Loknete Dr.Balasaheb Vikhe Patil (Padma Bhushan Awardee)
Pravara Rural Education Society’s

SIR VISVEVSRAYA INSTITUTE OF TECHNOLOGY, NASHIK

DEPARTMENT OF COMPUTER ENGINEERING

Lab Manual

Final Year Computer Engineering
Semester-VIII

PART-II 410253 : Elective VI :Business Intelligence
Subject Code: 410253(C)

Prepared By:
Mr. Pravin M. Tambe
(Assistant Professor, ME Computer Engineering)

Academic year 2022-23

Guidelines for Instructor's Manual:

List of recommended programming assignments and sample mini-projects is provided for reference.
Referring to these, Course Teacher or Lab Instructor may frame the assignments/mini-project by understanding the prerequisites, technological aspects, utility and recent trends related to the respective courses. Preferably there should be multiple sets of assignments/mini-project and distributed among batches of students. Real world problems/application based assignments/mini-projects create interest among learners serving as foundation for future research or startup of business projects. Mini-project can be completed in group of 2 to 3 students. Software Engineering approach with proper documentation is to be strictly followed. Use of open source software is to be encouraged. Instructor may also set one assignment or mini-project that is suitable to the respective course beyond the scope of syllabus.

Operating System recommended: - 64-bit Open source Linux or its derivative
Programming Languages: C++/JAVA/PYTHON/R
Programming tools recommended: Front End: Java/Perl/PHP/Python/Ruby/.net,
Backend: MongoDB/MYSQL/Oracle, Database Connectivity: ODBC/JDBC,
Additional Tools: Octave, Matlab, WEKA,powerBI

Guidelines for Laboratory /Term Work Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab Home Faculty of Engineering Savitribai Phule Pune University Syllabus for Fourth Year of Computer Engineering assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage.
Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness reserving weightage for successful mini-project completion and related documentation.

Sr. No.	Laboratory Assignments
1	Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)
2	Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver.
3	Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.
4	Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart
5	Perform the data classification using classification algorithm. Or Perform the data clustering using clustering algorithm.
6	Business Intelligence Mini Project: Each group of 4 Students (max) assigned one case study for this; A BI report must be prepared outlining the following steps: a) Problem definition, identifying which data mining task is needed. b) Identify and use a standard data mining dataset available for the problem.

Assessment:

Laboratory Assignments:

- The external continuous assessment will be 50 marks.
- The outline of distribution assignments Marks for various aspects /mechanisms towards Continuous Assessment is as follows:

Sr. No.	Distribution of Marks	Marks
1	Term Work	50
Total		**50**

Group : 1
Assignment No : 1

Title of the Assignment :

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 of the Assignment :

To introduce the concepts and components of Business Intelligence (BI)

Prerequisite:

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

Theory :

Legacy Data :

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

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.

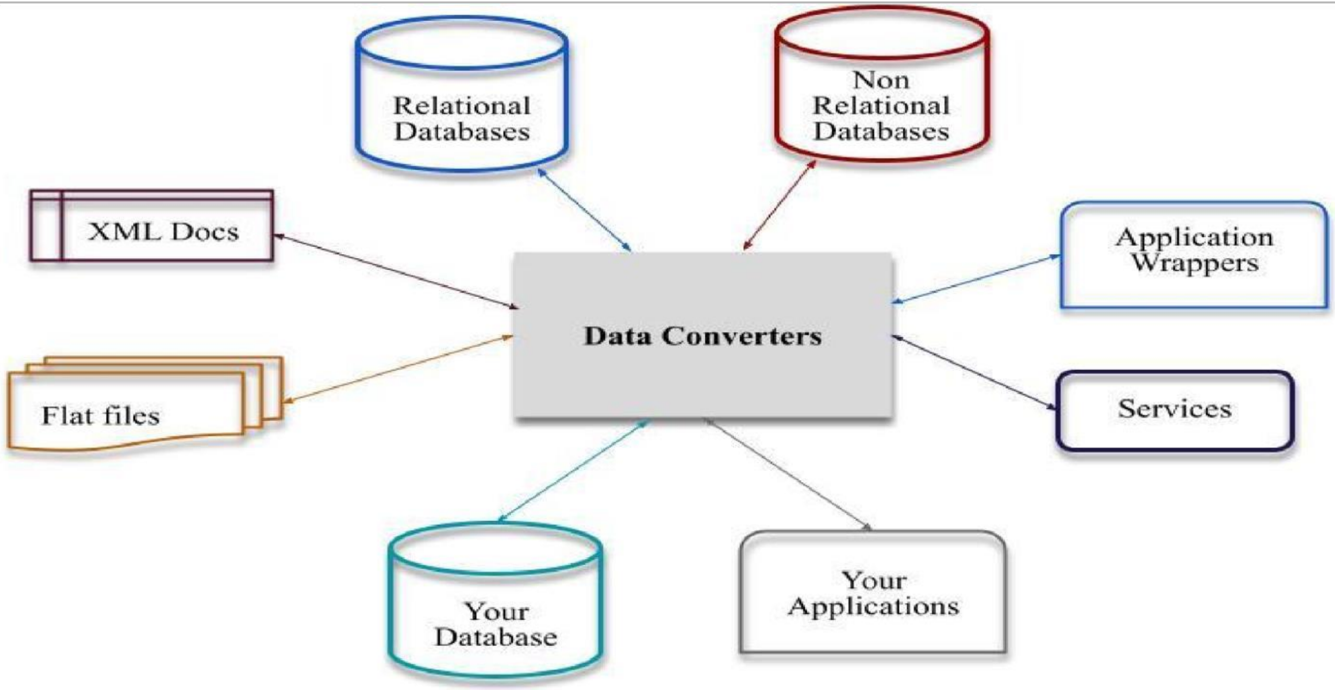
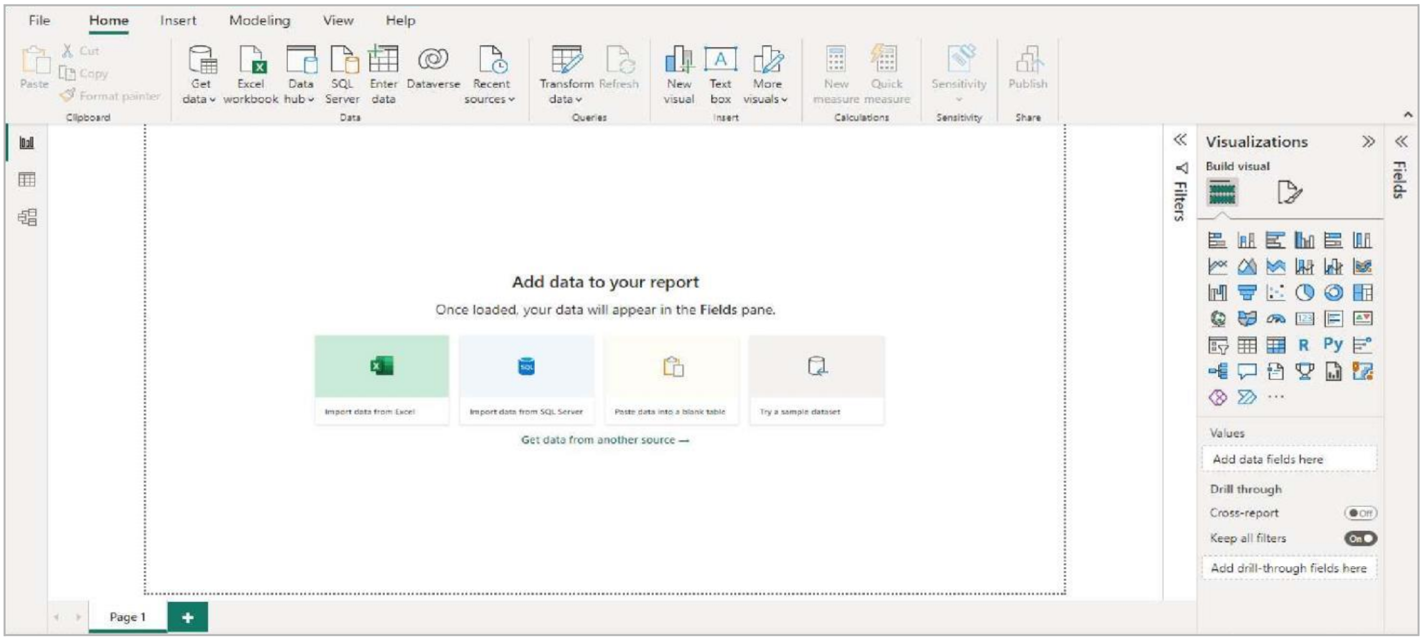


Figure 1. Legacy data sources.

How to import legacy data step by step.

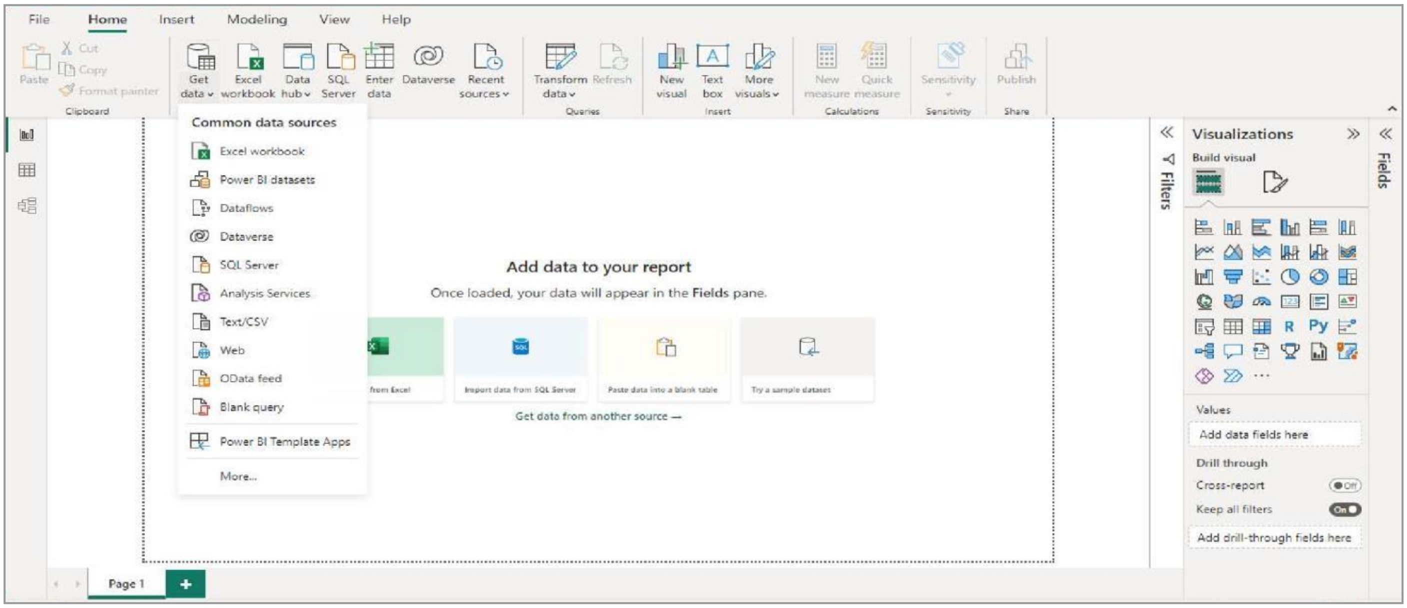
Step 1: Open Power BI



Add data to your report

Once loaded, your data will appear in the **Fields** pane.

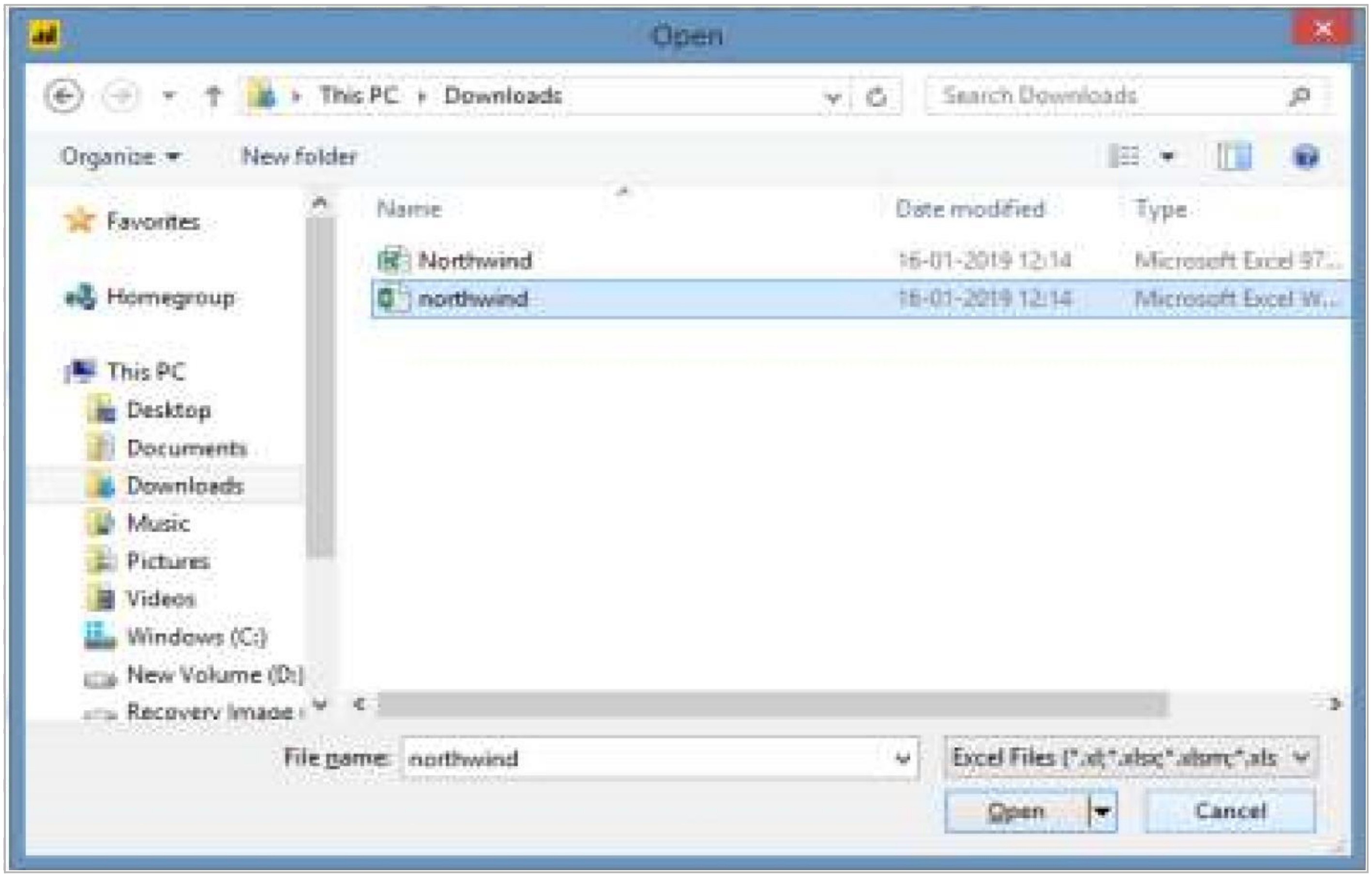
Step 2 : Click on Get data following list will be displayed → select Excel



Add data to your report

Once loaded, your data will appear in the **Fields** pane.

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

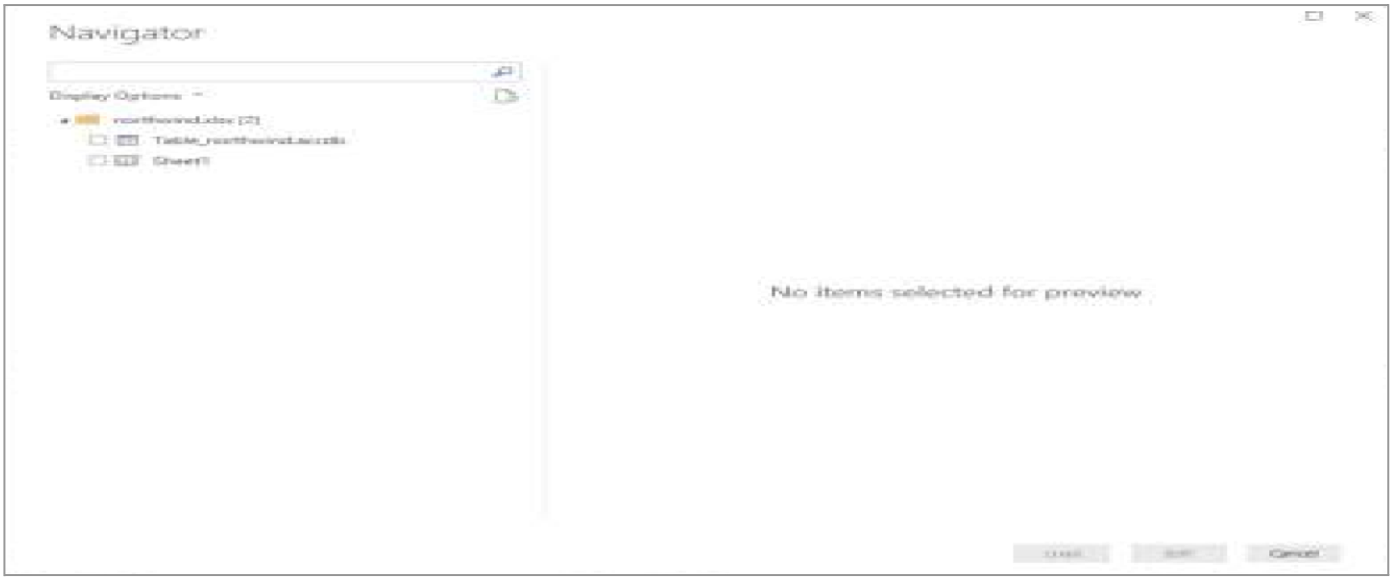


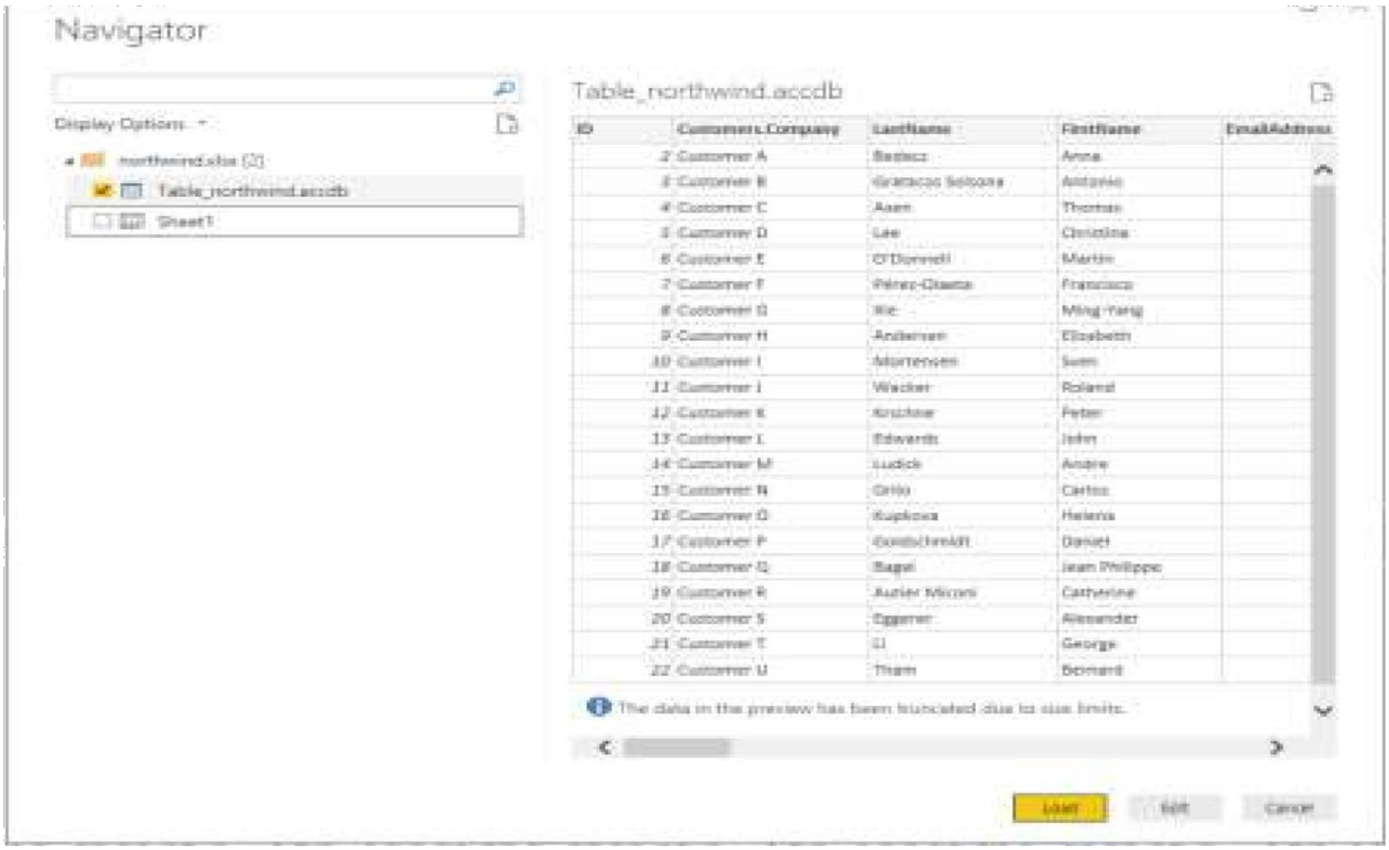
Name	Date modified	Type
Northwind	16-01-2019 12:14	Microsoft Excel 97...
northwind	16-01-2019 12:14	Microsoft Excel W...

File name: northwind

Excel Files (*.xl;*.xlsx;*.xlsm;*.xls

Step 4: Select file and click on edit



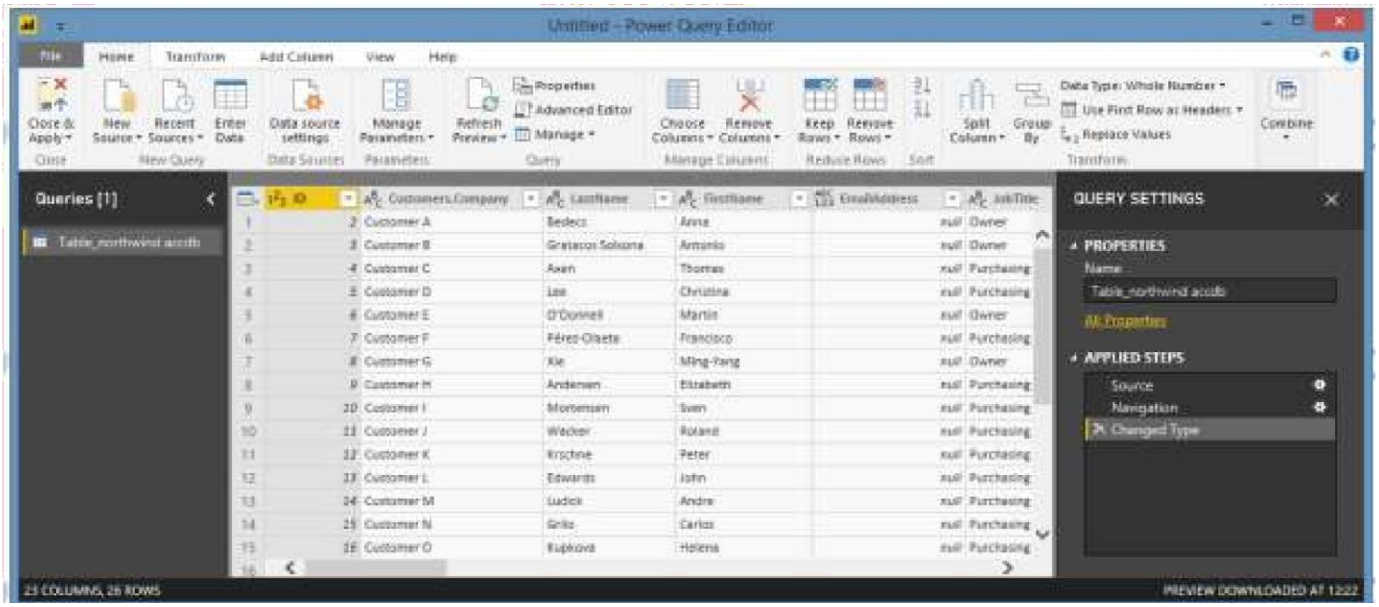


Table_northwind.accdb

ID	Customers.Company	LastName	FirstName	EmailAddress
2	Customer A	Bedecs	Anna	
3	Customer B	Gratacos Solsona	Antonio	
4	Customer C	Axen	Thomas	
5	Customer D	Lee	Christina	
6	Customer E	O'Donnell	Martin	
7	Customer F	Pérez-Olaeta	Francisco	
8	Customer G	Xie	Ming-Yang	
9	Customer H	Andersen	Elizabeth	
10	Customer I	Mortensen	Sven	
11	Customer J	Wacker	Roland	
12	Customer K	Krschne	Peter	
13	Customer L	Edwards	John	
14	Customer M	Ludick	Andre	
15	Customer N	Grilo	Carlos	
16	Customer O	Kupkova	Helena	
17	Customer P	Goldschmidt	Daniel	
18	Customer Q	Bagel	Jean Philippe	
19	Customer R	Autier Miconi	Catherine	
20	Customer S	Eggerer	Alexander	
21	Customer T	Li	George	
22	Customer U	Tham	Bernard	

The data in the preview has been truncated due to size limits.

Step 5: Power query editor appears



Name: Table_northwind accdb

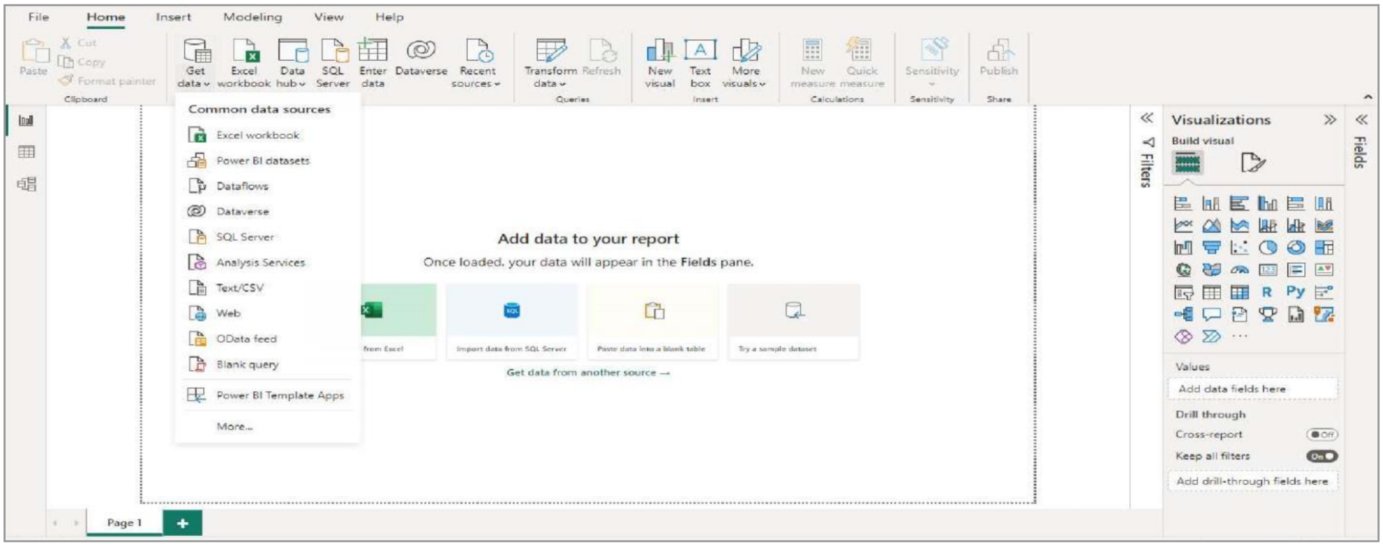
Applied steps: Source, Navigation, Changed Type

	ID	Customers.Company	LastName	FirstName	EmailAddress	JobTitle
1	2	Customer A	Bedecs	Anna	null	Owner
2	3	Customer B	Gratacos Solsona	Antonio	null	Owner
3	4	Customer C	Axen	Thomas	null	Purchasing
4	5	Customer D	Lee	Christina	null	Purchasing
5	6	Customer E	O'Donnell	Martin	null	Owner
6	7	Customer F	Pérez-Olaeta	Francisco	null	Purchasing
7	8	Customer G	Xie	Ming-Yang	null	Owner
8	9	Customer H	Andersen	Elizabeth	null	Purchasing
9	10	Customer I	Mortensen	Sven	null	Purchasing
10	11	Customer J	Wacker	Roland	null	Purchasing
11	12	Customer K	Krschne	Peter	null	Purchasing
12	13	Customer L	Edwards	John	null	Purchasing
13	14	Customer M	Ludick	Andre	null	Purchasing
14	15	Customer N	Grilo	Carlos	null	Purchasing
15	16	Customer O	Kupkova	Helena	null	Purchasing

23 COLUMNS, 26 ROWS

PREVIEW DOWNLOADED AT 12:22

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

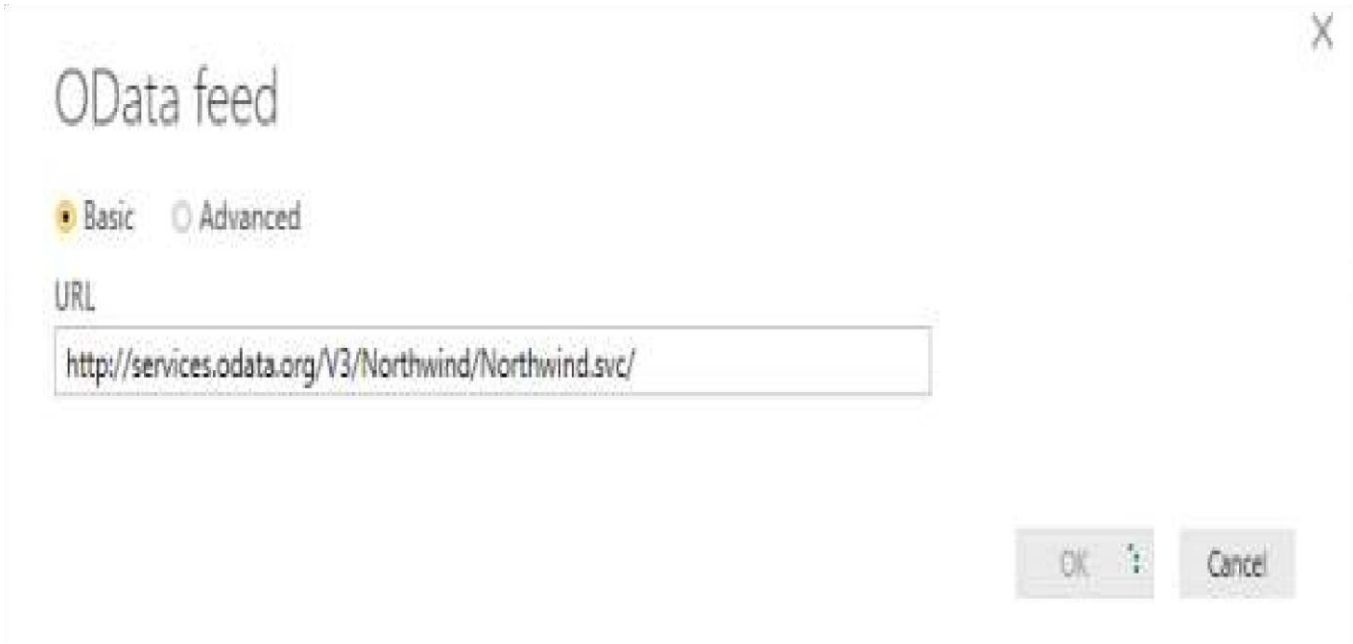


Add data to your report

Once loaded, your data will appear in the **Fields** pane.

Step 7:

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



OData feed

● Basic ○ Advanced

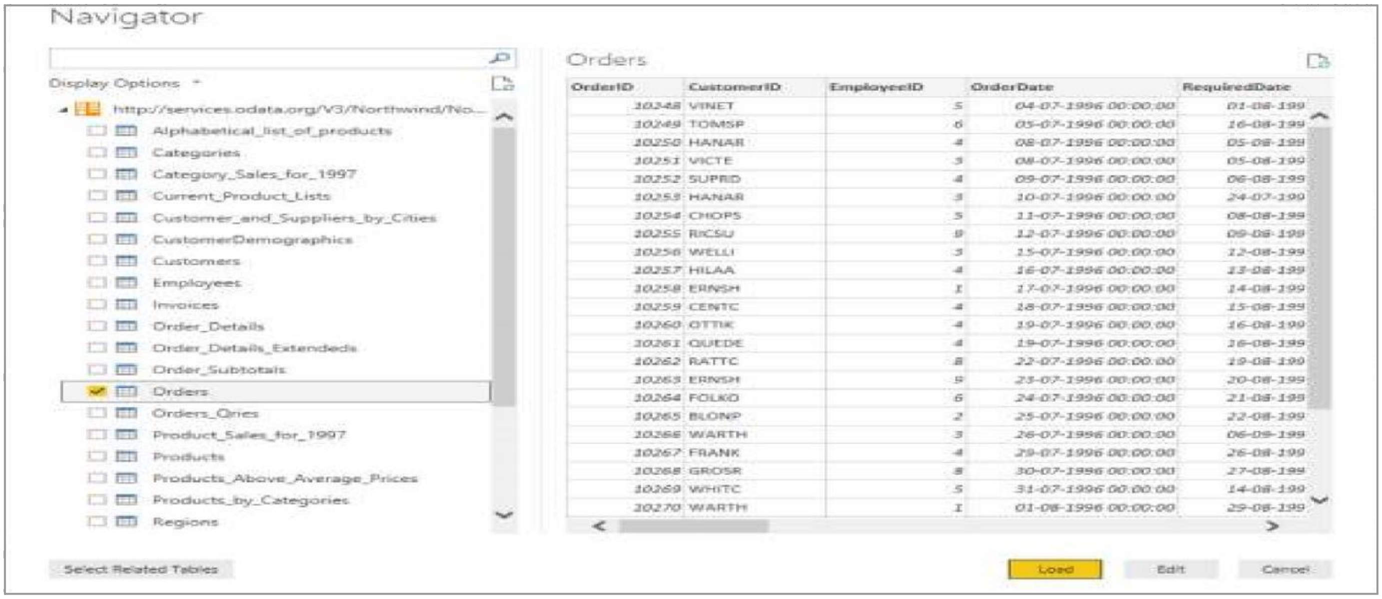
URL

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

Step 8: Select orders table

And click on edit

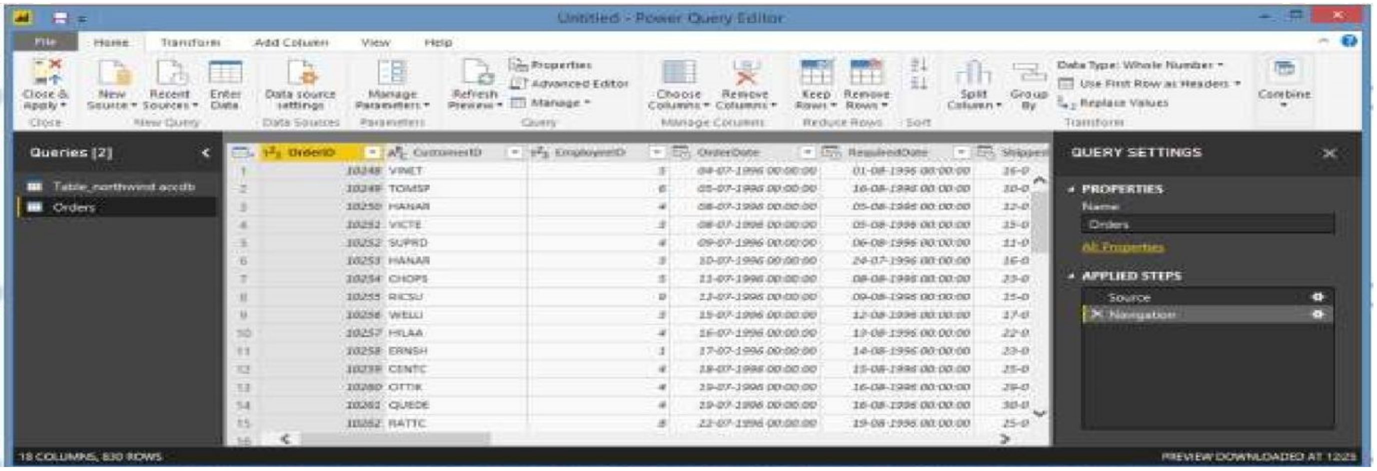
Click on edit to view table



☑ Orders

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	04-07-1996 00:00:00	01-08-199
10249	TOMSP	6	05-07-1996 00:00:00	16-08-199
10250	HANAR	4	08-07-1996 00:00:00	05-08-199
10251	VICTE	3	08-07-1996 00:00:00	05-08-199
10252	SUPRD	4	09-07-1996 00:00:00	06-08-199
10253	HANAR	3	10-07-1996 00:00:00	24-07-199
10254	CHOPS	5	11-07-1996 00:00:00	08-08-199
10255	RICSU	9	12-07-1996 00:00:00	09-08-199
10256	WELLI	3	15-07-1996 00:00:00	12-08-199
10257	HILAA	4	16-07-1996 00:00:00	13-08-199
10258	ERNSH	1	17-07-1996 00:00:00	14-08-199
10259	CENTC	4	18-07-1996 00:00:00	15-08-199
10260	OTTIK	4	19-07-1996 00:00:00	16-08-199
10261	QUEDE	4	19-07-1996 00:00:00	16-08-199
10262	RATTC	8	22-07-1996 00:00:00	19-08-199
10263	ERNSH	9	23-07-1996 00:00:00	20-08-199
10264	FOLKO	6	24-07-1996 00:00:00	21-08-199
10265	BLONP	2	25-07-1996 00:00:00	22-08-199
10266	WARTH	3	26-07-1996 00:00:00	06-09-199
10267	FRANK	4	29-07-1996 00:00:00	26-08-199
10268	GROSR	8	30-07-1996 00:00:00	27-08-199
10269	WHITC	5	31-07-1996 00:00:00	14-08-199
10270	WARTH	1	01-08-1996 00:00:00	29-08-199



Name: Orders

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipped
1	10248	VINET	5	04-07-1996 00:00:00	01-08-1996 00:00:00	16-0
2	10249	TOMSP	6	05-07-1996 00:00:00	16-08-1996 00:00:00	10-0
3	10250	HANAR	4	08-07-1996 00:00:00	05-08-1996 00:00:00	12-0
4	10251	VICTE	3	08-07-1996 00:00:00	05-08-1996 00:00:00	15-0
5	10252	SUPRD	4	09-07-1996 00:00:00	06-08-1996 00:00:00	11-0
6	10253	HANAR	3	10-07-1996 00:00:00	24-07-1996 00:00:00	16-0
7	10254	CHOPS	5	11-07-1996 00:00:00	08-08-1996 00:00:00	23-0
8	10255	RICSU	9	12-07-1996 00:00:00	09-08-1996 00:00:00	15-0
9	10256	WELLI	3	15-07-1996 00:00:00	12-08-1996 00:00:00	17-0
10	10257	HILAA	4	16-07-1996 00:00:00	13-08-1996 00:00:00	22-0
11	10258	ERNSH	1	17-07-1996 00:00:00	14-08-1996 00:00:00	23-0
12	10259	CENTC	4	18-07-1996 00:00:00	15-08-1996 00:00:00	25-0
13	10260	OTTIK	4	19-07-1996 00:00:00	16-08-1996 00:00:00	29-0
14	10261	QUEDE	4	19-07-1996 00:00:00	16-08-1996 00:00:00	30-0
15	10262	RATTC	8	22-07-1996 00:00:00	19-08-1996 00:00:00	25-0
16						

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

Group No: 1
Assignment No: 2

Title of the Assignment :-

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

Objective of the Assignment :

To introduce the concepts and components of Business Intelligence (BI)

Prerequisite:

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

Theory:

Extraction Transformation and Loading (ETL) :-

ETL, which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.

As the databases grew in popularity in the 1970s, ETL was introduced as a process for integrating and loading data for computation and analysis, eventually becoming the primary method to process data for data warehousing projects.

ETL provides the foundation for data analytics and machine learning workstreams. Through a series of business rules, ETL cleanses and organizes data in a way which addresses specific business intelligence needs, like monthly reporting, but it can also tackle more advanced analytics, which can improve back-end processes or end user experiences. ETL is often used by an organization to:

- Extract data from legacy systems
- Cleanse the data to improve data quality and establish consistency
- Load data into a target database

➢ How ETL works

The easiest way to understand how ETL works is to understand what happens in each step of the process.

Extract

During data extraction, raw data is copied or exported from source locations to a staging area. Data management teams can extract data from a variety of data sources, which can be structured or unstructured. Those sources include but are not limited to:

- SQL or NoSQL servers
- CRM and ERP systems
- Flat files
- Email
- Web pages

➢ The benefits and challenges of ETL

ETL solutions improve quality by performing data cleansing prior to loading the data to a different repository. A time-consuming batch operation, ETL is recommended more often for creating smaller target data repositories that require less frequent updating, while other data integration methods—including ELT (extract, load, transform), change data capture (CDC), and data virtualization—are used to integrate increasingly larger volumes of data that changes or real-time data streams.

➢ What Is ETL Process?

To put it simply, the data ETL process including *extracting* and compiling raw data, *transforming* it to make it intelligible, and *loading* it into a target system, such as a database or data warehouse, for easy access and analysis. ETL short for Extract, Transform, Load, is an important component in the data ecosystem of any modern businesses. The ETL process is what helps break down data silos and makes data easier to access for decision-makers.

Since data coming from multiple sources has a different schema, every dataset must be transformed differently before utilizing BI and analytics. For instance, if you are compiling data from source systems like SQL Server and Google Analytics, these two sources will need to be treated individually throughout the ETL process. The importance of this process has increased since big data analysis has become a necessary part of every organization.

➢ ETL tools :-

In the past, organizations wrote their own ETL code. There are now many open source and commercial ETL tools and cloud services to choose from. Typical capabilities of these products include the following:

- **Comprehensive automation and ease of use:** Leading ETL tools automate the entire data flow, from data sources to the target data warehouse. Many tools recommend rules for extracting, transforming and loading the data.
- **A visual, drag-and-drop interface:** This functionality can be used for specifying rules and data flows.
- **Support for complex data management:** This includes assistance with complex calculations, data integrations, and string manipulations
- **Security and compliance:** The best ETL tools encrypt data both in motion and at rest and are certified compliant with industry or government regulations, like HIPAA and GDPR.

Conclusion :-

In this way we import the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server.

Group No: 1
Assignment No: 3

Title of the Assignment :-

Create the cube with suitable dimension and fact tables based on OLAP, MOLAP and HOLAP model

Objective of the Assignment :

To introduce the concepts and components of OLAP, MOLAP and HOLAP model.

Prerequisite:

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

Theory:

What is OLAP?

OLAP was introduced into the business intelligence (BI) space over 20 years ago, in a time where computer hardware and software technology weren’t nearly as powerful as they are today. OLAP introduced a (typically analysts) to easily perform multidimensional analysis of large volumes of business data.

Aggregating, grouping, and joining data are the most difficult types of queries for a relational database to process. The magic behind OLAP derives from its ability to pre-calculate and pre-aggregate data. Otherwise, end users would be spending most of their time waiting for query results to be returned by the database. However, it is also what causes OLAP-based solutions to be extremely rigid and IT-intensive.

Limitations of OLAP cubes

- OLAP requires restructuring of data into a star/snowflake schema
- There is a limited number of dimensions (fields) a single OLAP cube
- It is nearly impossible to access transactional data in the OLAP cube
- Changes to an OLAP cube requires a full update of the cube – a lengthy process

OLAP stands for Relational Online Analytical Processing. ROLAP stores data in columns and rows (also known as relational tables) and retrieves the information on demand through user submitted queries. A ROLAP database can be accessed through complex SQL queries to calculate information. ROLAP can handle large data volumes, but the larger the data, the slower the processing times.

Because queries are made on-demand, ROLAP does not require the storage and pre-computation of information. However, the disadvantage of ROLAP implementations are the potential performance constraints and scalability limitations that result from large and inefficient join operations between large tables. Examples of popular ROLAP products include Metacube by Stanford Technology Group, Red Brick Warehouse by Red Brick Systems, and AXSYS Suite by Information Advantage.

What is MOLAP?

MOLAP stands for Multidimensional Online An

MOLAP stands for Multidimensional Online Analytical Processing. MOLAP uses a multidimensional cube that accesses stored data through various combinations. Data is pre-computed, pre-summarized, and stored (a difference from ROLAP, where queries are served on-demand).

A multicube approach has proved successful in MOLAP products. In this approach, a series of dense, small, precalculated cubes make up a hypercube. Tools that incorporate MOLAP include Oracle Essbase, IBM Cognos, and Apache Kylin.

Its simple interface makes MOLAP easy to use, even for inexperienced users. Its speedy data retrieval makes it the best for “slicing and dicing” operations. One major disadvantage of MOLAP is that it is less scalable than ROLAP, as it can handle a limited amount of data.

What is HOLAP?

HOLAP stands for Hybrid Online Analytical Processing. As the name suggests, the HOLAP storage mode connects attributes of both MOLAP and ROLAP. Since HOLAP involves storing part of your data in a ROLAP store and another part in a MOLAP store, developers get the benefits of both.

With this use of the two OLAPs, the data is stored in both multidimensional databases and relational databases. The decision to access one of the databases depends on which is most appropriate for the requested processing application or type. This setup allows much more flexibility for handling data. For theoretical processing, the data is stored in a multidimensional database. For heavy processing, the data is stored in a relational database.

Microsoft Analysis Services and SAP AG BI Accelerator are products that run off HOLAP.

Sisense and Elasticubes

Similar to OLAP-based solutions, Sisense is a Business Intelligence software designed to enable solutions where multiple business users perform ad-hoc data analysis on a centralized data repository. On the other hand, Sisense does not achieve this by pre-calculating query results, but rather by utilizing state-of-the-art technology called ElastiCube. It is a sophisticated columnar database, which was specifically designed for Business Intelligence solutions. Its unique storage and memory processing technology radically change the way business intelligence solutions access data.

Powered by ElastiCube, Sisense delivers distinct advantages over OLAP-based solutions:

- Instant query response times, without pre-calculation or pre-aggregation of data
- Creation of complicated star/snow flake schemas is not required
- A data warehouse is not required, but easily supported

- There are no physical limits to the number of dimensions an ElastiCube can hold
- ElastiCube provides access to data in any granularity (not merely to aggregated data)
- Changes to ElastiCubes can be done without re-building the entire data model
- An ElastiCube requires significantly less powerful hardware than a similar OLAP cube.

Conclusion :-

In this way we import of the OLAP, MOLAP and HOLAP model.

Group No: 1
Assignment No: 4

Title of the Assignment :-

Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Char

Objective of the Assignment :

To introduce the concepts and components of data warehouse data in Microsoft Excel and create the Pivot table and Pivot Char.

Prerequisite:

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

Theory:

Data into Excel, and Create a Data Model

the first tutorial in a series designed to get you acquainted and comfortable using Excel and its built-in data mash-up and analysis features. These tutorials build and refine an Excel workbook from scratch, build a data model, then create amazing interactive reports using Power View. The tutorials are designed to demonstrate Microsoft Business Intelligence features and capabilities in Excel, PivotTables, Power Pivot, and Power View.

In these tutorials you learn how to import and explore data in Excel, build and refine a data model using Power Pivot, and create interactive reports with Power View that you can publish, protect, and share.

At the end of this tutorial is a quiz you can take to test your learning.

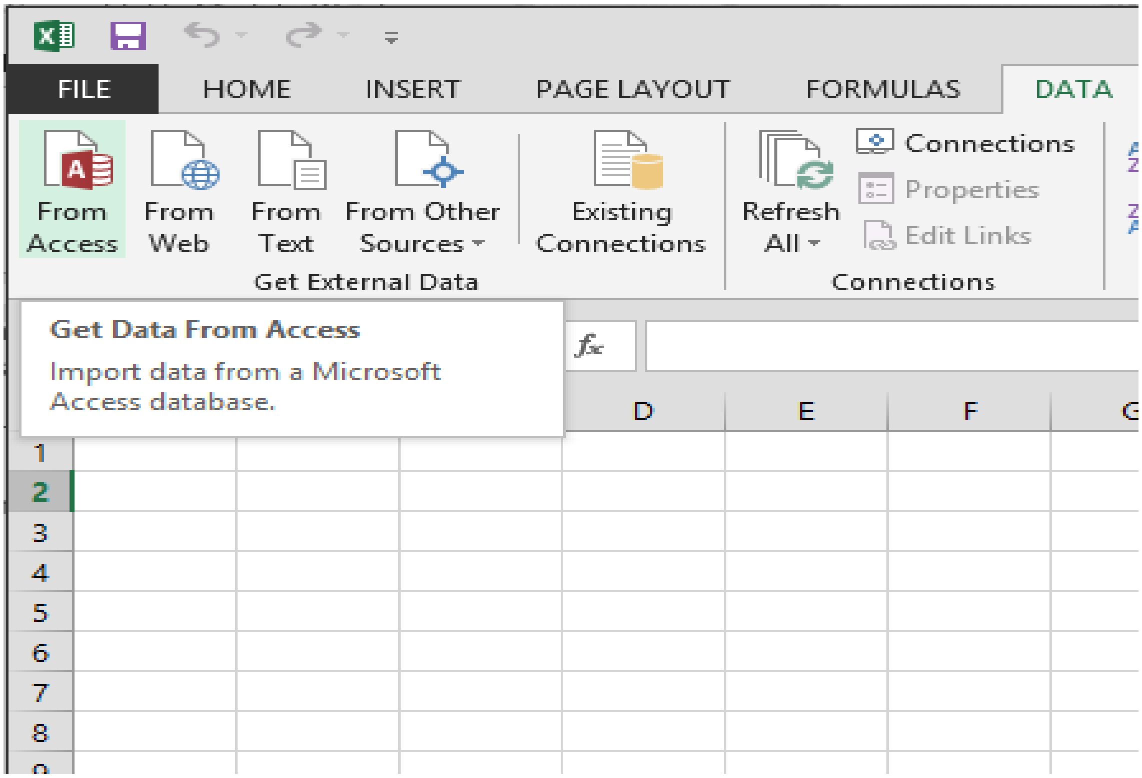
This tutorial series uses data describing Olympic Medals, hosting countries, and various Olympic sporting events. We suggest you go through each tutorial in order. Also, tutorials use Excel 2013 with Power Pivot enabled. For more information on Excel 2013, click here. For guidance on enabling Power Pivot, click here.

Import data from a database

We start this tutorial with a blank workbook. The goal in this section is to connect to an external data source, and import that data into Excel for further analysis.

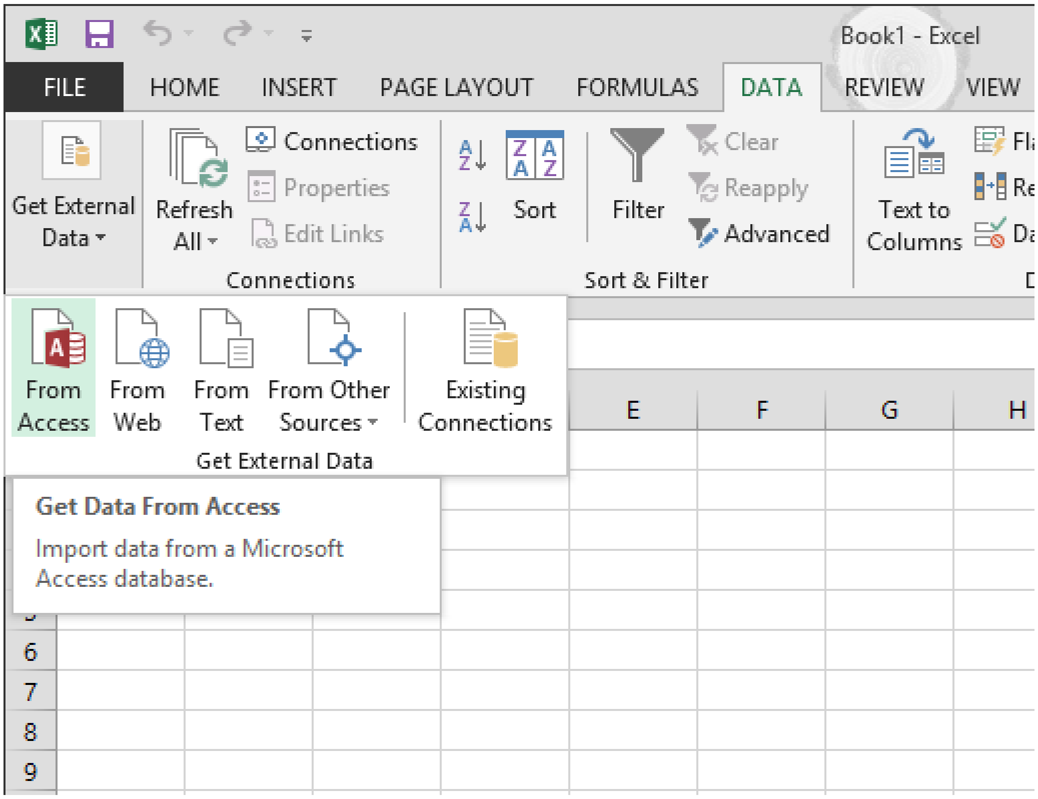
Let's start by downloading some data from the Internet. The data describes Olympic Medals, and is a Microsoft Access database.

1. Click the following links to download files we use during this tutorial series. Download each of the four files to a location that's easily accessible, such as *Downloads* or *My Documents*, or to a new folder you create:
2. In Excel 2013, open a blank workbook.
3. Click **DATA > Get External Data > From Access**. The ribbon adjusts dynamically based on the width of your workbook, so the commands on your ribbon may look slightly different from the following screens. The first screen shows the ribbon when a workbook is wide, the second image shows a workbook that has been resized to take up only a portion of the screen.



Get Data From Access

Import data from a Microsoft Access database.

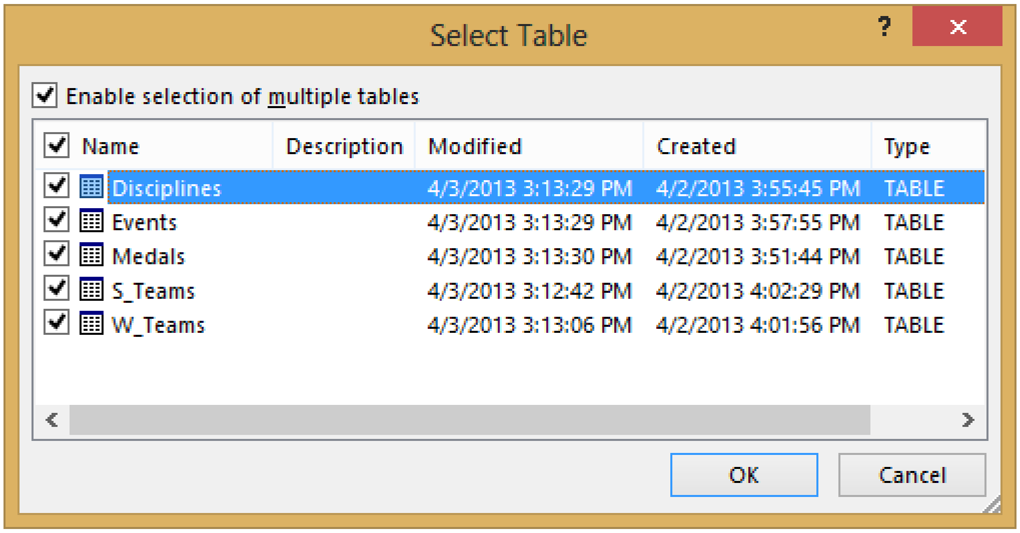


Get Data From Access

Import data from a Microsoft Access database.

4. Select the OlympicMedals.accdb file you downloaded and click **Open**. The following Select Table window appears, displaying the tables found in the database. Tables in a database are similar to worksheets or tables in Excel.

Check the **Enable selection of multiple tables** box, and select all the tables. Then click **OK**.



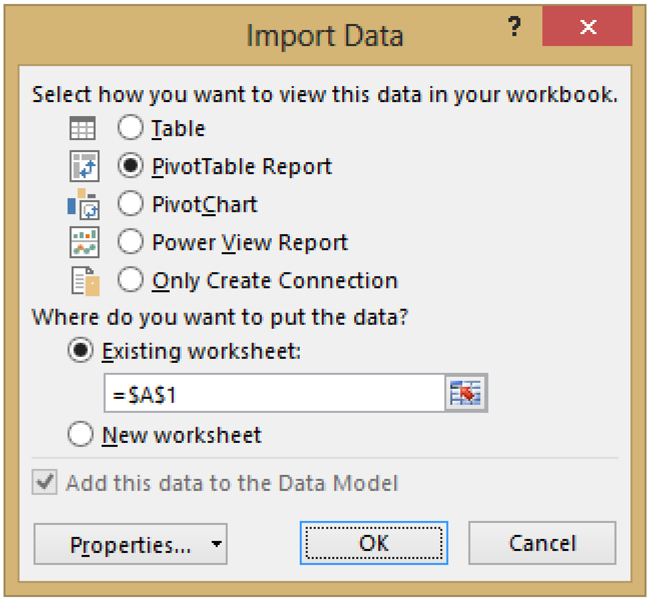
Select Table

☑ Enable selection of multiple tables

☑ Name	Description	Modified	Created	Type
☑ Disciplines		4/3/2013 3:13:29 PM	4/2/2013 3:55:45 PM	TABLE
☑ Events		4/3/2013 3:13:29 PM	4/2/2013 3:57:55 PM	TABLE
☑ Medals		4/3/2013 3:13:30 PM	4/2/2013 3:51:44 PM	TABLE
☑ S_Teams		4/3/2013 3:12:42 PM	4/2/2013 4:02:29 PM	TABLE
☑ W_Teams		4/3/2013 3:13:06 PM	4/2/2013 4:01:56 PM	TABLE

5. The Import Data window appears.

Select the **PivotTable Report** option, which imports the tables into Excel and prepares a PivotTable for analyzing the imported tables, and click **OK**.



Import Data

Select how you want to view this data in your workbook.

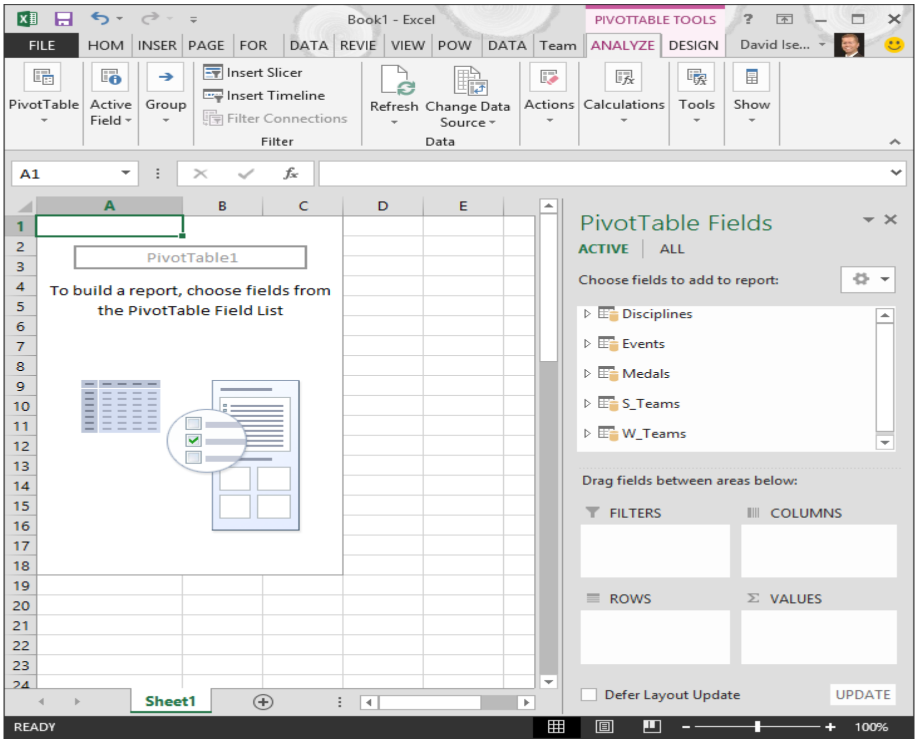
○ Table
● PivotTable Report
○ PivotChart
○ Power View Report
○ Only Create Connection

Where do you want to put the data?

● Existing worksheet:
=$A$1
○ New worksheet

☑ Add this data to the Data Model

6. Once the data is imported, a PivotTable is created using the imported tables.

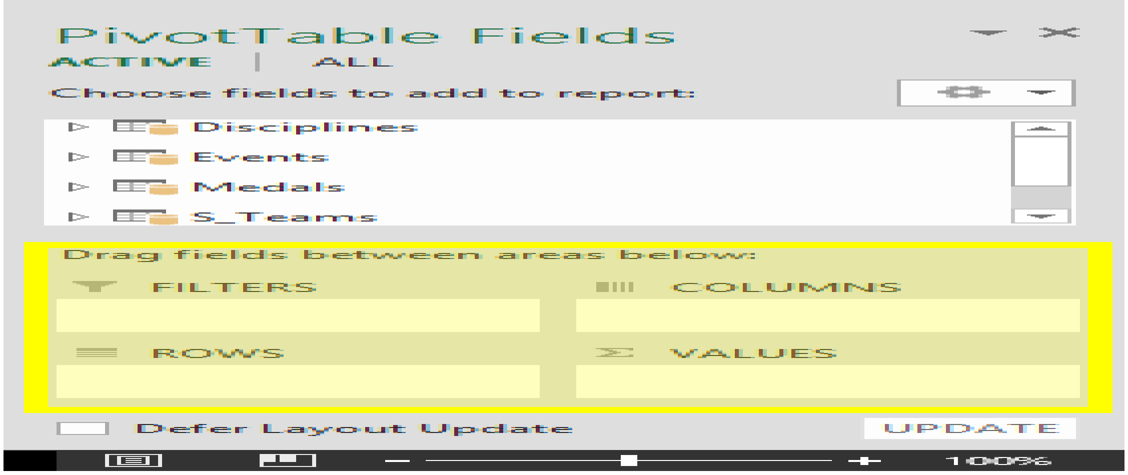


To build a report, choose fields from the PivotTable Field List

With the data imported into Excel, and the Data Model automatically created, you’re ready to explore the data.

Explore data using a PivotTable

Exploring imported data is easy using a PivotTable. In a PivotTable, you drag fields (similar to columns in Excel) from tables (like the tables you just imported from the Access database) into different **areas** of the PivotTable to adjust how it presents your data. A PivotTable has four areas: **FILTERS**, **COLUMNS**, **ROWS**, and **VALUES**.



It might take some experimenting to determine which area a field should be dragged to. You can drag as many or few fields from your tables as you like, until the PivotTable presents your data how you want to see it. Feel free to explore by dragging fields into different areas of the PivotTable; the underlying data is not affected when you arrange fields in a PivotTable.

Let’s explore the Olympic Medals data in the PivotTable, starting with Olympic medalists organized by discipline, medal type, and the athlete’s country or region.

1. In **PivotTable Fields**, expand the **Medals** table by clicking the arrow beside it. Find the NOC_CountryRegion field in the expanded **Medals** table, and drag it to the **COLUMNS** area. NOC stands for National Olympic Committees, which is the organizational unit for a country or region.
2. Next, from the **Disciplines** table, drag Discipline to the **ROWS** area.
3. Let’s filter Disciplines to display only five sports: Archery, Diving, Fencing, Figure Skating, and Speed Skating. You can do this from within the **PivotTable Fields** area, or from the **Row Labels** filter in the PivotTable itself.

Click anywhere in the PivotTable to ensure the Excel PivotTable is selected. In the **PivotTable Fields** list, where the **Disciplines** table expanded, hover over its Discipline field and a dropdown arrow appears to the right of the field. Click the dropdown, click **(Select All)**to remove all selections, then scroll down and select Archery, Diving, Fencing, Figure Skating, and Speed Skating. Click **OK**.

Or, in the **Row Labels** section of the PivotTable, click the dropdown next to **Row Labels** in the PivotTable, click **(Select All)** to remove all selections, then scroll down and select Archery, Diving, Fencing, Figure Skating, and Speed Skating. Click **OK**.

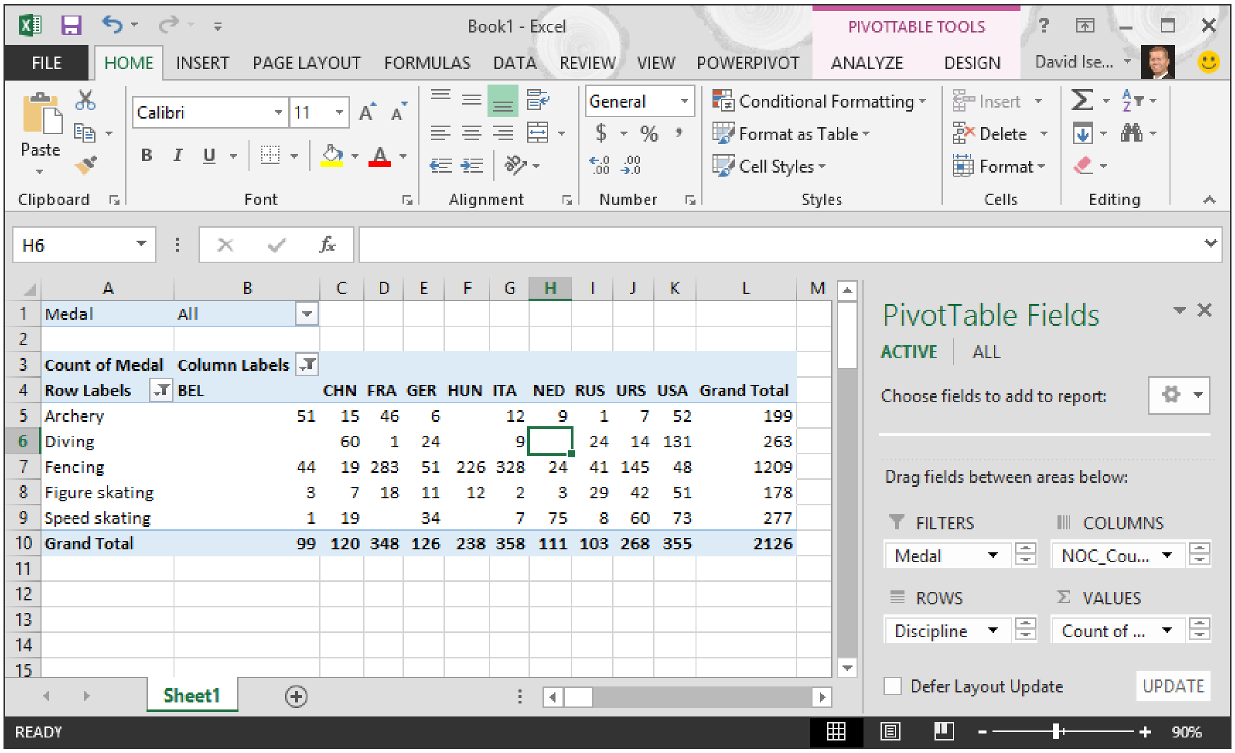
In **PivotTable Fields**, from the **Medals** table, drag Medal to the **VALUES** area. Since Values must be numeric, Excel automatically changes Medal to **Count of Medal**.

From the **Medals** table, select Medal again and drag it into the **FILTERS** area.

Let’s filter the PivotTable to display only those countries or regions with more than 90 total medals. Here’s how.

a. In the PivotTable, click the dropdown to the right of **Column Labels**.
b. Select **Value Filters** and select **Greater Than….**

Your PivotTable looks like the following screen.



Medal: All

Count of Medal	Column Labels										
Row Labels	BEL	CHN	FRA	GER	HUN	ITA	NED	RUS	URS	USA	Grand Total
Archery	51	15	46	6		12	9	1	7	52	199
Diving		60	1	24		9		24	14	131	263
Fencing	44	19	283	51	226	328	24	41	145	48	1209
Figure skating	3	7	18	11	12	2	3	29	42	51	178
Speed skating	1	19		34		7	75	8	60	73	277
Grand Total	99	120	348	126	238	358	111	103	268	355	2126

PivotTable Fields

ACTIVE ALL

Choose fields to add to report:

Drag fields between areas below:

FILTERS: Medal

COLUMNS: NOC_Cou...

ROWS: Discipline

VALUES: Count of ...

□ Defer Layout Update UPDATE

With little effort, you now have a basic PivotTable that includes fields from three different tables. What made this task so simple were the pre-existing relationships among the tables. Because table relationships existed in the source database, and because you imported all the tables in a single operation, Excel could recreate those table relationships in its Data Model.

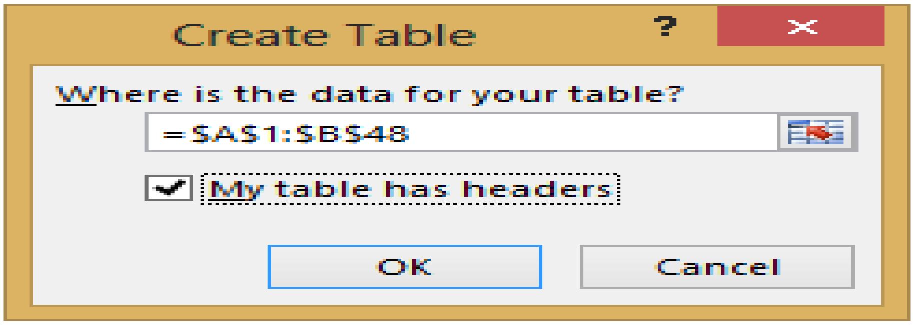
But what if your data originates from different sources, or is imported at a later time? Typically, you can create relationships with new data based on matching columns. In the next step, you import additional tables, and learn how to create new relationships.

Import data from a spreadsheet

Now let’s import data from another source, this time from an existing workbook, then specify the relationships between our existing data and the new data. Relationships let you analyze collections of data in Excel, and create interesting and immersive visualizations from the data you import.

Let’s start by creating a blank worksheet, then import data from an Excel workbook.

1. Insert a new Excel worksheet, and name it **Sports**.
2. Browse to the folder that contains the downloaded sample data files, and open **OlympicSports.xlsx**.
3. Select and copy the data in **Sheet1**. If you select a cell with data, such as cell A1, you can press Ctrl + A to select all adjacent data. Close the OlympicSports.xlsx workbook.
4. On the **Sports** worksheet, place your cursor in cell A1 and paste the data.
5. With the data still highlighted, press Ctrl + T to format the data as a table. You can also format the data as a table from the ribbon by selecting **HOME > Format as Table**. Since the data has headers, select **My table has headers** in the **Create Table** window that appears, as shown here.

   Where is the data for your table?

   =$A$1:$B$48

   ☑ My table has headers

   Formatting the data as a table has many advantages. You can assign a name to a table, which makes it easy to identify. You can also establish relationships between tables, enabling exploration and analysis in PivotTables, Power Pivot, and Power View.
6. Name the table. In **TABLE TOOLS > DESIGN > Properties**, locate the **Table Name** field and type **Sports**. The workbook looks like the

following screen.

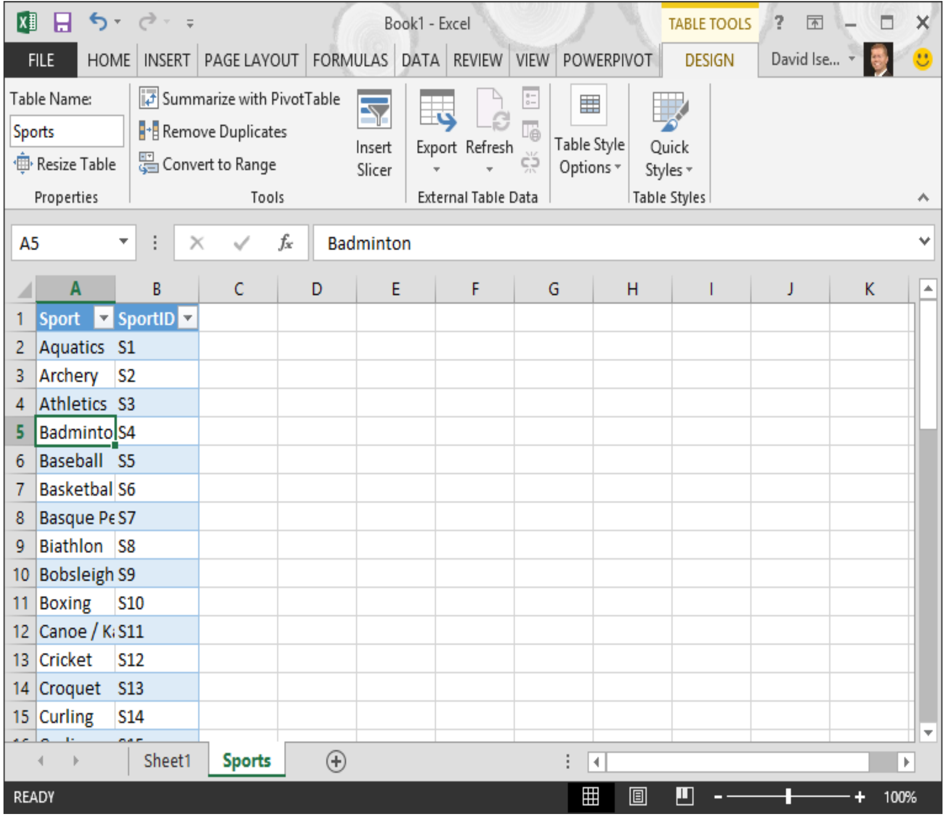


Table Name: Sports

A5 | Badminton

	A	B
1	Sport	SportID
2	Aquatics	S1
3	Archery	S2
4	Athletics	S3
5	Badminto	S4
6	Baseball	S5
7	Basketbal	S6
8	Basque Pe	S7
9	Biathlon	S8
10	Bobsleigh	S9
11	Boxing	S10
12	Canoe / K	S11
13	Cricket	S12
14	Croquet	S13
15	Curling	S14

7. Save the workbook.

Import data using copy and paste

Now that we've imported data from an Excel workbook, let's import data from a table we find on a web page, or any other source from which we can copy and paste into Excel. In the following steps, you add the Olympic host cities from a table.

1. Insert a new Excel worksheet, and name it **Hosts**.
2. Select and copy the following table, including the table headers.

Conclusion :-

In this way we import of data warehouse data in Microsoft Excel and create the Pivot table and Pivot Char.

Group No: 1

Assignment No: 5

Title of the Assignment :-

Perform the data classification using classification algorithm. Or Perform the data clustering using clustering algorithm.

Objective of the Assignment :

To introduce the concepts classification using classification algorithm. Or Perform the data clustering using clustering algorithm.

Prerequisite:

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

Theory:

K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

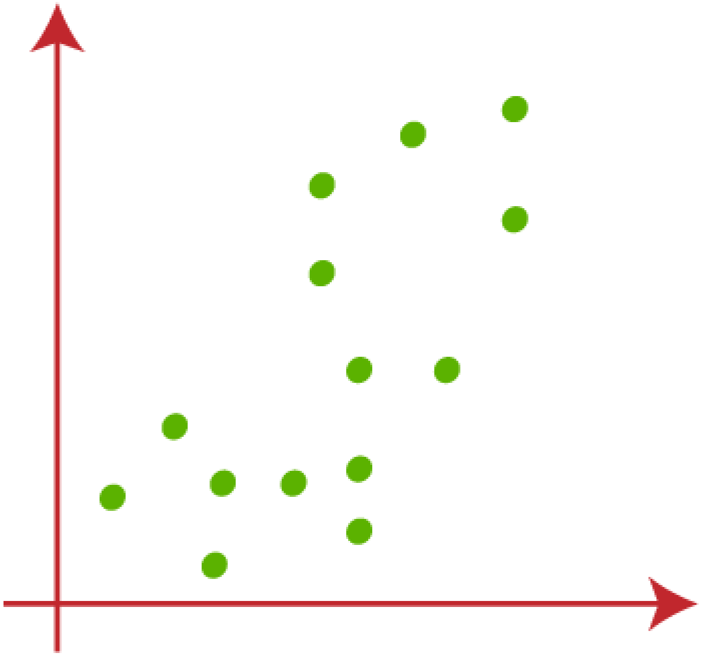
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

Let's understand the above steps by considering the visual plots:

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



KMeans Clustering for Classification

Clustering as a method of finding subgroups within observations is used widely in applications like market segmentation wherein we try and find some structure in the data. Although an unsupervised machine learning technique, the clusters can be used as features in a supervised machine learning model.

KMeans is a clustering algorithm which divides observations into k clusters. Since we can dictate the amount of clusters, it can be easily used in classification where we divide data into clusters which can be equal to or more than the number of classes.

I’ll be using the MNIST dataset which comes with scikit learn which is a collection of labelled handwritten digits and use KMeans to find clusters within the dataset and test how good it is as a feature.

I have created a class named clust for this purpose which when initialized takes in a sklearn dataset and divides it into train and test dataset.

The function KMeans applies KMeans clustering to the train data with the number of classes as the number of clusters to be made and creates labels both for train and test data. The parameter output controls how do we want to use these new labels, ‘add’ will add the labels as a feature in the dataset and ‘replace’ will use the labels instead of the train and test dataset to train our classification model.

- **Conclusion** :-

In this way we import data clustering using clustering algorithm.