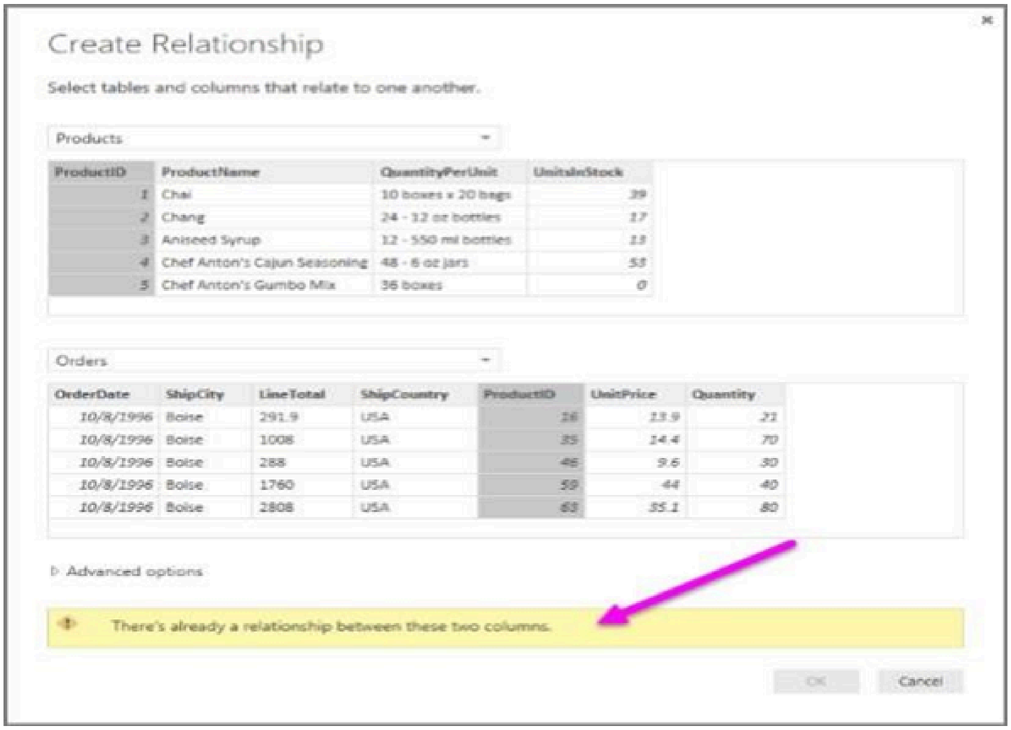
```
= Table.TransformColumnTypes(Products_Table,
```

pplierID	CategoryID	Quanti		
1	1	10 boxe		39
1	1	24 - 12		17
1	2	12 - 55(		13
2	2	48 - 6 o		53
2	2	36 boxe		0
3	2	12 - 8 o		120
3	7	12 - 1 k		15
3	2	12 - 12		6
4	6	18 - 50(		29
4	8	12 - 20(		31
5	4	1 kg pkg		22
5	4	10 - 50(		86
6	8	2 kg box		24
6	7	40 - 100 g pkgs.	23.25	35
6	2	24 - 250 ml bottles	15.5	39
7	3	32 - 500 g boxes	17.45	29
7	6	20 - 1 kg tins	39	0
7	8	16 kg pkg.	62.5	42
8	3	10 boxes x 12 pieces	9.2	25
8	3	30 gift boxes	81	40
8	3	24 pkgs. x 4 pieces	10	3

Name: Products

10 COLUMNS, 77 ROWS

PREVIEW DOWNLOADED AT 9:44 AM



Create Relationship

Select tables and columns that relate to one another.

Products

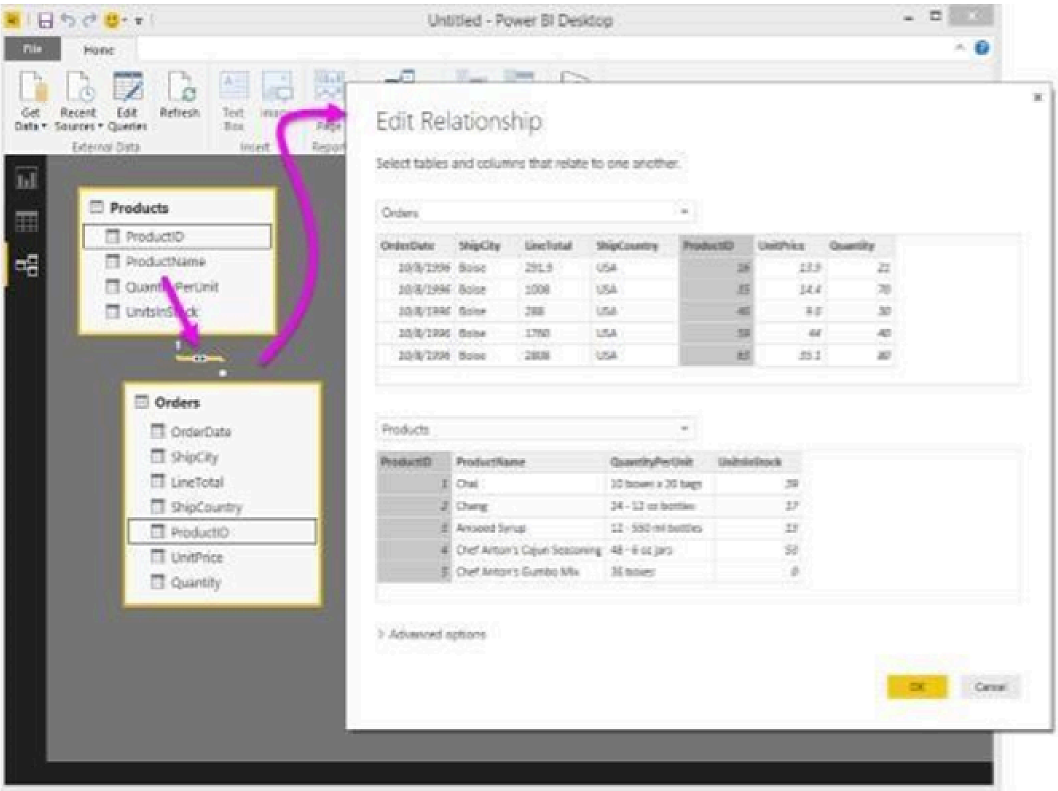
ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

Orders

OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	16	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	46	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	65	35.1	80

▷ Advanced options

There's already a relationship between these two columns.



Edit Relationship

Select tables and columns that relate to one another.

Orders

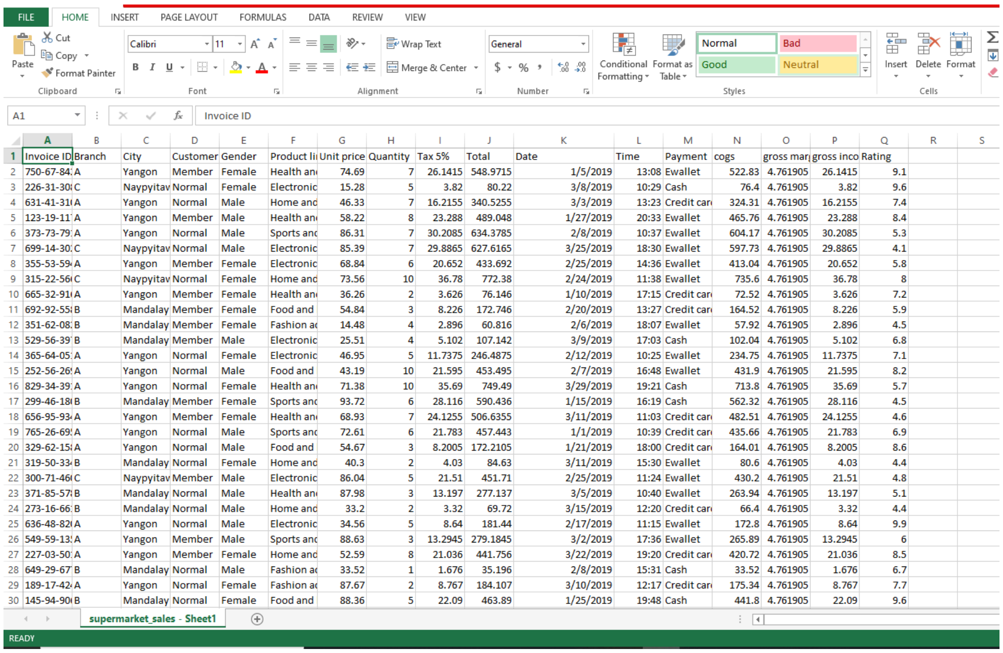
OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	16	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	46	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	65	35.1	80

Products

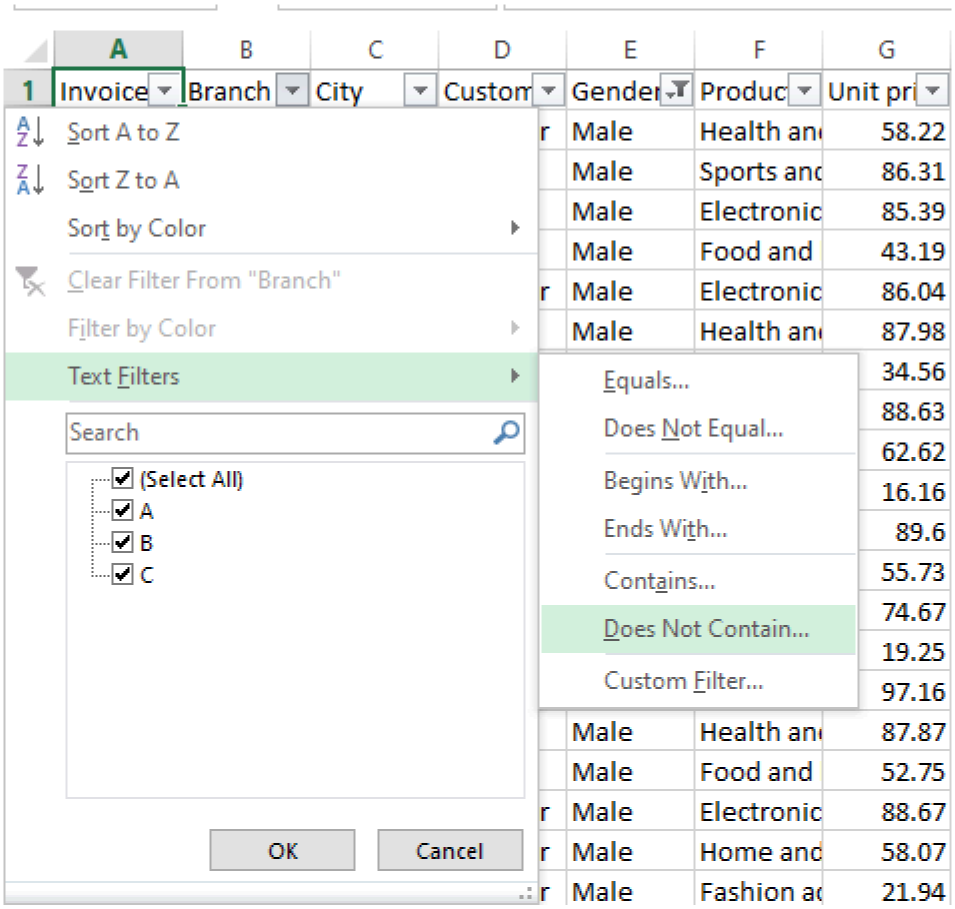
ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

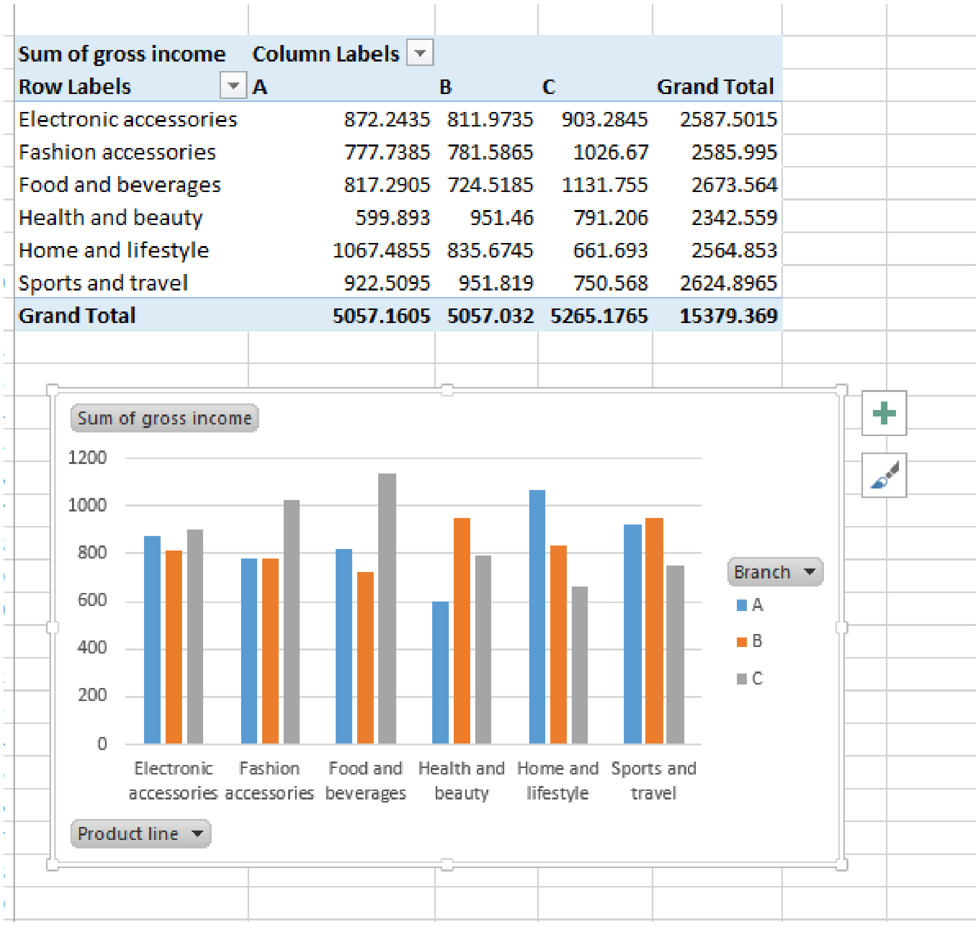
▷ Advanced options

ASSIGNMENT 04:

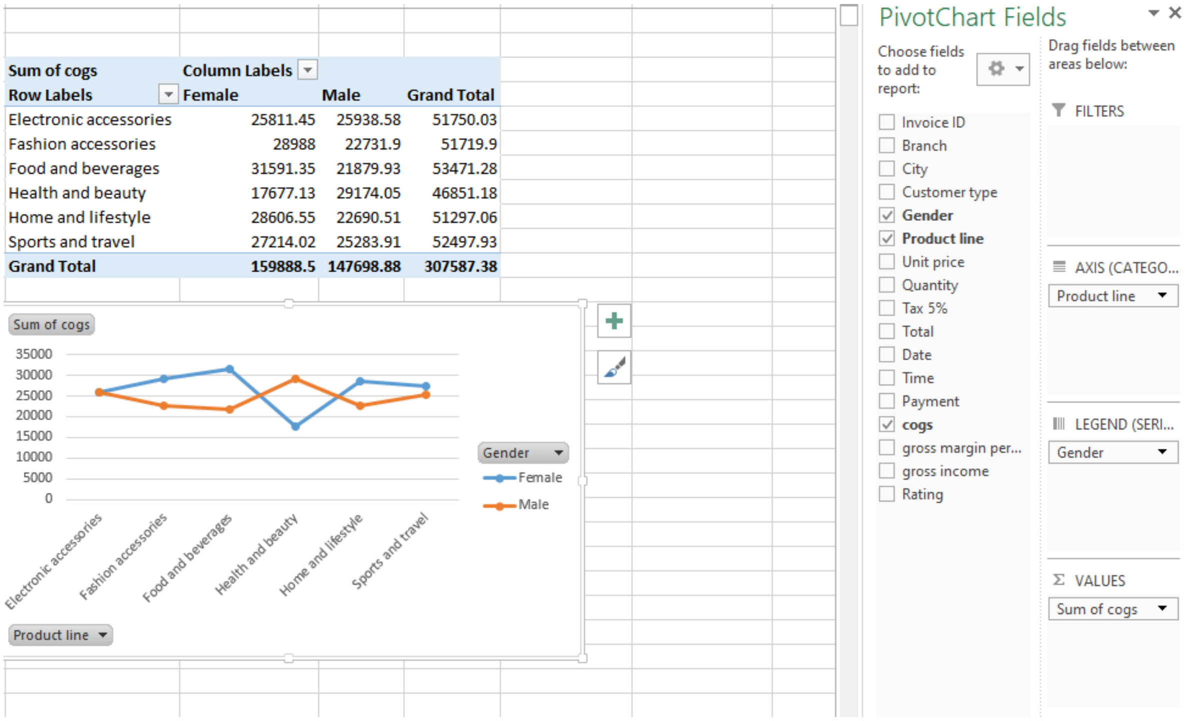


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Invoice ID	Branch	City	Customer	Gender	Product li	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross mar	gross inco	Rating
2	750-67-84	A	Yangon	Member	Female	Health an	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
3	226-31-30	C	Naypyitav	Normal	Female	Electronic	15.28	5	3.82	80.22	3/8/2019	10:29	Cash	76.4	4.761905	3.82	9.6
4	631-41-31	A	Yangon	Normal	Male	Home and	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit car	324.31	4.761905	16.2155	7.4
5	123-19-11	A	Yangon	Member	Male	Health an	58.22	8	23.288	489.048	1/27/2019	20:33	Ewallet	465.76	4.761905	23.288	8.4
6	373-73-79	A	Yangon	Normal	Male	Sports and	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3
7	699-14-30	C	Naypyitav	Normal	Male	Electronic	85.39	7	29.8865	627.6165	3/25/2019	18:30	Ewallet	597.73	4.761905	29.8865	4.1
8	355-53-59	A	Yangon	Member	Female	Electronic	68.84	6	20.652	433.692	2/25/2019	14:36	Ewallet	413.04	4.761905	20.652	5.8
9	315-22-56	C	Naypyitav	Normal	Female	Home and	73.56	10	36.78	772.38	2/24/2019	11:38	Ewallet	735.6	4.761905	36.78	8
10	665-32-91	A	Yangon	Member	Female	Health an	36.26	2	3.626	76.146	1/10/2019	17:15	Credit car	72.52	4.761905	3.626	7.2
11	692-92-55	B	Mandalay	Member	Female	Food and	54.84	3	8.226	172.746	2/20/2019	13:27	Credit car	164.52	4.761905	8.226	5.9
12	351-62-08	B	Mandalay	Member	Female	Fashion ac	14.48	4	2.896	60.816	2/6/2019	18:07	Ewallet	57.92	4.761905	2.896	4.5
13	529-56-39	B	Mandalay	Member	Male	Electronic	25.51	4	5.102	107.142	3/9/2019	17:03	Cash	102.04	4.761905	5.102	6.8
14	365-64-05	A	Yangon	Normal	Female	Electronic	46.95	5	11.7375	246.4875	2/12/2019	10:25	Ewallet	234.75	4.761905	11.7375	7.1
15	252-56-26	A	Yangon	Normal	Male	Food and	43.19	10	21.595	453.495	2/7/2019	16:48	Ewallet	431.9	4.761905	21.595	8.2
16	829-34-39	A	Yangon	Normal	Female	Health an	71.38	10	35.69	749.49	3/29/2019	19:21	Cash	713.8	4.761905	35.69	5.7
17	299-46-18	B	Mandalay	Member	Female	Sports and	93.72	6	28.116	590.436	1/15/2019	16:19	Cash	562.32	4.761905	28.116	4.5
18	656-95-93	A	Yangon	Member	Female	Health an	68.93	7	24.1255	506.6355	3/11/2019	11:03	Credit car	482.51	4.761905	24.1255	4.6
19	765-26-69	A	Yangon	Normal	Male	Sports and	72.61	6	21.783	457.443	1/1/2019	10:39	Credit car	435.66	4.761905	21.783	6.9
20	329-62-15	A	Yangon	Normal	Male	Food and	54.67	3	8.2005	172.2105	1/21/2019	18:00	Credit car	164.01	4.761905	8.2005	8.6
21	319-50-33	B	Mandalay	Normal	Female	Home and	40.3	2	4.03	84.63	3/11/2019	15:30	Ewallet	80.6	4.761905	4.03	4.4
22	300-71-46	C	Naypyitav	Member	Male	Electronic	86.04	5	21.51	451.71	2/25/2019	11:24	Ewallet	430.2	4.761905	21.51	4.8
23	371-85-57	B	Mandalay	Normal	Male	Health an	87.98	3	13.197	277.137	3/5/2019	10:40	Ewallet	263.94	4.761905	13.197	5.1
24	273-16-66	B	Mandalay	Normal	Male	Home and	33.2	2	3.32	69.72	3/15/2019	12:20	Credit car	66.4	4.761905	3.32	4.4
25	636-48-82	A	Yangon	Normal	Male	Electronic	34.56	5	8.64	181.44	2/17/2019	11:15	Ewallet	172.8	4.761905	8.64	9.9
26	549-59-13	A	Yangon	Member	Male	Sports and	88.63	3	13.2945	279.1845	3/2/2019	17:36	Ewallet	265.89	4.761905	13.2945	6
27	227-03-50	A	Yangon	Member	Female	Home and	52.59	8	21.036	441.756	3/22/2019	19:20	Credit car	420.72	4.761905	21.036	8.5
28	649-29-67	B	Mandalay	Normal	Male	Fashion ac	33.52	1	1.676	35.196	2/8/2019	15:31	Cash	33.52	4.761905	1.676	6.7
29	189-17-42	A	Yangon	Normal	Female	Fashion ac	87.67	2	8.767	184.107	3/10/2019	12:17	Credit car	175.34	4.761905	8.767	7.7
30	145-94-90	B	Mandalay	Normal	Female	Food and	88.36	5	22.09	463.89	1/25/2019	19:48	Cash	441.8	4.761905	22.09	9.6





Sum of gross income	Column Labels			
Row Labels	A	B	C	Grand Total
Electronic accessories	872.2435	811.9735	903.2845	2587.5015
Fashion accessories	777.7385	781.5865	1026.67	2585.995
Food and beverages	817.2905	724.5185	1131.755	2673.564
Health and beauty	599.893	951.46	791.206	2342.559
Home and lifestyle	1067.4855	835.6745	661.693	2564.853
Sports and travel	922.5095	951.819	750.568	2624.8965
Grand Total	**5057.1605**	**5057.032**	**5265.1765**	**15379.369**



Sum of cogs	Column Labels		
Row Labels	Female	Male	Grand Total
Electronic accessories	25811.45	25938.58	51750.03
Fashion accessories	28988	22731.9	51719.9
Food and beverages	31591.35	21879.93	53471.28
Health and beauty	17677.13	29174.05	46851.18
Home and lifestyle	28606.55	22690.51	51297.06
Sports and travel	27214.02	25283.91	52497.93
Grand Total	**159888.5**	**147698.88**	**307587.38**

PivotChart Fields

Choose fields to add to report:

- ☐ Invoice ID
- ☐ Branch
- ☐ City
- ☐ Customer type
- ☑ **Gender**
- ☑ **Product line**
- ☐ Unit price
- ☐ Quantity
- ☐ Tax 5%
- ☐ Total
- ☐ Date
- ☐ Time
- ☐ Payment
- ☑ **cogs**
- ☐ gross margin per...
- ☐ gross income
- ☐ Rating

Drag fields between areas below:

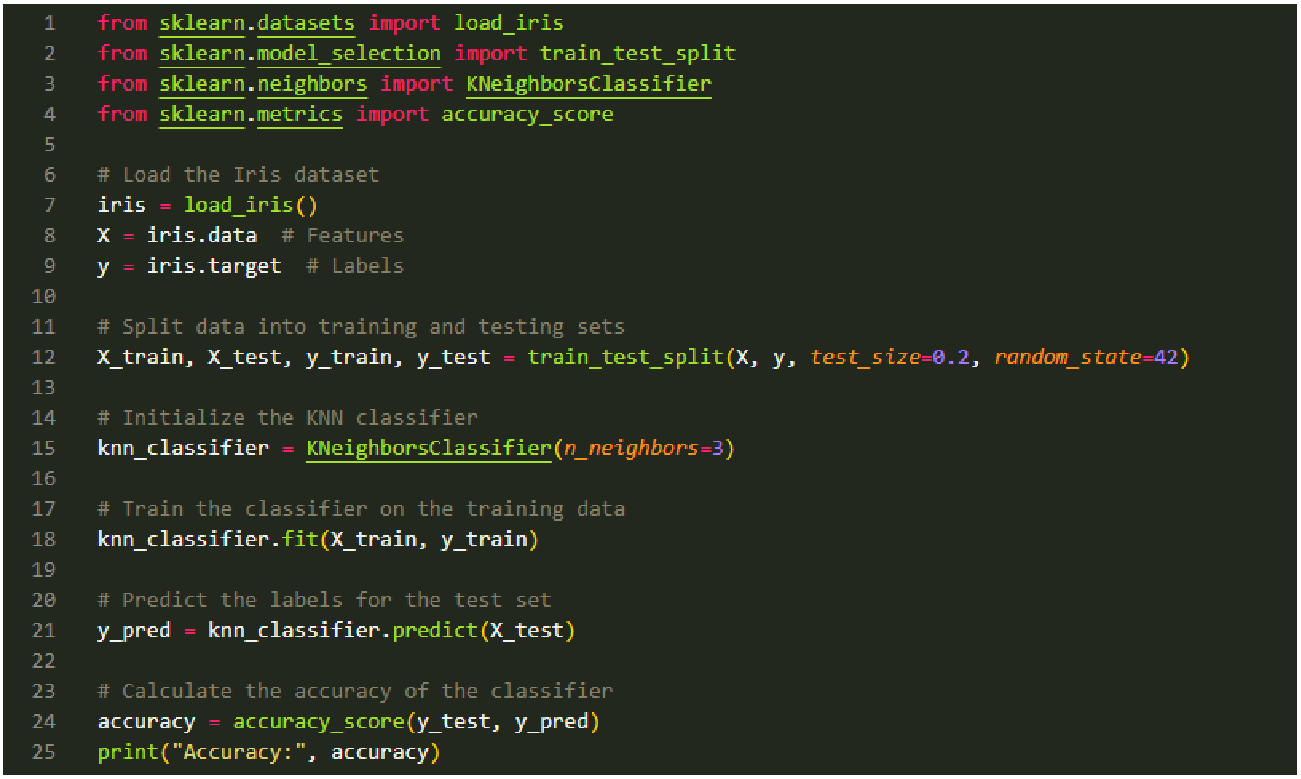
FILTERS

AXIS (CATEGO... : Product line

LEGEND (SERI... : Gender

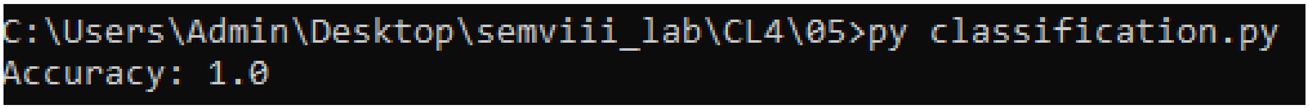
Σ VALUES: Sum of cogs

ASSIGNMENT 05 (code):



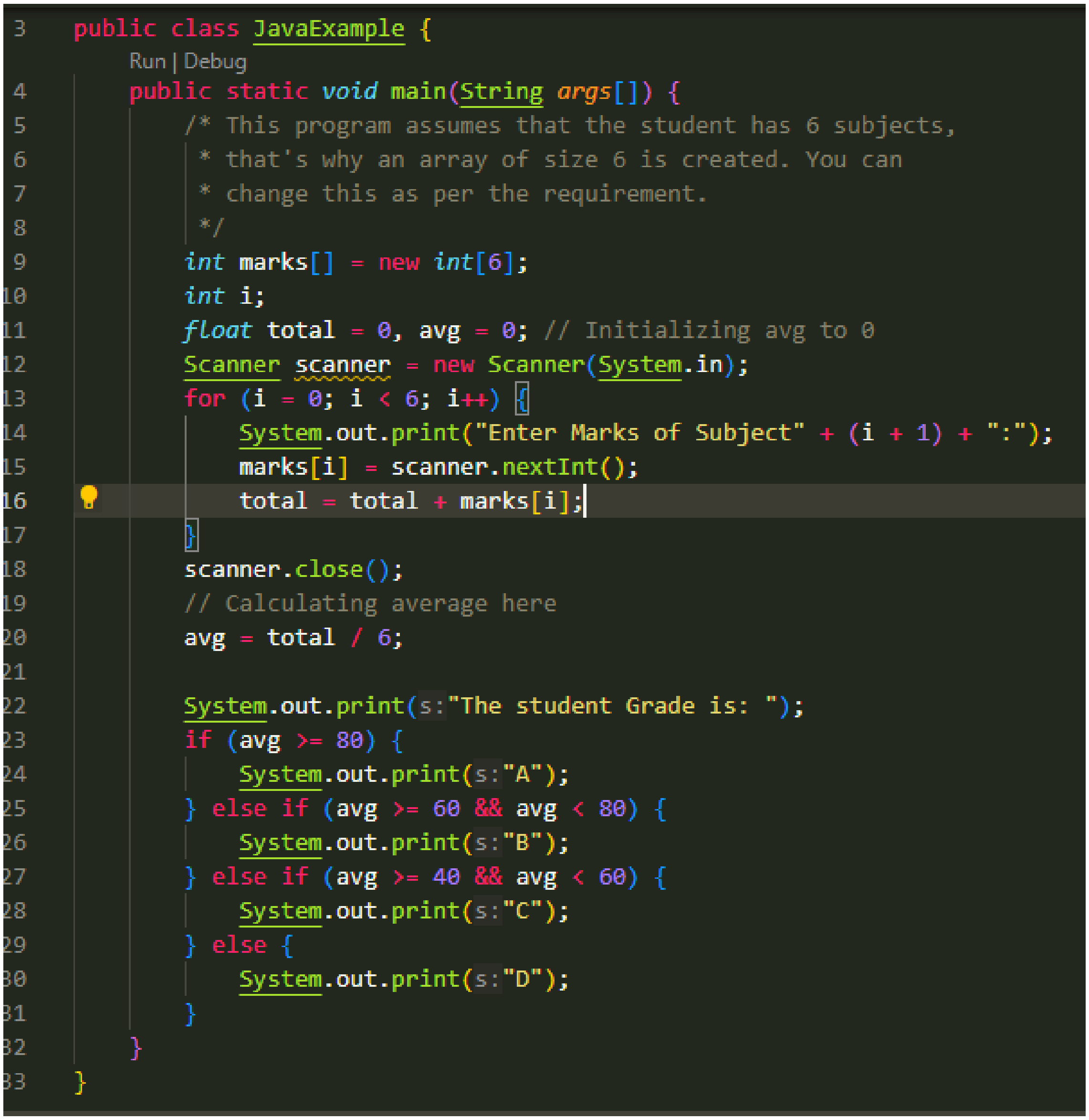
```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Load the Iris dataset
iris = load_iris()
X = iris.data  # Features
y = iris.target  # Labels

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

# Initialize the KNN classifier
knn_classifier = KNeighborsClassifier(n_neighbors=3)

# Train the classifier on the training data
knn_classifier.fit(X_train, y_train)

# Predict the labels for the test set
y_pred = knn_classifier.predict(X_test)

# Calculate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Output:



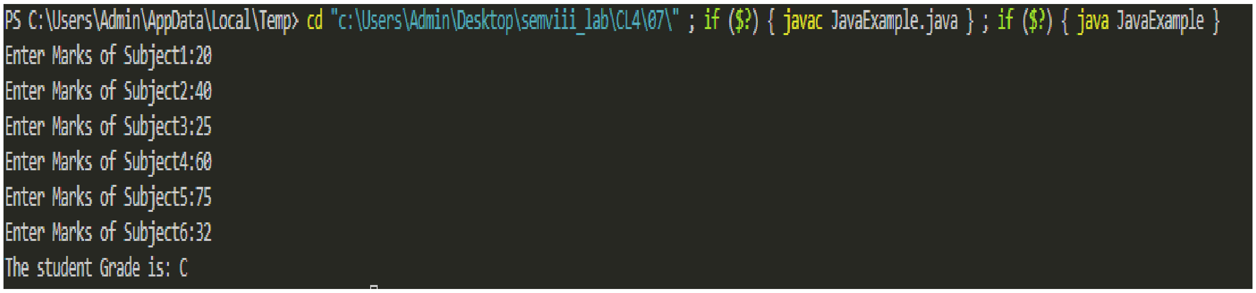
```
C:\Users\Admin\Desktop\semviii_lab\CL4\05>py classification.py
Accuracy: 1.0
```

ASSIGNMENT 07 (code):



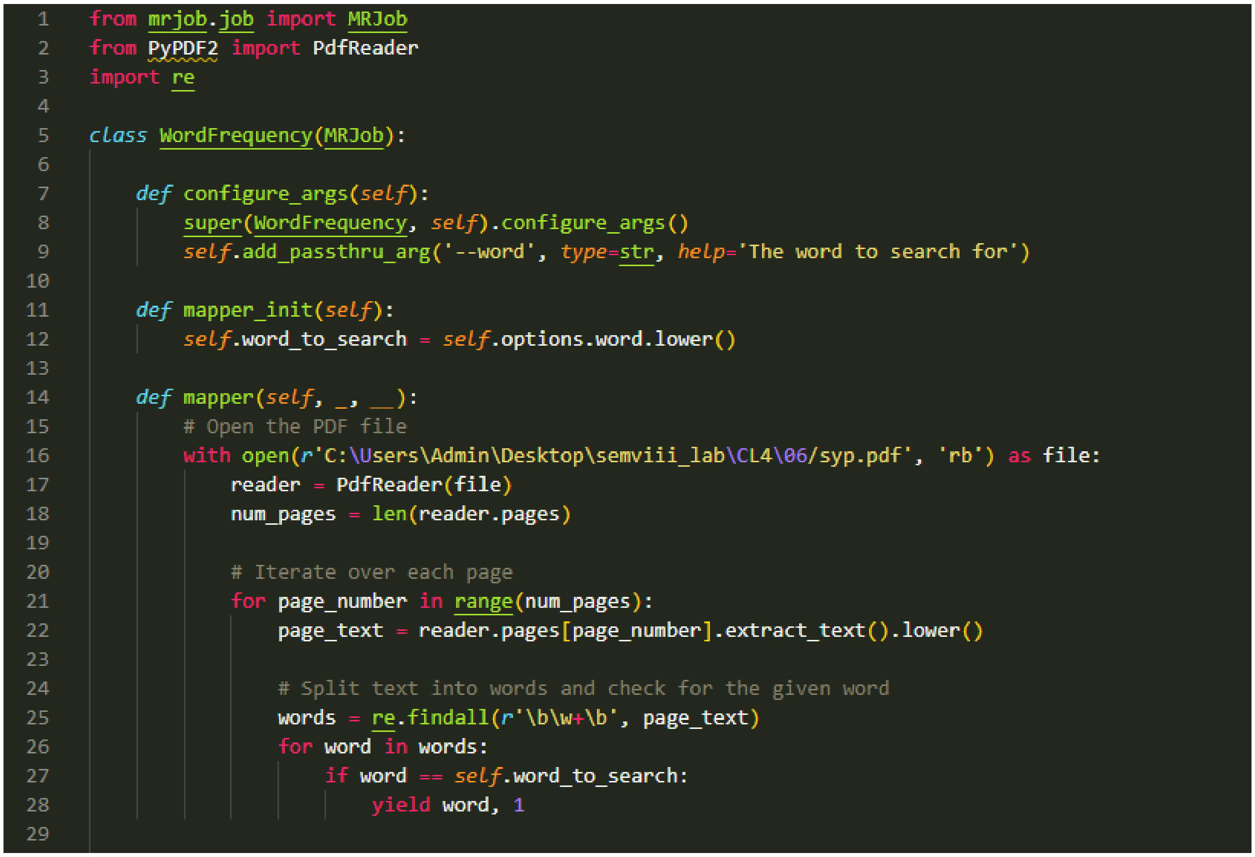
```java
public class JavaExample {
    public static void main(String args[]) {
        /* This program assumes that the student has 6 subjects,
         * that's why an array of size 6 is created. You can
         * change this as per the requirement.
         */
        int marks[] = new int[6];
        int i;
        float total = 0, avg = 0; // Initializing avg to 0
        Scanner scanner = new Scanner(System.in);
        for (i = 0; i < 6; i++) {
            System.out.print("Enter Marks of Subject" + (i + 1) + ":");
            marks[i] = scanner.nextInt();
            total = total + marks[i];
        }
        scanner.close();
        // Calculating average here
        avg = total / 6;

        System.out.print(s:"The student Grade is: ");
        if (avg >= 80) {
            System.out.print(s:"A");
        } else if (avg >= 60 && avg < 80) {
            System.out.print(s:"B");
        } else if (avg >= 40 && avg < 60) {
            System.out.print(s:"C");
        } else {
            System.out.print(s:"D");
        }
    }
}
```

Output:

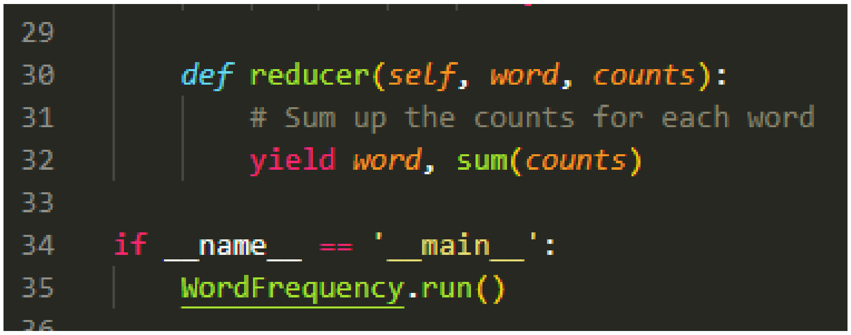


```
PS C:\Users\Admin\AppData\Local\Temp> cd "c:\Users\Admin\Desktop\semviii_lab\CL4\07\" ; if ($?) { javac JavaExample.java } ; if ($?) { java JavaExample }
Enter Marks of Subject1:20
Enter Marks of Subject2:40
Enter Marks of Subject3:25
Enter Marks of Subject4:60
Enter Marks of Subject5:75
Enter Marks of Subject6:32
The student Grade is: C
```

ASSIGNMENT 06 (code):

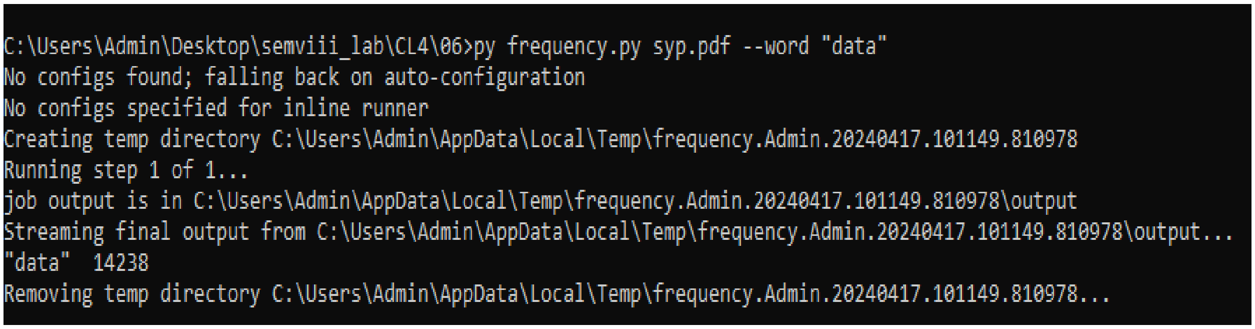


```python
from mrjob.job import MRJob
from PyPDF2 import PdfReader
import re

class WordFrequency(MRJob):

    def configure_args(self):
        super(WordFrequency, self).configure_args()
        self.add_passthru_arg('--word', type=str, help='The word to search for')

    def mapper_init(self):
        self.word_to_search = self.options.word.lower()

    def mapper(self, _, __):
        # Open the PDF file
        with open(r'C:\Users\Admin\Desktop\semviii_lab\CL4\06/syp.pdf', 'rb') as file:
            reader = PdfReader(file)
            num_pages = len(reader.pages)

            # Iterate over each page
            for page_number in range(num_pages):
                page_text = reader.pages[page_number].extract_text().lower()

                # Split text into words and check for the given word
                words = re.findall(r'\b\w+\b', page_text)
                for word in words:
                    if word == self.word_to_search:
                        yield word, 1

```



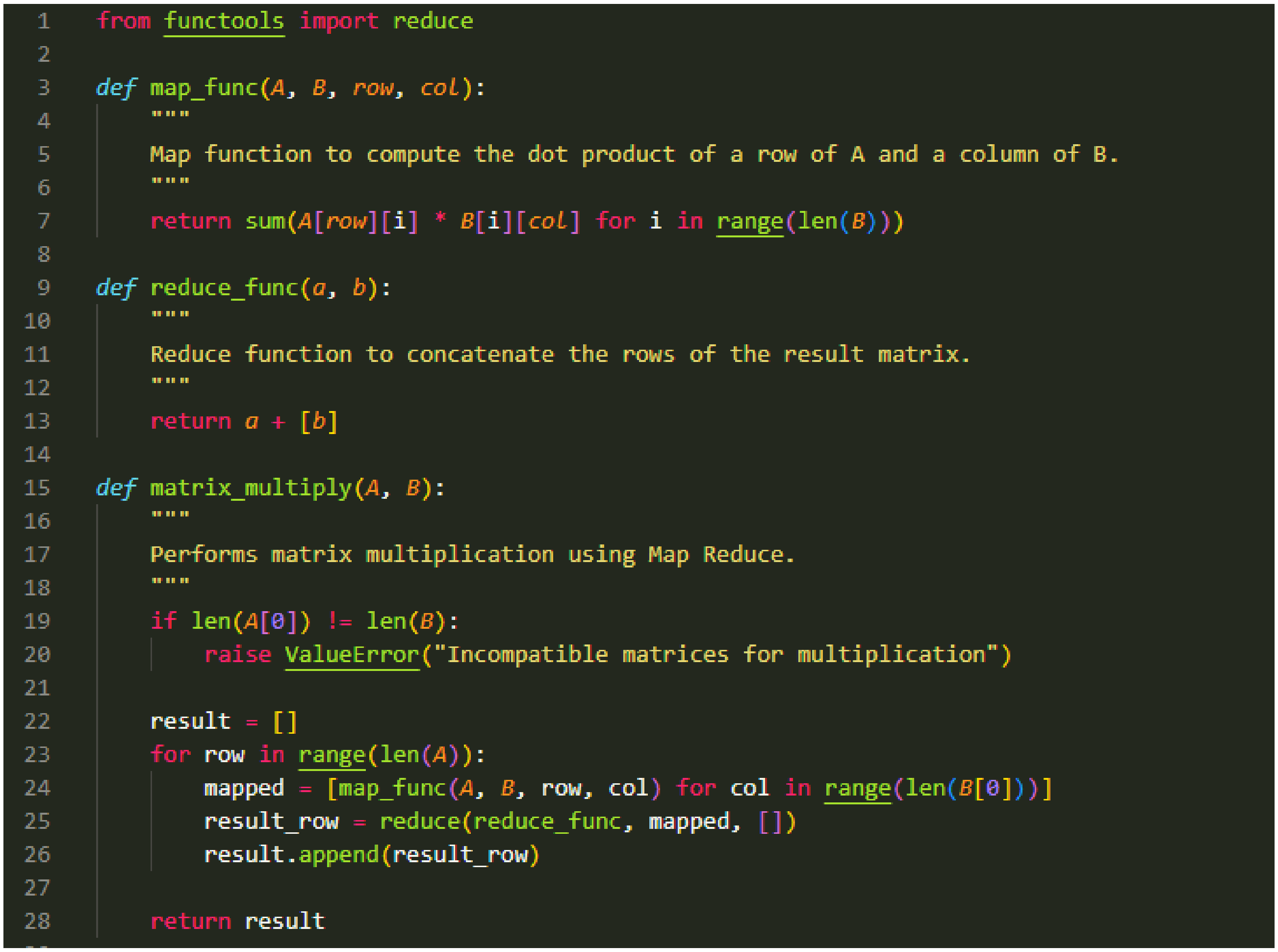
```python

    def reducer(self, word, counts):
        # Sum up the counts for each word
        yield word, sum(counts)

if __name__ == '__main__':
    WordFrequency.run()
```

Output:

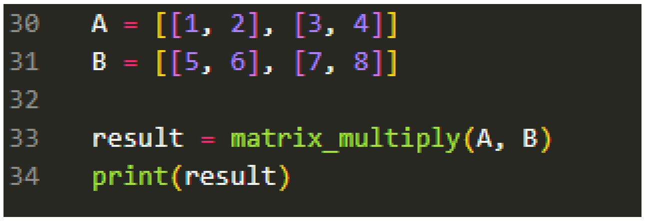


```
C:\Users\Admin\Desktop\semviii_lab\CL4\06>py frequency.py syp.pdf --word "data"
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\Admin\AppData\Local\Temp\frequency.Admin.20240417.101149.810978
Running step 1 of 1...
job output is in C:\Users\Admin\AppData\Local\Temp\frequency.Admin.20240417.101149.810978\output
Streaming final output from C:\Users\Admin\AppData\Local\Temp\frequency.Admin.20240417.101149.810978\output...
"data"  14238
Removing temp directory C:\Users\Admin\AppData\Local\Temp\frequency.Admin.20240417.101149.810978...
```

Assignment 08 (code):

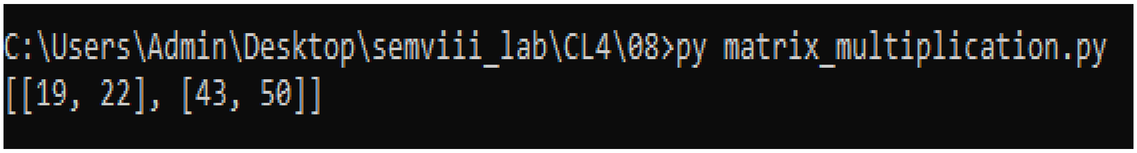


```python
from functools import reduce

def map_func(A, B, row, col):
    """
    Map function to compute the dot product of a row of A and a column of B.
    """
    return sum(A[row][i] * B[i][col] for i in range(len(B)))

def reduce_func(a, b):
    """
    Reduce function to concatenate the rows of the result matrix.
    """
    return a + [b]

def matrix_multiply(A, B):
    """
    Performs matrix multiplication using Map Reduce.
    """
    if len(A[0]) != len(B):
        raise ValueError("Incompatible matrices for multiplication")

    result = []
    for row in range(len(A)):
        mapped = [map_func(A, B, row, col) for col in range(len(B[0]))]
        result_row = reduce(reduce_func, mapped, [])
        result.append(result_row)

    return result
```



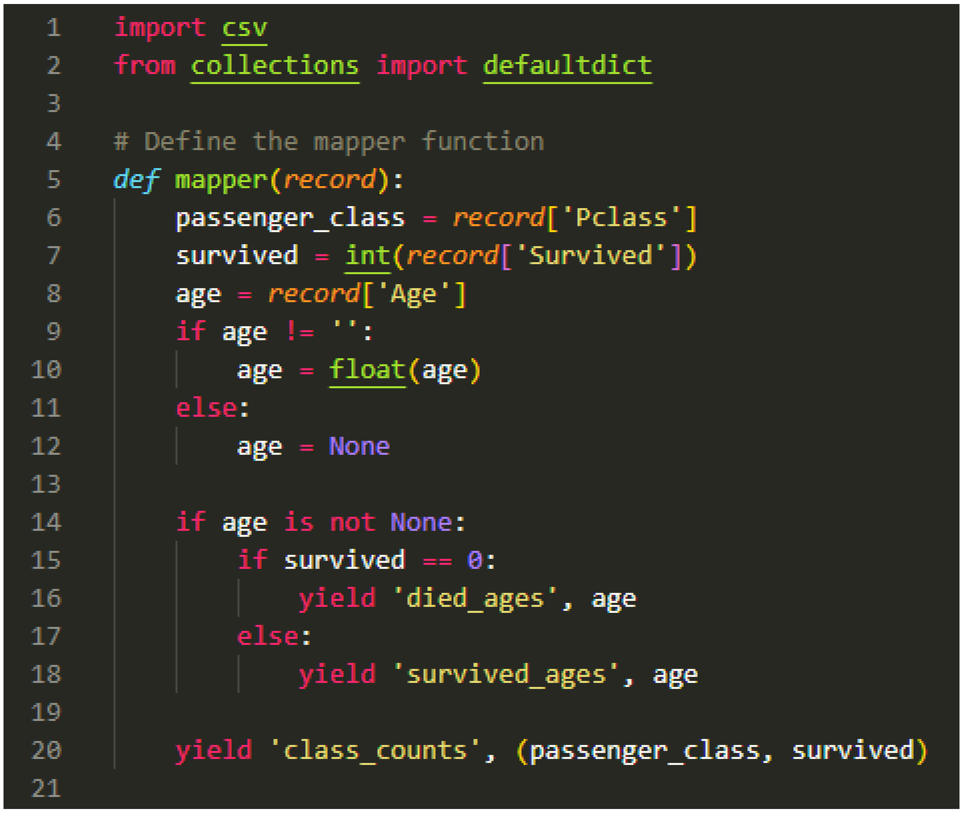
```python
A = [[1, 2], [3, 4]]
B = [[5, 6], [7, 8]]

result = matrix_multiply(A, B)
print(result)
```

Output:

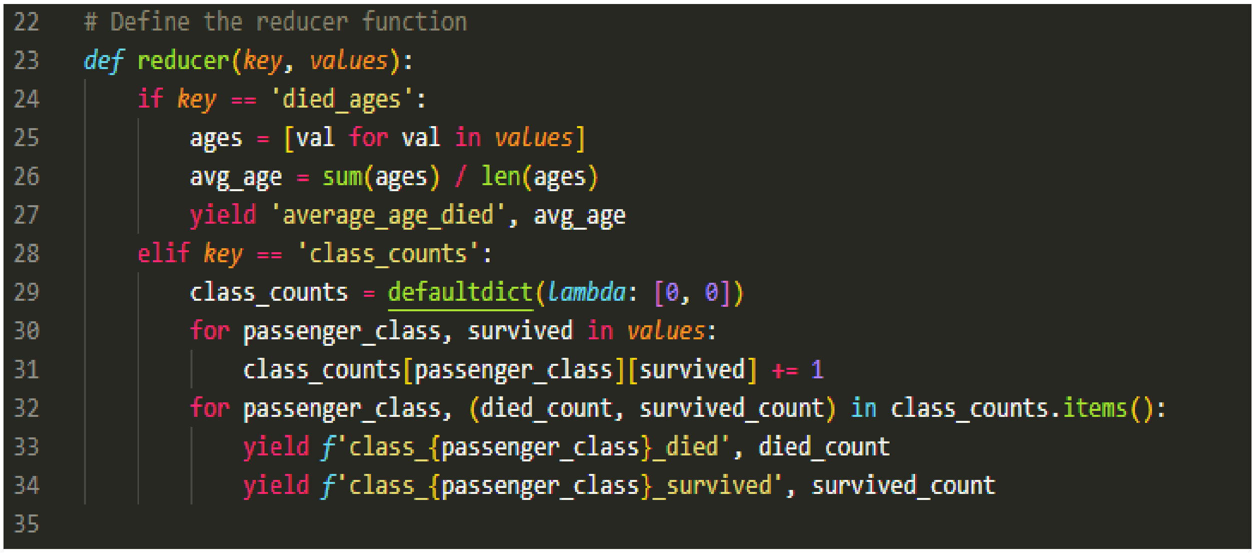


```
C:\Users\Admin\Desktop\semviii_lab\CL4\08>py matrix_multiplication.py
[[19, 22], [43, 50]]
```

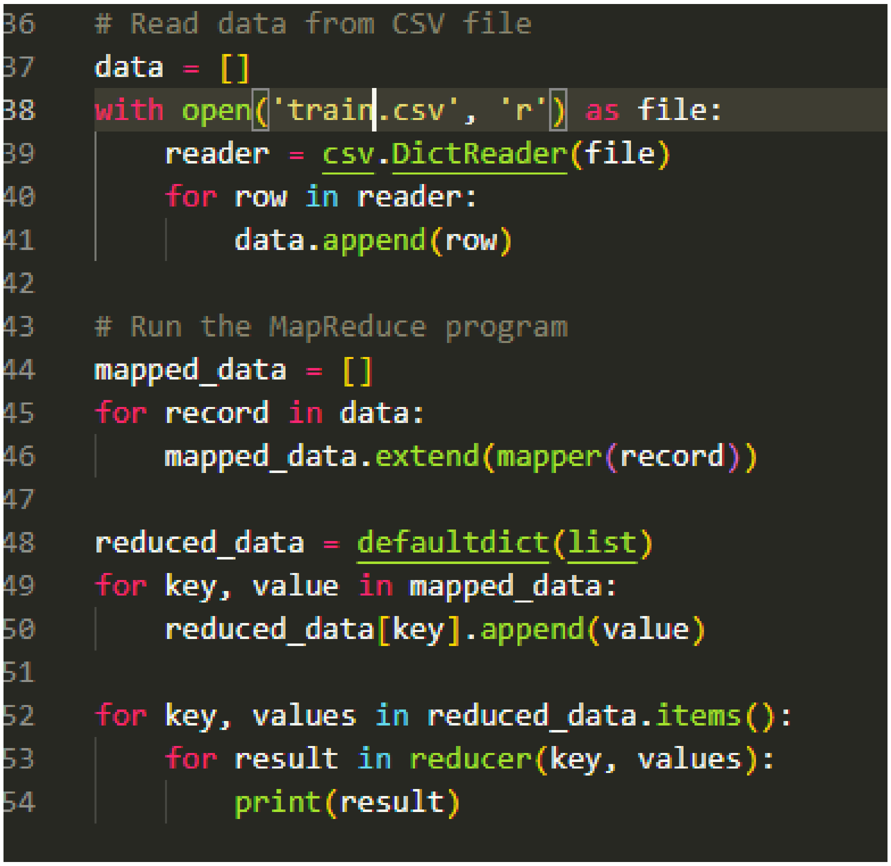
ASSIGNMENT 09 (code):



```python
import csv
from collections import defaultdict

# Define the mapper function
def mapper(record):
    passenger_class = record['Pclass']
    survived = int(record['Survived'])
    age = record['Age']
    if age != '':
        age = float(age)
    else:
        age = None

    if age is not None:
        if survived == 0:
            yield 'died_ages', age
        else:
            yield 'survived_ages', age

    yield 'class_counts', (passenger_class, survived)

```

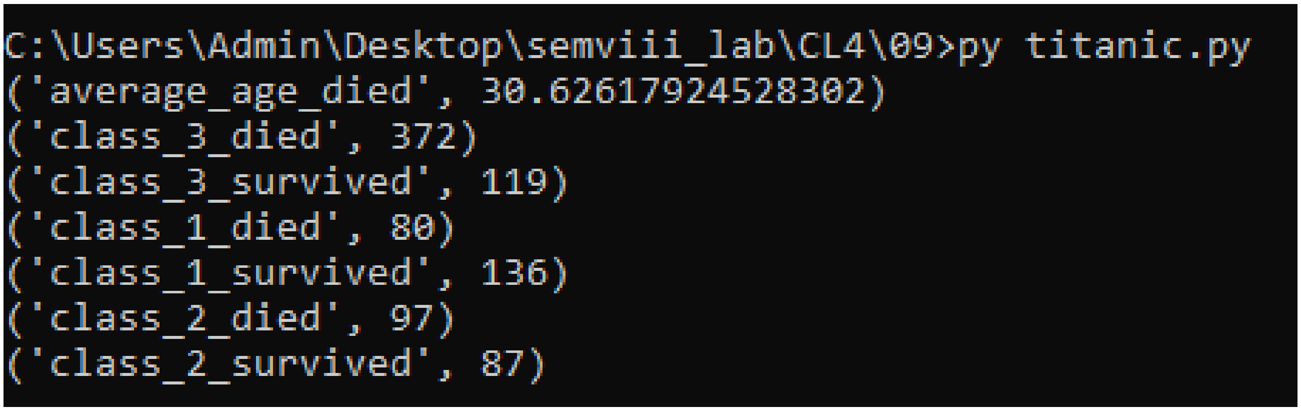


```python
# Define the reducer function
def reducer(key, values):
    if key == 'died_ages':
        ages = [val for val in values]
        avg_age = sum(ages) / len(ages)
        yield 'average_age_died', avg_age
    elif key == 'class_counts':
        class_counts = defaultdict(lambda: [0, 0])
        for passenger_class, survived in values:
            class_counts[passenger_class][survived] += 1
        for passenger_class, (died_count, survived_count) in class_counts.items():
            yield f'class_{passenger_class}_died', died_count
            yield f'class_{passenger_class}_survived', survived_count

```



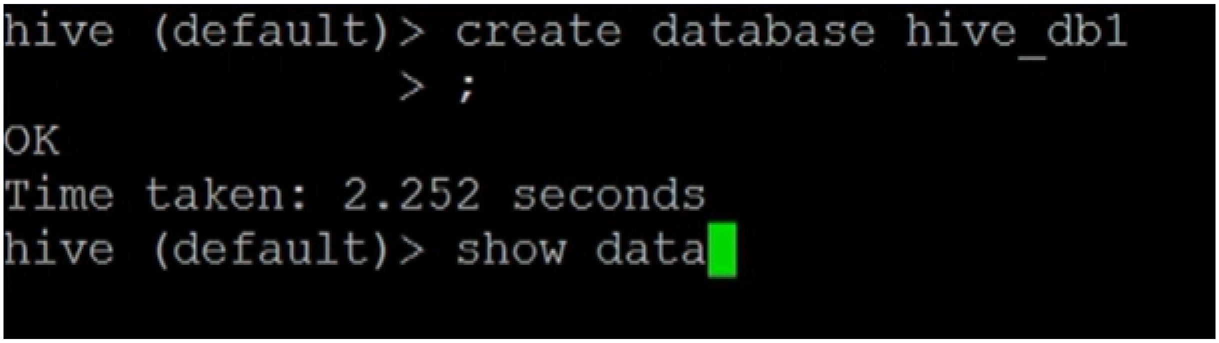
```
# Read data from CSV file
data = []
with open('train.csv', 'r') as file:
    reader = csv.DictReader(file)
    for row in reader:
        data.append(row)

# Run the MapReduce program
mapped_data = []
for record in data:
    mapped_data.extend(mapper(record))

reduced_data = defaultdict(list)
for key, value in mapped_data:
    reduced_data[key].append(value)

for key, values in reduced_data.items():
    for result in reducer(key, values):
        print(result)
```

Output:

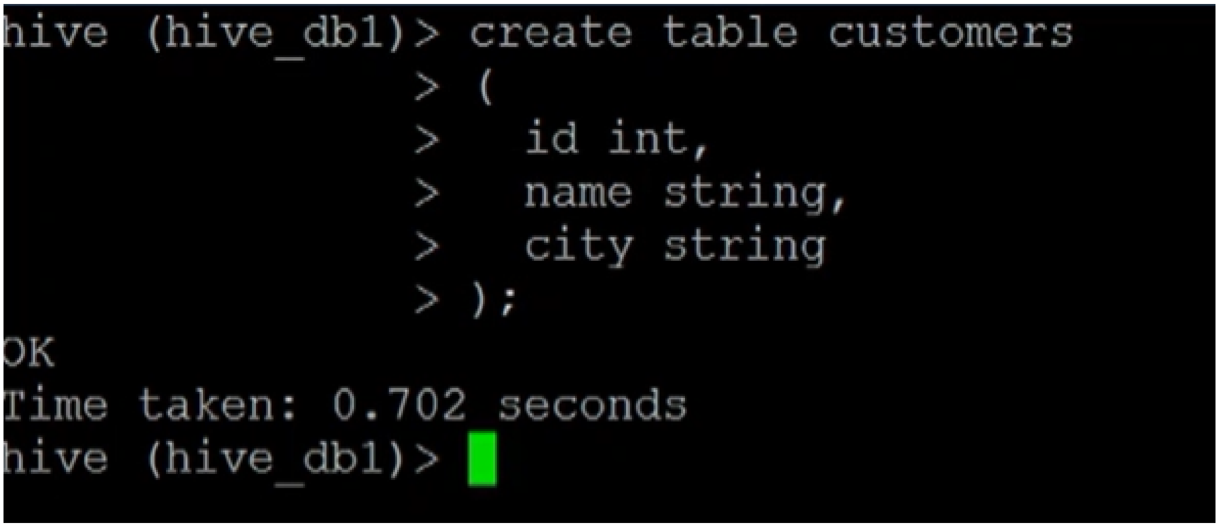


```
C:\Users\Admin\Desktop\semviii_lab\CL4\09>py titanic.py
('average_age_died', 30.62617924528302)
('class_3_died', 372)
('class_3_survived', 119)
('class_1_died', 80)
('class_1_survived', 136)
('class_2_died', 97)
('class_2_survived', 87)
```

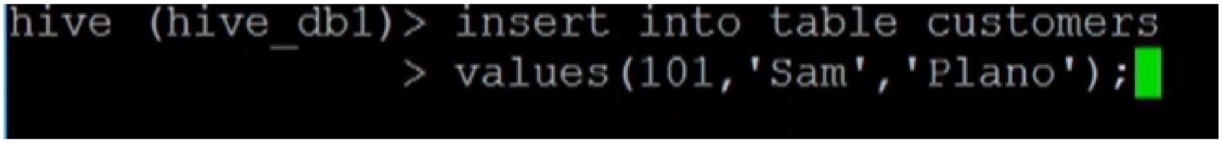
ASSIGNMENT 10:



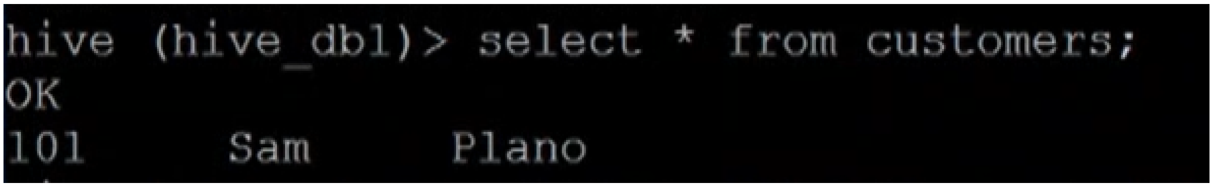
```
hive (default)> create database hive_db1
              > ;
OK
Time taken: 2.252 seconds
hive (default)> show data
```



```
hive (hive_db1)> create table customers
               > (
               >   id int,
               >   name string,
               >   city string
               > );
OK
Time taken: 0.702 seconds
hive (hive_db1)>
```



```
hive (hive_db1)> insert into table customers
               > values(101,'Sam','Plano');
```



```
hive (hive_db1)> select * from customers;
OK
101     Sam     Plano
```