MSBI COURSE TRAINING MATERIAL

IT INCLUDES

1. DATA WAREHOUSING CONCEPTS
2. SQL SERVER INTEGRATION SERVICES
3. SQL SERVER REPORTING SERVICES
4. SQL SERVER ANALYSIS SERVICES
MS-BI [Microsoft Business Intelligence]

MS-BI:

Business intelligence is a process or set of activities such as extracting the business data from various operational sources, transform, and load the data into consolidated database or data warehouse systems.

That consolidated data should be reported and analyzed to make better decisions in order to improve the organization.

Note:

Simply Business Intelligence is a broad category of applications and technologies for gathering, storing analyzing and providing access to data to help enterprise users make better business decisions.

BI Applications:

BI applications include the activities of decision support systems, query and reporting, online analytical processing, statistical analysis, forecasting and data mining.

Dimensional Data Model:

Dimensional data model is used in data warehousing systems that means designing facts, dimensions, hierarchy.

1. Dimension table:

The Dimension Table provides hierarchy and detailed information about the attributes.

   For example:

   Dim product, Dim customer and Dim time etc.

2. Fact table:

A fact table is a table that contains measures.

Note:

Measure is a numerical value and it is key value to analyze your business data and it also evaluates the performance of the organization.

   For example:
Sales amount would be a measure. This measure stores in fact table.

3. **Hierarchy:**

   The specification of levels that represents relationship between different attributes with in a dimension.

   **For example:**

   One possible hierarchy in the time dimension is, Year
   
   \[ \begin{array}{c}
   \text{Year} \\
   \downarrow \\
   \text{Quarter} \\
   \downarrow \\
   \text{Month} \\
   \downarrow \\
   \text{Week} \\
   \downarrow \\
   \text{Day}
   \end{array} \]

4. **Dimension (Attribute):**

   A unique level with in a dimension is known as attribute.

   **For example:**

   PID, PName are attributes or dimensions in Dim product.

**Data warehouse:**

Data warehousing is a Relational data base it has its own characteristics.

- Time variant.
- Integrated Data Base.
- Non – Volatile.
- Subject oriented.

1. **Time variant:**

   Data warehouse is a time variant data base source, the business users perform analysis on their business information with respect to various time periods.
Example:

Year
↓
Quarter
↓
Month
↓
Week
↓
Day

From the above example the business users or business analytics can analyze their business information with respect to year wise or quarter wise or month wise or week wise or day wise as well.

2. Integrated Data Base:

Data warehouse is a built by integrate the data various operational sources into single data base.

For example:
From the above example integrating the data from various operational sources (Excel, OLEDB, flat file source) into data warehouse.

Simply data warehouse can be by integrate the data from various operational sources.

3. Non–Volatile:
   
   Once the source data is inserted into the data warehousing it doesn’t reflect the changes since it is static or read only data.

   **For example:**

<table>
<thead>
<tr>
<th>Policy ID</th>
<th>Policy Name</th>
<th>Customer Name</th>
<th>Address</th>
<th>Modified date</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>ULP</td>
<td>Rama</td>
<td>LIG-13</td>
<td>2009-12-12</td>
</tr>
<tr>
<td>123</td>
<td>ULP</td>
<td>Rama</td>
<td>LIG-14</td>
<td>2010-10-12</td>
</tr>
<tr>
<td>123</td>
<td>ULP</td>
<td>Rama</td>
<td>LIG-15</td>
<td>2010-10-13</td>
</tr>
</tbody>
</table>

4. Subject oriented:
   
   Data warehouse is a subject oriented data base and it stores specific data about specific department in the complete organization. It is also known as Data mart.

   **For example:**

   ![Diagram](image_url)

   **Note:**

   Data mart is also known as HPQS. (High Performance Query Structures)
**Data warehousing Architectures:**

In designing data models for data warehouse or data marts, the mart commonly used schema types are,

1) Star schema.
2) Snowflake schema.

**Star schema:**

The star schema data warehousing design contains at least one fact table and surrounded by dimension tables like a star. Each dimension is represented as a single table. The primary key in each dimension table is related to foreign key in the fact table.

**Note:**

i) A simple star schema consists of one fact table and a complex star schema have more than one fact table.

ii) All measures in the fact table are related to all the dimension table.

**Example:**

- Cust ID (P key)
- PID (P key)
- Cust ID (FK)
- PID (FK)
- Fact sales (cost)
- Time key (Fk)
- Add ID(P key)
- Add ID(FK)
- Time key (P key)
- Dim customer
  - [cust ID Name]
- Dim product
  - [PID PName, colour]
- Dim Address
  - [Add ID H.NO, State country]
- Dim time
  - [time key year]
From the above diagram the fact table contains customer ID, PID, Address ID, cost and time key attributes.

**Note:** the different dimensions are not related to one another.

**Snowflake schema:**

The snowflake schema is an extension to star schema, where each point of the star schema explodes or divides into more points. In star schema each dimension is represented by a single dimension table, where as in a snowflake schema the dimension table is normalized into multiple lookup tables, each representing a level in the dimensional hierarchy.
Lookup table

In the above data warehousing schema example, we have three lookup tables (Dim category, Dim product sub category and Dim address type). Product sub category lookup table is connected to product category and Add type is connected to Add Dimension.

**SQL Server Integration Services (SSIS)**

SQL Server Integration Service is an ETL (Extraction Transformation Loading) tool. By using SSIS we can create data transformation service (extract data from various operational sources like Excel, flat files, SQL Server, Oracle etc.) Transform the source business data by using existing transformation in staging area or transformation area and load and store it into destination data base or file system.
SSIS Architecture is categorized into two components.

- SSIS Runtime Engine
- Data flow transformation pipeline engine.

1) **SSIS Runtime Engine:**

The SSIS Runtime Engine completely handles the control flow of the package.

**Control flow:**

Control flow of a package defines actions that are to be executed when the package run. The control flow contains various tasks and containers as well.

**Task:**

A unit of work in a work flow.

**For example:** Data flow task, execute SQL task etc.

**Container:**

Container is used to divide the package into multiple blocks.

**For example:** For loop container, for each look container, sequence contains and task host container.

2) **Data Flow Transformation Pipeline Engine:**

The Data flow transformation pipeline engine completely handles the data flow of the package. The data flow contains, data flow sources (excel source, flat file, OLEDB etc), data flow transformations (conditional split transformation) derived column transformation, lookup transformation etc.) and data flow destination.

**Note:**

Whenever the data flow task occurs in control flow the SSIS Runtime Engine throws the control from control flow to data flow to achieve or to run ETL process while the package is running.

**Connection Manager:**

A logical connection between SSIS application and data base or file system.

**Note:**

Connection Manager can be established by using various providers in SSIS.
**Package:**

Package is a core component of SSIS. Package can be created simple graphical user interface or programmatically.

**Execute SQL Task:**

Execute SQL Task is used to execute relational queries such as DDL, DML against connection.

**Basic Parameters or Properties of Executive SQL Task:**

**Connection:**

In Execute SQL Task connection is nothing but Connection Manager. Provide the following steps to create Connection Manager in Execute SQL Task.

1. Open Business Intelligence Development Studio
2. Create a new package and rename it as execute SQL.dtsx
3. In control flow drag and drop execute SQL Task onto design area
4. Double Click on Execute SQL Task to edit it
5. Provide the following steps
   - Select New Connection
   - Click New
   - Provide Server Name (Localhost (or) . (or) Server name)
   - Select Adventure Work database from drop down list
   - Click test Connection to evaluate the connection between database and SSIS
   - Click ok
   - Click ok
SQL Source type – Select Direct Input (Default)

Note:

Here, we have 3 options

i) **Direct input:**

   User can provide or enter SQL command directly on available notepad.

ii) **File Connection:**

   We can the query through any file system by providing path and file name.

iii) **Variable:**

   We can pass the query through available which is already declared it on SSIS variable section.

**SQL Statement:**

```
Truncate Table .any valid Table-Name
Click ok to save the changes
In Solution Explorer (Alt + Ctrl + L), Select Execute SQL .dtsx
Right Click and Select Execute Package option (2nd option)
```

**Execute Package Task:**

The Execute Package Task is used to execute a package within the Parent package.

**Steps to Configure Execute Package Task:**

Open Business Intelligence Development Studio in Solution explorer

Create a new package and rename it as Exec Pkg .dtsx

In Control Flow, drag and drop execute package task

Rename the Execute Package Task as EPT calling Exec SQL Package

Double Click on Execute Package Task to Configure it

Select Package option from left pane and set the following properties
Location – Select File System
Connection – Select New Connection

- Click Browse
- Navigate to the path where the Package is stored in,
- Select Execute SQL .dtsx
- Click open
- Click ok

In solution Explorer, Select Execute Package .dtsx

- Right Click and Select Execute Package option
- The linked Package will be automatically set opened by current package and then executes.

**Execute DTS 2000 Package Task:**

This task is used to execute a package designed on SQL server 2000 environment

**Steps to configure execute DTS 2000 package task:**

- In control flow, drag & drop execute DTS 2000 package task
- Double click on it and set,
- Storage location-select structured storage file
- File- Click browse and select any package (.dts) and click open
- Package name-specify the package name, which you would to execute it on SQL Server 2005/2008 environment
- Click ok
- Execute package
Send Mail Task:

Send mail task is used to send an E-mail message to the recipients via SMTP connection.

Steps to configure send mail task:

Drag & Drop send mail task (In Control flow)

Double click on send mail task to configure it

Provide the following information

SMTP connection - select new connection and provide server name, click ok

From - Provide from Email address

To - Provide To email address

CC - Provide carbon copy email address (If any)

Bcc - Provide blind Carbon copy email address (if any)

Subject - Test mail

Message Source type - Direct input

Message source - Click browse & type body of mail, click ok

Priority - High

Attachments - Click browse

Select any file to be attached to the mail and click open

Example for SMTP connection - ASIHXCE.corporateit.com

Bulk Insert Task:
Bulk Insert Task can be used to data from a text file or a flat file into SQL server table. Mainly it is advised to use in applications which require quick loading of large amounts of data.

**Steps to configure Bulk Insert table:**

Prepare the following data within a text file and rename it as student details .txt the sample flat file can be used for the above mentioned methods (Method 1 and Method 2)

1. Rama, mca, 72
2. vizay, btec, 71
3. Siva, Mca, 70
4. laksmi, Mca, 80
5. Arpitha, Bsc, 90

Open Business Intelligence Development Studio

Create a new package and rename it as bulk insert .dtsx

Control flow drag and drop bulk insert task

In connection manager section, right click and select new flat file connection

Provide connection manager name and description

Click browse and select student details.txt

Click open

Select advanced tab and rename the column names as mentioned below

Column0 – Sno
Column1 – Sname
Column2 – Qualification
Column3 – Marks

Click ok

Open SSMS (SQL server Management Studio) to create destination table to store the data which is coming from student details.txt

USE (Adventure works)

Go
CREATE TABLE [dbo].[Bulk insert]

([SNO] [Varchar](50),
[SName] [Varchar](50) NULL,
[Qualification] [Varchar] (50) NULL,
[Marks] [Varchar] (50) NULL, Press
F5 to execute the query in SSMS

Double click on bulk insert task to configure it and set the following properties

- Source Connection
  File – Select student details from drop down list
- Destination connection
  Connection – select local host .advent
- Use work
  Destination table – select [Adv.works] . [Dbo]
  [Bulk Insert] from drop down list
  Column Delimited – Select comma {,} option since
  Student details flat file is comma {,} delimited
  Select options tab or page

1) Batch Size:

Batch size specifies the no .of rows in a batch. Each batch is copied to the destination
server as 1 Trans direction .server commits or role back in case of failure for each batch.

Note:

The default value is zero i.e. all the data in the specified data file is 1 batch.

2) Last Row:

The last row option specifies the row at which the insertion of the data stops. The
default is zero, indicates the last row in the specified data file.

3) First Row:

Specifies at which the insertion of the input data begins. The default is 1, indicates the
1 st Row in the specified data file.

TIPS to improve the performance of bulk insert task:

i) It is advised not to perform any sort operation while loading the data from text file
using bulk insert task.
ii) The data will be loaded faster when there are no indexes created on staging table or destination table.

**Data Conversion Transformation:**

Data Conversion transformation is used to convert the data from one data type to another data type and also adds a new column to the data set.

**Steps to configure data conversion transformation:**

Go to Start, select All Programs and Select Microsoft SQL Server 2005

Select SQL Server Business Intelligence Development Studio

Go to File Menu, Select New Project Business

Intelligence project options under project types

Integration Services Project under Templates section

Change the Project name & location then Click ok Select package. Dtsx in solution explorer and rename it.

In Control Flow task, drag & drop data flow task and rename it as data flow task space data conversion.

Select the data flow task and right click and select edit option from right click popup menu.

In data flow task drag & drop OLEDE Source.

Double click on OLEDB source to edit it.
In connection manager page, click new to create new connection manager. Click new, provide server name & database name.

Click test connection to evaluate the connection, then click ok.

Click ok.

Select table or view option from data access mode drop down list.

Select columns option from left pane & click ok.

Drag & drop data conversion transformation and make a connection from OLEDB Source to data conversion transformation.

Double click on data conversion transformation.

Check title and marital status from available input columns and change the data type from [DT_WSTR] to string [DT_STR] and also rename the output alias columns as Title DC and Marital status DC. Click ok to save the changes.

Drag and Drop OLEDE destination from data flow destination section.

Make a connection from data conversion transformation to OLEDB destination.

Double click on OLEDB destination.

In connection manager page click new to create destination connection manager.

Click new.

Provide destination server name (Local host) and Select Adventure works database from the drop down list.

Click test connection.

Click ok.
Click OK

Click new to create destination table remove copy of Title and copy of Martial status and rename the table as converted data.

Click OK

Select mappings options from left panel to make a mapping between input columns and Destination Columns.

**Steps to execute SSIS package:**

In Business Intelligence Development Studio, Alt + Control + L for Solution explosion, select data conversion .dtsx package

Right click select execute package

**Derived Column Transformation:**

The Derived Column Transformation enables in-line transformations using SSIS expressions to transform the data. The typical uses of Derived Column transformation is to create or derive new columns by using the existing source columns or by using variable available functions.

**Steps to Configure Derived Columns Transformation:**

Start

All Programs

Microsoft SQL Server 2005

Select Microsoft Business Intelligence Development Studio

Select File Menu

Select New Projects

Select Business Intelligence Projects Option
Integration Services Project

Change Project Location and Name

Click Ok

In Business Intelligence Development Studio, in control flow drag and drop data flow task and rename it is data flow task space derived column.

Double Click on data flow task to edit it

In data flow, drag and drop OLEDB Source

Double Click on OLEDB Source to configure it Select

Click new to create source connection manager. Click new

In the connection manager editor provide server name [local host (or) . (or) server name]

Select Adventure works from the drop down list

Click ok twice

Select human resources . employee address table – from the drop down list

Select columns from left panel

Click ok

Open SSMS (SQL Server Management Studio) and run the following query to create destination table.

Create Table [Derived column] (  

[Employee ID] Integer,

[Address ID] Integer,

[Row guid] Unique Identifier,
Go to Business Intelligence Development Studio and drag and drop derived column transformation and make a connection from OLEDB Source to derived column using green data flow path.

Double Click on derived column

Define the following expression

<table>
<thead>
<tr>
<th>Derived Column Name</th>
<th>Expression</th>
<th>Data type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref data</td>
<td>(DT_DB Date) get date</td>
<td>DT_DB date</td>
<td></td>
</tr>
</tbody>
</table>

Note:

The above expression is defined by drag and drop get date time functions section and remove the derived column 1 as Ref date the same will be carry forwarded to destination in our scenario.

**Flat File Source:**

**Scenario:**

Remove duplicate records from file systems using either sort transformation or aggregate transformation.

**Aggregate transformation:**

**Method 1:** (Using Aggregate Transformation)

Open Business Intelligence Development Studio

Create a new package and rename it a Flat File source with Aggregate .dtsx

In control flow drag and drop data flow task and rename it as avoid duplicate records using aggregate
In data flow drag and drop flat file source

Double click on flat file source to configure it

Click new to create a new flat file connection manager

Provide connection manager name and description (storing details)

Click Browse

Navigate to the path

Select student details .txt and click open

Select columns from left pane

Click ok

Rename the output columns as mentioned below

Column 0 – SNO
Column 1 – SName
Column 2 – Qualification
Column 3 – Marks

Click ok

Drop and drag aggregate transformation and make a connection from flat file source to aggregate

Double click on aggregate to configure it

Select or check all available input columns

(SNO, SName, SQualification, Marks)

Note: Make sure that the operation is group by for all the selected columns

Click ok

Drag and drag OLEDB destination and make a connection from aggregate to OLEDB destination

Double click on OLEDB destination to configure it
Provide destination connection manager

Click new to create new destination table if it is not exists

Rename the table as aggregated data

Select Mappings

Click ok

Execute package

**Method 2:** Remove duplicate records and also sort the data using sort transformation

**Sort the data using sort transformation:**

Open Business Intelligence Development Studio

Create a new package and rename it as avoid duplicate using sort .dtsx

In control flow drag and drop data flow task and rename it as DFT sort transformation

In connection manager name and description

Click browse and select student details flat file connection manager from drop down list

Select columns and rename the output columns as mentioned below.

- **Column 0 – SNO**
- **Column 1 – SName**
- **Column 2 – Qualification**
- **Column 3 – Marks**

Click ok

Drag and drop sort transformation and make a connection from flat file source to sort

Double click on sort to configure it and Select all input columns

Check remove rows with duplicate sort values to remove duplicate records from the sorted data sets
Click ok

Drag and drop OLEDB destination and make a connection from sort to OLEDB destination

Double click on OLEDB destination

Provide destination connection manager if exists and click new
to create a new destination table if not exists

Rename the new table as sorted data

Click ok

Select mapping from left panel

Click ok

**Flat file formats:**

In SSIS flat file can be configured in 3 ways

1) **Delimited:**
   
   Every column is delimited by a special character except the last column which is delimited by a new line character (carriage return or line feed) \{CR\} {LF}

   **Example:**
   
   1, Rama, MCA
   2, Vijay, BTech

2) **Fixed width:**

   The columns are defined by fixed width

3) **Ragged Right:**

   The columns are defined by fixed width except the last column which is delimited by new line character.

**OLEDB Command Transformation:**

It is used to execute SQL command that works for each and every record (or) row in a data sheet.

**Note:** OLEDB command works like cursor in the SQL (or) T-SQL
**Steps to configure OLEDB command:**

Open Business Intelligence Development Studio

Create a new package and rename it as OLEDB command.dtsx

In Data Flow drag and drop OLEDB source

Double click on OLEDB source to configure it

Provide connection manager if exists and select SQL command option from Data access mode drop down list.

Provide the following SQL command text select * from Human Resources.

Employee where Employee 1D=1

Select columns and click ok

Drag and drop OLEDB command and make a connection from OLEDB source to OLEDB command

Double click on OLEDB command to configure it.

In connection manager tab select the connection manager.

In components property tab, set SQL command - SQL command---- update

Human Resources. Employee,

Set marital status='D’ where employee 1D=1

Click Refresh and click ok

Execute package
**Fuzzy Look Up Transformation:**

It is used to compare the data from source to reference dataset by using Fuzzy matching [closely matching].

**Note:** It can be used to identify the miss spelled customers in customers dimensional table.

Fuzzy look up uses Error Tolerant Index (ETI) to find matching rows in the reference table.

Each record in the reference table is broken up into tokens or words. If you take the name as an example from the Customer Dimension Table and if your reference dataset contains Rama Rao, the ETI will contain Rama and Rao by using Token delimiters.

**Steps configure Fuzzy Look Up**

1. Open Business Intelligence Development Studio
2. Create a new package and rename it as Fuzzy lookup. dtsx
3. In control Flow drag and drop Data Flow Task
4. In Data Flow drag and drop OLEDB source
5. Double click on OLEDB source to configure it
6. Provide source connection manager if exists
7. Select production, product category table from the dropdown list.
8. Select columns
9. Click ok
10. Drag and drop Fuzzy Look up transformation and make a connection from source to Fuzzy Lookup
Double click on Fuzzy Lookup to configure it.

In Reference Tab,

Connection - Provide destination connection Manager

Reference Table Name – Select [production].[product sub category]

In columns Tab,

Make a join or Mapping between product category ID from input column to output column. Click ok

Drag and drop OLEDB destination and make a connection from Fuzzy Lookup to OLEDB destination.

Provide destination connection Manager and click new to create a new table Rename it as Fuzzy Lookup

Select Mapping

Click ok

Execute package

**Look Up Transformation:**

It is used to compare the data from source to reference data set.

**Note:** While implementing Data Warehousing the reference data must be dimension table.

**Steps to implement Look Up Transformation:**

Open Business Intelligence Development Studio

Create a new package and rename it as lookup.

Dtsx In Control flow drag & drop Data flow task

In Data flow, drag & drop OLEDB source

Double click on OLEDB to edit it
Provide Connection Manager if exists and select production. product category

Select columns & click ok

Drag & Drop Look Up transformation and make a connection from source to look up

Double click on look up to edit it

Provide connection manager and select production .product sub category

Click on Configure Error Output and set Redirect Rows and Error header,

Click Ok

Click Ok

Drag and Drop a OLEDE destination and make a connection from Look Up to destination.

Double click on OLEDB destination to configure it

Provide destination connection manager and click new to create a new destination table and rename the table as Matched_data

Click Ok & Select Mappings

Drag & Drop one more OLEDB destination to capture unmatched records from source to reference data set

Make a connection from look up to OLEDB destination using error output(red data flow path)

Click ok in look up error output editor

Double click on destination to configure it

Provide destination connection manager and click new to create destination table and rename it as unmatched_records

Select mappings and click ok

Execute Package

**Note:**
1. Matched records will be updated to the destination.
2. All the Unmatched records will be inserted to destination.
3. In Look Up transformation editor remove the mappings between all the columns.
except between product category ID.

**Pivoting and Unpivoting**

The presentation of the data is required for easy analysis turning columns into rows and rows into columns is another way of presentation of data. So that the end user can understand it easily.

**Unpivot:**

A process of turning columns to rows is known as Unpivot.

**Steps to configure Unpivot**

Prepare the following excel sheet:

<table>
<thead>
<tr>
<th>Year</th>
<th>Category</th>
<th>Jan</th>
<th>Feb</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Bikes</td>
<td>100</td>
<td>200</td>
<td>300</td>
<td>400</td>
</tr>
<tr>
<td>2008</td>
<td>Accessories</td>
<td>200</td>
<td>270</td>
<td>300</td>
<td>320</td>
</tr>
<tr>
<td>2009</td>
<td>Components</td>
<td>100</td>
<td>120</td>
<td>300</td>
<td>150</td>
</tr>
<tr>
<td>2009</td>
<td>phones and Components</td>
<td>400</td>
<td>800</td>
<td>400</td>
<td>300</td>
</tr>
</tbody>
</table>

Open Business Intelligence Development studio

Create a new package and rename it as unpivot.dtsx

In control flow drag and drop data flow task

In data flow task drag and drop excel source Double click on excel source to configure it Click new Click browse

Select unpivot.xlsx file and click open Click ok
Select sheet 1 from the drop down list. Select columns and click ok.

Drag and drop unpivot transformation and make a connection from the excel source to unpivot.

Double click on unpivot.

Select below columns to Unpivot Jan, Feb, March and April.

Rename pivot key value column name as months.

Specify sales amount as a derived or destination column for all selected pivoted key values or input column values.

Click ok.

Make sure that excel source file is closed.

Drag & drop excel destination.

Make a connection from Unpivot to excel destination.

Double click on excel destination to configure it.

Click new to create new destination excel sheet and rename it as unpivot data.

Click ok.

Select mappings.

Execute package.

**Pivot:**

A process of turning rows to columns is known as pivot.

**Steps to configure pivot:**

Prepare following excel sheet for source data.
<table>
<thead>
<tr>
<th>Year</th>
<th>Quarter</th>
<th>Sales amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>100</td>
</tr>
<tr>
<td>2009</td>
<td>Q2</td>
<td>200</td>
</tr>
<tr>
<td>2009</td>
<td>Q3</td>
<td>300</td>
</tr>
<tr>
<td>2009</td>
<td>Q4</td>
<td>400</td>
</tr>
</tbody>
</table>

Open Business Intelligence Development studio

Create a new package and rename it as pivot.dtsx

In control flow drag and drop data flow task

In data flow task drag and drop excel source

Double click on excel source to configure it

Click new

Click browse

Select pivot.xlsx file and click open

Click ok

Select sheet 1 from the drop down list

Select columns and click ok

Drag and drop pivot transformation and make a connection from the excel source to pivot

Double click on pivot

Select input columns tab and check all input columns
Select input output properties tab

Expand pivot default output

Expand input columns

Select year and set pivot usage-1

Select Quarter input columns and pivot usage-2

Select sales amount input column and set pivot usage-3

Pivot Usage:

Pivot usage tells SSIS how to treat the data what its role during the transformation process.

0-The column is not pivoted
1-The column is part of the set key that identifies 1 or more rows as part of 1 set. All input the same set key are considered into 1 output row.
2-The column is pivot column
3-The value from this column that are created as a result of the pivot

Expand pivot default output and create the following columns by click add column button.

Year, Q1, Q2, Q3, Q4

Copy or Note the Lineage ID of input column year

Select year output column and set,

Pivot key value - Q1
Source column - 64
Select Q2 and set
Pivot key value-Q2
Source column-64

And follow the same process for Q3&Q4 output columns
Click refresh

Click ok

Drag & drop excel destination

Double click ok excel destination to edit it

Provide Excel connection manager

Click new to create a new table (sheet) and rename it as pivot data

Click ok twice

Execute package

For Each Loop Container:

For Each Loop Container is used to loop through multiple files, data sets, xml files, multiple sheets from a single excel work book.

The above mentioned operations can be achieved by available enumerators in collection tab.

Enumerators:

1. For each file enumerator:
   For each file enumerator is used to enumerate or loops through a collection of files in the specified location.

2. For each item enumerator:
   For each item enumerator is used to enumerate or loops through items in a collection.

3. For each ADO.Net Schema row set enumerator:
   It is used to enumerate or loops through schema information about a data source.

Ex:
To loop through a list of tables from the specified data base.

4. For Each Node List Enumerator:
   It is used to enumerates a collection of XML files

File System task
It can be used to perform certain operations on files such as copy file, Move file and rename file and also perform operations on directories as well as copy, create, delete and move directories.

**Import column Transformation:**

It is used to load binary data (photos, documents and media etc..) from the file system into data flow. From the data flow it can then be loaded into a table.

**Steps to Configure Import column Transformation:**

1. Open Business Intelligence Development Studio
2. Create a new package and rename it as Import column. dtsx
3. In control Flow drag and drop data flow task and it as DFT import column.
4. In Data Flow drag and drop OLEDB source
5. Double click on OLEDB source to configure it
6. Provide source connection manager if exists
7. Select production. Product photo
8. Select columns and click ok
9. Drag and drop Import column transformation and make a connection from source to Import column.
10. Double click on import column transformation to configure it.
11. Select input columns and click thumbnail photo file name. input column (the input column must be of type DT_STR or DT_WSTR)
12. Select Input and output properties tab and expand input column and expand output columns.
13. Click add column to add a new output column and rename it as image file and note down the ID of image file output column.
14. Expand Import column Input and expand Input columns.
Select thumbnail photo file name and set, File Data column ID- 65 (ID of output column i.e. Image file).

**Script Component:**

It is used to execute scripts on Microsoft Visual Basics. Net to perform certain operations.

**Steps to configure script component**

Open Business Intelligence Development Studio

Create a new package and rename it as script component, dtsx

In control Flow drag and drop Data Flow Task

In Data Flow drag and drop script component and select “source” radio button as script component type [Here script component is a source in the data flow and provides data to the output column].

Click ok

In inputs and outputs page Expand output 0

Select output column and click add columns to add new output columns which will be carry forward it to next level.

Rename the column 0 as SNO and column 1 as SName

Select script page and click design script.

In Microsoft Visual Studio for application editor providing the fall code to add new rows by adding Add row () method.

```
Output 0 Buffer. Add Row ()
Output 0 Buffer. SNO=123
Output 0 Buffer. SName="Rama"
```
Click ok and close VS for application Editor.

Drag and drop OLEDB destination and make a connection from script component to OLEDB destination.

Double click on destination to configure it and provide destination connection manager.

Click new to create destination table and Rename it as script component.

Click ok

Select mappings

Click ok

WMI (Windows Management Instrumentation) Data reader task:

It is used to read the data about a system using WMI Query language.

Steps to configure WMI data reader task:

Open Business Intelligence Development Studio

Create a new package and rename it as WMI Data reader task. dtsx

In control Flow drag and drop WMI data reader task.

Double click on it to configure and set

WMI connection – New connection

Check windows authentication

Check box

Check text to evaluate the connection

Click ok

WQL Query source Type – Direct input (default)

WQL Query source – Select name, start mode from win32-service

Destination Type – File connection(default)
Destination - New connection
Create file
Click Browse
Provide file name as WIN32-Service details.csv
Click open
Click ok

Example WMI Queries:

Below query used to display the installed software’s list from Microsoft Manufacturer:
select* from Win32-pnp Entity where manufacturer='Microsoft'

Deployee project (SSIS) to integration services server:

In BIDS select “Project Menu” and select properties (last option)
Select deployment utility and set create deployment utility – True
Deployment output path – bin/deployment
In solution explorer select project and Right click
Select Rebuild
Open Windows Explorer and navigate to the project go to ‘bin’ folder and go to deployment folder, select integration services deployment manifest file with name as project [Morning 8:30]
Select Manifest File and right click and select deployee
In welcome to the package installation wizard
Click next
Check file system deployment radio button and also check validate package after installation check box
Slowly Changing Dimension (SCD):

In SSIS slowly changing dimension or SCD is categorized into 3 parts.

1. **Changed Attribute**: (Type I in terms of Data warehousing)
   Select this type when changed values should overwrite with existing values.

2. **Historical Attribute**: (Type II)
   Select this type when changes in a particular column values are saved as new records and previous values as stored as expired records.

3. **Fixed Attributes**: (Type III in terms of Data Warehousing)
   Select this type when the value in a column should not change. Changes are treated as errors.

**Steps to configure slowly changing Dimension:**

Open Business Intelligence Development Studio

Create a new package and rename it as SCD.dtsx

In control flow drag & drop data flow task and rename it as DFTSCD.

Open SQL Server Management Studio and run the following SQL Statements to create temporary tables for source and destination.

Table for source

Select into Employee_Address from Employee
In data flow drag & drop OLEDB source

Double click on OLEDB source to configure it

Provide connection manager if exists

Select Employee_Address from drop down list

Drag & drop SCD and make a connection from OLEDB source to SCD

Double click on SCD to configure it

In welcome to the slowly changing dimension wizard, Click next

Provide destination connection manager and select employee destination and set Login Id-

Business Key

Click next

Birth Date-Fixed Attribute

Gender- Fixed Attribute

Contact Id- Fixed Attribute

Hire Date- Fixed Attribute

Marital Status- Historical Attribute

Sick Leave Hours-Historical Attribute

Title-Change Attribute

Click Next

Make sure that fail the transformation, if changes are detected in the fixed attribute check box is checked.

Click next

Use a single column to show current and expire records radio button.

Column to indicate current record-Select Nation ID Number
Value when current-Select Current
Expiration value-Expired
Click Next
Click finish

Note:
For Cross – Testing SCD, make some changes in source table for a fixed attributes
(update [Human Resources].[Employee] set Gender='M' Where Employee ID =1
Execute Package

Row sampling Transformation:
It takes fixed number of rows from source data set. It splits the data set into 2 sets, sample and Unsampled output.

Steps to Configure Row Sampling:
Open Business Intelligence Development Studio
Create a new package and rename it as row sampling. Dtsx
In control flow drag and drop data flow task
In data flow task drag & drop OLEDB source
Double click on OLEDB source to configure it
Provide source Connection Manager if exists
Select Human Resources. Employee table from the drop down list
Select columns and click ok
Drag and drop Row sampling transformation and make a connection from source to Row Sampling
Double click on row sampling to configure it and set,

Number of rows – 10

Click ok

Drag & Drop OLEDB destination and make a connection from row sampling to destination.

Select sampling selected output in input output selection editor

Click ok

Double click on OLEDB destination to configure it.

Provide destination connection manager and click new to create destination table and rename it as row sample.

Click ok twice

**Note:** If you want to test your package with the same data set, use Random seed.

**Percentage Sample:**

Percentage sampling splits a data set by randomly directing rows to one of two outputs. We need to set the percentage of rows split into the 2 inputs (sampling & Unsampling).

**Steps to Configure Percentage Sampling:**

Open Business Intelligence Development Studio

Create a new package and rename it as percentage sampling.dtsx

In data flow task drag & drop OLEDB source & configure it.

Drag & Drop percentage sampling and make a connection from source to percentage sampling

Double click on percentage sampling to edit it and specify percentage of rows 10

Click ok
Drag & Drop OLEDB destination and configure it.

**Audit Transformation:**

Audit Transformation is used to display audit information such as Package name, Execution start time, Task name, User name, Machine name etc.

**Copy Column:**

Copy column transformation is used to copy input columns.

Example: Gender-Copy of Gender

**Character Map:**

Character Map transformation is used to apply string operation in the selected data set.

**Steps to configure Audit, Copy column and character map transformation:**

Create a new package

In data flow task drag & drop OLEDB source

Double click on OLEDB source to configure it

Provide connection manager if exists and select Human Resources. Employee from the drop down list

Drag & drop audit transformation and make a connection from source to audit

Double click on Audit and select the following audit type,

Package name
Task name
Machine name
User name

Click ok

Drag & drop copy column and make a connection from audit to copy column
Select Gender from available input columns and click ok

Drag & drop Character map transformation and make a connection copy column to character map

Double click on character map and check copy of gender available input columns and set.

Operation - lower case

Click ok

Drag & drop OLEDB destination and make a connection from character Map to destination

Double click on destination and provide destination connection manager

Click new to create destination and provide destination connection manager

Click new to create destination table and rename it as audit data.

Click ok twice

**Migrate DTS 2000 package to 2005 dtsx package:**

Open Business Intelligence Development Studio 2005 & 2008

Select project menu and select migrate dts 2000 package option

In package migration wizard, click next

Set, source-select structured storage file

File name: Click Browse and Select any package (.dts) and click open

Click next

Source file system

Click next

Click finish
Finally the migrated (.dtsx) package arrives in the specified project under SSIS packages section

Execute DTS

**Providing Security for SSIS Packages:**

Protection level is in SSIS package that is used to specify how sensitive information is saved with in the package and also whether to encrypt the package or sensitive portions of the package.

Ex: The sensitive information would be the password to the database.

**Steps to configure protection levels in SSIS:**

Open Business Intelligence Development studio

Create OLEDB connection with server authentication and provide user name and password of the server.

Design package

Select package in control flow, right click

Select properties, find the option protection level and select it.

**Variables:**

In SSIS the variables are categorized into two parts.

♣ System Variables,

♣ User Defined Variables.

**System Variables:**

System variables are built in variables and System variables can be accessed throughout the package.
For example:

System :: Creation Name
System :: Package Name
System :: Task ID etc.

Note:

System Defined Variables can be identified system scope resolution operated (::). That means all system variables should start with system:::

**User Defined Variables:**

User Defined Variable can be created by developer and User Defined Variable can have its own name, data type, value and scope as well.

Note:

User Defined Variables can be identified by USER :: Variable Name

1. If we create a variable with respect to a package the scope of that particular variable is complete package.
2. If we create a variable with respect to a container that variable can be used or accessed in the entire container.
3. If we create a variable with respect to a task the scope of the variable is within the specified task only.

**Example:** Package for Excel Source

Open Business Intelligence Development Studio Create

New Package and rename it as Excel Source .dtsx In control flow drag and drop data flow task

Double Click on data flow task to configure it
In Data flow, drag and drop Excel Source

Double Click on Excel Source to configure it

Click New to Create Connection Manager for Excel

Click browse and select excel Alliance details excel file and Click open

Ensure that the first row has column names Check box is checked

Click ok

Select Data Access mode as table or view

Select Sheet 1 from drop down list

Select columns

Click ok

Note:

Prepare the following excel file i.e. already linked or connected to excel source.

<table>
<thead>
<tr>
<th>Source Code</th>
<th>SBAT Type</th>
<th>Partner Type</th>
<th>Funded Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>81818</td>
<td>MSA</td>
<td>Builder</td>
<td>50000</td>
</tr>
<tr>
<td>81540</td>
<td>BI</td>
<td>Realtor</td>
<td>40000</td>
</tr>
<tr>
<td>12345</td>
<td>MAP</td>
<td>Realtor</td>
<td>9000</td>
</tr>
</tbody>
</table>

Find out SRC Row Count

Make a connection from SRC to Row Count

Double Click on Row Count to edit it

In component properties tab, provide the following property

Custom properties

Variable Name – UV src Count
Click refresh and Click ok

Drag and drop derived column Transformation and make a connection from Row Count to Derived Column

Double Click on Derived column to define execution Data provide the following expression. Execution Date – get date ()

Click ok

Drag and drop row count Transformation and rename as RCDest.Count

Make a connection from Derived column to Row Count

Double Click on Row Count and provide the following property Custom properties

Variable name – UNDST

Count Click Refresh

Click ok

Drag and drop OLEDB Destination and make a connection from Row count to Destination Double click on Destination to configure it

Provide Destination Connection Manager if exists

Click New to create Destination table if it is not existing and rename the destination table as Tested-Data

Click ok select Mappings

Click ok

Note:

Define the following variables in control flow write package
<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVSrc Count</td>
<td>Int 32</td>
<td></td>
</tr>
<tr>
<td>UVDst Count</td>
<td>Int 32</td>
<td></td>
</tr>
<tr>
<td>UVSolution Name</td>
<td>String</td>
<td>Morning 8.30 batch</td>
</tr>
<tr>
<td>UVTable Name</td>
<td>String</td>
<td>Tested Data</td>
</tr>
</tbody>
</table>

In control Flow, Drag and Drop Data Flow Task

Rename it as Data Flow Task Test condition

Double click on Data Flow Task

In Data Flow, Drag and Drop OLEDB Source

Double click on OLEDB src to configure it

Provide src Connection Manager if exists and set the following properties

Data access Mode – Select SQL command

SQL command text – Provide the following query to fetch execute date,

Select distinct getdate () as [execution Date] from [Tested – Data]

Drag and Drop Derived column Transformation to derive the following columns using the existing variables

Make a connection from OLEDB src to derived column

Double click on Derived column

Solution Name @ [User::UVSolution Name]

Package Name @ [SYST_]:Package Name

Table Name @ [User::UV Table Name]

Source count @ [User::UV src count]

Destination count @ [User::UV Dst count]

Click ok

Drag and Drop, conditional split Transformation to check the condition
Make a connection from Derived column to conditional split

In Connection split Transformation Editor, provide the following condition

<table>
<thead>
<tr>
<th>Output Name</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>[source count] :: [Destination count]</td>
</tr>
</tbody>
</table>

Rename case 1 as src count is equal to Dst count

Rename conditional split default output as src count is not equal to Dst count

Click ok

Drag and Drop Derived column Transformation and make a connection from conditional split to Derived column

Select src count is equal to Dst count from Input/output editor Rename the Derived column 1 as success status

Double click on Derived column and derive the following expression

<table>
<thead>
<tr>
<th>Derived column Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>“success”</td>
</tr>
</tbody>
</table>

Click ok

Drag and Drop, derived column Transformation and rename it as Failure Status

Make a connection from conditional split to Derived column

Double click on failure status to define the status

<table>
<thead>
<tr>
<th>Derived column Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>“failure”</td>
</tr>
</tbody>
</table>
Drag and Drop Union All Transformation and make a connection from success
status to Union All and also make a connection from failure to Union All
Drag and Drop OLEDB Destination to capture the log information
Make a connection from Union All to Destination
Double click on Destination to configure it
Provide Destination connection manager it exists
Click New to create new destination table and rename it as SSIS_Log
Click ok
Select Mappings
Click ok
Execute package

**Data Viewer:**

Data viewer used to debug the package and also user or developer can monitor the
data which is following from source to other stages(transformations or destinations). In SSIS
Data viewers are categorized into 4 parts.

1) **Grid:**
   
   Select Grid type and also select Grid tab.
   
   Select the columns to be displayed in the Grid format.

2) **Histogram:**
   
   Select histogram type and click histogram tab
   
   Select the column which you want to model with the histogram.
     i) Histogram graph works only with 1 column at a time.
     ii) Histogram graph works only with numeric data.

3) **Scatter Plot (x,y):**
   
   Select scatter plot type and also click select scatter plot (x,y) tab and then select a
column for the x - axis and a column for y - axis.
i) Scatter plot (x,y) graph works only with numeric data.

4) **Column Chart:**

   Select column chart type and click column chart tab.
   Select the column that you want to model in the column chart.

i) This graph only works with 1 column at a time.

**Steps to configure Data viewers:**

- Open Business Intelligence Development Studio
- Create a new package and rename it as Data viewer's .dtsx
- In control flow drag and drop data flow task
- In data flow drag and drop OLEDB Source and configure it
- Drag and drop OLEDB destination and configure it
- Make a connection from OLEDB source to OLEDB destination
- Double click on data flow path
- Select data viewer's option
- Click Add to add any data viewers
- Select grid type
- Select grid tab
- Select the columns which you want to display in grid format and click ok
- Click ok
- Execute package
Note:

Data viewers cannot be implemented on production environment. But data viewers can be implemented only on development environment for debugging the package.

**Loggings:**

Logging is a feature which creates the record that traces the execution of tasks and container within the package.

**Steps to configure Logging:**

1. Open Business Intelligence Development Studio
2. In Control flow, right click and select logging
3. Check container check box
4. In providers and logs tab, Provider type-select SSIS log providers for text file
5. Click add
6. Click newly added log provider check box
7. In Configuration header, select new connection
8. Usage type - select create File option
9. File - Select browse
10. Provide file name and check open
11. Click ok
12. Select details tab,
   - Check events check boxes on which you want to capture log information On Error, On warning, On Query failed etc.
13. Click advanced button
14. Click ok to save the changes done.
**Event Handlers on SSIS:**

It provides the ability to handle any type of event associated with execution of its task and container. The following list of events is more significant and commonly used.

**On Error:**

It is generated as the result of error condition. It falls into the category of most frequently implemented types of event handlers.

**On Warning:**

Similar to on error event, it raised in response to problem.

**On Task Failed:**

It signals the failure of a task and typically follows on error event.

**On Post validate:**

Occurs as soon as the validation process of component is completed.

**On Pre validate:**

Marks the beginning of the component validation stage. The main purpose of validation of potential problems that might execution from completing successfully.

**Steps to Configure Event Handlers:**

1. Open Business Intelligence Development studio
2. Create a new package and rename it as event handler .dtsx
3. In control flow task, drag & drop OLEDB source
4. Configure it (provide connection manager and select any table from the list)
5. Drag & Drop OLEDB destination and make a connection from OLEDB source to connection
6. Configure the destination (provide destination connection manager and create a new table named as Event handlers)
7. Select Event Handler tab and set,
   - Executables-select data flow task event handler option
   - Event handler-select on pre validate option
Drag & drop execute sql task on to event handler section and it will be executed wherever on pre validate event is raised on the selected executable (DFT Event handler).

Double click on Execute SQL task to configure it. Provide connection and write sql statement.

Click ok

Execute package

**Check Points:**

Check points allows a package to restart at the point of failure.

**Steps to configure check points:**

Open Business Intelligence Development Studio.

Create a new package and rename it as checkpoints .dtsx.

In control flow drag and drop 3 execute SQL tasks and configure as mentioned below.

**Execute SQL Task:**

Rename it as Task 1.

Double click on it to edit provide the following information:

**Connection:**

Select New Connection

Click New

Provide Server Name (local host)

Select Adventure works data base

SQL command – Select 1|1

Page 53
Execute SQL Task 1:

- Rename it as task 1
- Double click on it to edit
- Provide the following information

Connection:

- Select New Connection
- Click New
- Provide Server Name (local host)
- Select Adventure Works database
- SQL Command – Select 1|0

Execute SQL Task 2:

- Rename it as task 3
- Double click on it to edit
- Provide the following information

Connection:

- Select New Connection
- Click New
- Provide Server Name (local host)
- Select Adventure Works database
SQL Command – Select 1|10

Make a connection from task 1 to task 2 and task 2 to task 3

Right click on control flow and select properties

Set the following settings,

- Check points
  Check point file name – Click Browser
  Provide check points .XML as a title name
  Click open

  Check point usage – if exists
  Save check points - true

Select Task 1 and set the following properties

  Fail parent on failure – True

  Fail package on failure – True

  Set the above 2 properties for task 2 and task 3

  also Execute package

  You will observe that task 2 is failed

  Select or click F5 to start the debugging

  Double click on Task 2 to edit the SQL query, change the SQL command, Select one by one

  (any valid SQL statement)

  Click ok
Event Handlers in SSIS:

SQL server 2005 Integration Services provide the ability to handle any type of event associated with execution of its task and container (through the ability to configure corresponding handlers). The following list contains more significant and commonly monitored types of events (You might be able to spot some of them in the output window during package execution in Debug mode).

On Errors:

Generated as the result of an error condition. It falls into the category of the most frequently implemented types of event handler. Its purpose can be, for additional information simplifying trouble shooting, or to notify about a problem and need for remediation.

On Warning:

Similar to the On Error event, it is raised in response to a problem (although not as significant in terms of severity).

On Information:

Produces reporting information relating to the outcome of either validation or execution of a task or container (other than warning or error).

On Task Failed:

Signals the failure of a task and typically follows On Error Event.

On Pre Execute:

Marks the beginning of the component validation stage, following the On pre Execute event. The main purpose of validation is detection of potential problems that might prevent execution from completing successfully.

On Post validate:

Occurs as soon as the validation process of the component is completed (following On Pre Validate event).

On Post Execute:

Takes place after an executable component finishes running.
On Variable Value Changed:

Allows you to detect changes to variables. The scope of the variable determines which executable will raise the event. In addition, in order for the event to take place, the variable’s change Event property must be set to true (the default is false).

On progress:

Raised at the point where measurable progress is made by the executable (for example, when running execute SQL Task). This can be evaluated by monitoring the values of the system variables associated with the On Progress event handler, such as progress complete, progress count low, and progress count high.

Steps to configure Event Handler:

Scenario:

Clean up or truncate the destination table before executing or loading data into destination.

Open Business Intelligence Development Studio

Create a new package and rename it as Event Handler. dtsx

In control flow drag and drop data flow task and Rename it as DFT Event Handler.

In Data Flow drag and drop OLEDB source

Double click on OLEDB source to edit it

Provide connection manager if exists

Select [Human Resources] . [Employee] table from the dropdown list

Select columns from left panel

Click ok

Drag and drop OLEDB destination and make a connection from OLEDB source to destination.

Double click on OLEDB destination
Provide destination connection manager

Click new to create a destination table and Rename OLEDB destination as Employee details

Click ok twice

Go to Event handler tab

Drag and Drop Execute SQL Task [on pre validate Event handler for package Executable]

Double click on Executable SQL task provide connection if exists

Provide click on Execute SQL command, to clean up the data from destination table

Truncate Table Employee details

Click ok

Execute package

**Different ways to execute SSIS package:**

1. **Execute SSIS package using BIDS:**

   During development phase of the project developers can test the SSIS Package execution by running the package from BIDS.

   Open BIDS, Configure a new package

   Press Alt+Ctrl+L for solution explorer

   Select the package

   Right click and select the execute package option

2. **Execute SSIS package using DTEXEC.EXE command line utility:**

   Using the DTEXEC command line utility can execute an SSIS package that is stored in

   File system or SQL Server.

   **Example:** DTEXEC.EXE" C:\Packages\Checkpoints. dtsx"
3. Execute SSIS using DTEXEC UI.EXE:

Using the execute package utility graphical interface can execute an SSIS package that is stored in file system or SQL Server.

In command line type DTEXEC UI.EXE and press enter which open up will execute Package utility editor.

In Execute package utility editor select General tab,

Package Source-File system

Package - click browse (select any package from the list and click open)

Click execute to execute the linked/embedded package

4. The Execute package utility is also used when you execute the SSIS Package from integration service mode in SQL Server management studio.

Open SQL Server Management Studio

Connect to Integration Services

Expand Stored Procedures

Expand File Systems

Select File Systems and right click, Select import package editor.

Package Location-File System

Package-Click source (select any package from the list and click open)

Package Name-Place Cursor
Click Ok

Select imported package

Right click and select run package

In execute package utility editor click execute.

5. Execute SSIS package using SQL Server Agent (Using Jobs):

Open SQL Server Management Studio, Connect to Database Engine & Select SQL Server Agent [Note: Ensure that SQL server Agent Service is start mode]. Select Jobs, Right Click and select new job

Provide Job name as “Load for each loop container”

Select Steps page & Click new
Step Name: Load for each loop container
Type: SQL Server Integration Service Package

In General Tab, Package Source: File System
Package: Click Browse
Select any package from list of package

Click open

Click ok

Select schedule package & Click new
Provide the Job name as “Load for each loop container” & Schedule Type: Recurring

Set the Frequency (When to start the execution of the specified package

Click Ok twice

DBA Tasks in SSIS 2005

Data base Administrators can create database maintenance plans either by using Maintenance plan Wizard or by using SSIS Designer. By using maintenance plan wizard a very basic Maintenance plan can be created for all the system and user Data bases. However to create an enhanced work flow it is advised to create Maintenance plan using SSIS designer.
**Backup Database Task:**

Different types of Databases backups to be performed; Such as Differential or Transactional log Backups based on the Recovery Model of the system or user Databases. To learn more about the different Recovery Models in SQL Server refers to the article titled “Database Recovery Models in SQL Server”.

**Check Database Integrity Tasks:**

It can used to check the allocation and structural Integrity of all the user and system tasks within a database. This task also has an option to check the allocation and structural integrity of all the Indexes available within a database. This task internally Executes DBCC CHECK DB Statements.

**Execute SQL Server Agent Job Tasks:**

Execute SQL Server Agent Job Tasks can be used to run SQL Server Agent Jobs that are created on the SQL Server Instance.

**Execute T_SQL Statement Task:**

Execute T_SQL Statement Task can be used to execute to transact SQL Query against Databases. This task is only available when creating maintenance plans using SSIS Designer.

**Transfer Database Task:**

The Transfer Database Task is used to move a database to another SQL Server Instance or create a copy on the same instance (with different database name). This task works in two modes as shown below:

- **Offline:** In this mode, the source database is detached from the source server after putting it in single network location. on the destination server the copies are taken from the network location to the destination server and then finally both databases are attached on the source and destination servers. This mode is transfer, but a disadvantage with mode is that the source database will not available during copy and move operation also, the person executing the package with this mode must be system admin and both source and destination instances.

- **Online:** In this mode, the task uses SMO to transfer the database objects to the destination server. In this mode, the database is online during the copy and move operations, but it will take longer as it has to copy each object from the database.
individually. Someone executing the package with this mode must be either system admin or Database owner of the specified databases.

**Transfer Database Task:**

Drag the Transfer Database Task “from the toolbox (which is normally on the left side) to the control flow pane. Right Click on the task and select Edit… as shown below:

In the “Transfer Database Task Editor”, select databases on the left and now you are ready to configure this task. Source connection is the property to specify the connection for the source SQL Server Instance, if you have already created a connection manager then you can reuse it here or can create a new one as shown below:

This will also need to be done to configure the destination connection property as well. Next you need to specify the values.

* **Connections:**
  - Source connection - The source Instance.
  - Destination Connection - The destination instance.

* **Destination Database:**
  - Destination Database Name - Name of the new database Destination
  - Database Files - Name and location of the database files.
  - Destination Overwrite - If the database already exists on the destination server it will give you an error, so if you want to overwrite the destination database you can set this property to true.

* **Source Database:**
  - Action - Whether you want to copy or move a database.
  - Method - whether you want the copy and move operation to be offline (in this case you also need to provide the network share name which will be used to transmit the database files)
  - Source Database Name - Name of the source database.
  - Source Database Files - Name and location of the database files.
  - Reattach source database – Is another property which you can set to true to reattach the source database after the copy operation.
Once you are done with all the configurations you can hit F5 or Click on the play icon to execute the package. Your task will turn yellow during the execution and then either red or green depending on the execution outcome. You can go to the progress tab to see error messages if the execution fails. Although failure is rare, it is possible if your source database is smaller than the size of the model database on the destination server.

**Transfer SQL Server Object Task:**

The Transfer SQL Server Object Task is used to transfer one or more SQL Server Objects to a different database either on the same or another SQL Server Instance. This allows you to select different types of objects you want to transfer. You can select tables, views, stored procedure, user defined functions etc. Not only this, you can select a combination of these types of objects to transfer and even select particular objects of a particular object type.

Create another package in the same project and drag a "Transfer SQL Server Objects tasks" to the control flow. Right click on the task and then select edit. In the “Transfer SQL Server Object Task Editor” click on objects to set the different properties for this tasks as shown below:

Similar to the way you configure the source connection and destination connection for the Transfer Database Task, you need to configure these properties for this task as well. Source database is the name of the database from where you are copying the objects and destination database is the name of the database to which you are copying the objects to.

* **Connection:**
  - Source connection-the source instance
  - Source Database- Name of the source data base
  - Destination Connection- The Destination instance
  - Destination Data Base-name of the new Data base

* **Destination**
  - Drop objects First - Drop selected objects on the target before copy
  - Include extended properties – While copying operations also include extended properties of SQL objects being copied
  - Copy data – While copying tables, transfer the data of the selected tables as well.
  - Existing Data – Whether to append r replace data
  - Copy Schema – Copy the schema of the objects being copied.
Use Collation – Make sure collation of columns are approximately set on copied tables
Include Dependent Objects – Include all the objects in copy operation which are dependent on selected objects.

- **Destination Copy Objects:**
  Copy all objects – Do you want to copy all objects from the source database, if set to false, next property objects to copy will get enabled.

Objects to Copy – With this property you select types of the objects you want to copy. You can select all objects of one or more types or select particular version, type of objects selection will vary.

- **Security:**
  Copy Database Users – Whether to include users
  Copy Database Roles – Whether to include roles
  Copy SQL Server Logins – Whether to include logins
  Copy Object Level Permissions – Whether to include object level permissions

- **Table options:**
  Copy Indexes – Whether to include indexes
  Copy Triggers – Whether to include triggers
  Copy Full Text Indexes – Whether to include full text indexes
  Copy All DRI Objects – Whether to include referential integrity objects
  Copy primary Keys – Whether to include primary keys
  Copy Foreign Keys – Whether to include foreign keys
  Generate Script In Unique – Whether to create script in Unicode or not

**Transfer Jobs Task:**

Transfer Jobs Task is used to transfer SQL Server agent jobs from one SQL Server instance to another. This task gives you an option to copy all the jobs or selected jobs from source server to the destination server.

Open BIDS. Drag a Transfer jobs Task from the Tool box to the control flow pane as shown below. Right click on this task and click on Edit.
**Steps to Configure Transfer Jobs Task:**

- **Connections:**
  
  Source connections - Specify the connection for the source SQL Server instance, if you have already created a connection then you can reuse it here or can create a new one.

  Destination Connection – Specify the connection for the destination SQL Server instance

- **Jobs:**

  **Transfer All Jobs:** If this is set to true it will transfer all jobs. If this is set to false you can select specific jobs you want to transfer.

  **Jobs List:** This will be enabled if Transfer All Jobs is set to false. Then you can select specific Jobs to transfer. See the image below where I am only selecting two jobs to transfer to the destination.

- **Options:**

  **If Object Exists** – If the Jobs already exist on the destination then you have three choices, first Fail Task execution, Second overwrite the destination job and third skip the existing job and continue with others.

  Enable Jobs At Destination – After the transfer you can also specify whether to enable or disable the job.

**Transfer Logins Task**

Transfer Logins Task is used to transfer either all logins (except sa) or selected logins or all logins of selected databases from the source to the destination SQL Server instance. After the transfer, all the transferred SQL logins are assigned random passwords and SQL logins are disabled. The DBA needs to change the password and enable the SQL login before it can be used on the destination.

- Create another package in the current project and drag a Transfer Logins Task from the Tool Box to the Control flow. Right click on the task and select Edit to configure the tasks properties as shown below.

**Steps To Configure Transfer Logins Task**

- **Connections:**

  Source Connection – Specify the connection for the source SQL Server instance

  Destination Connection – Specify the connection for the destination SQL Server instance
- **Logins:**
  
  **Login Transfer:** You have three options for this:
  
  * All logins – This will transfer all logins from the source
  
  * Selected Logins – This allows you to select specific logins
  
  * All logins From Selected Databases – This allows you to transfer all logins associated with one or more databases as shown in the image below.

- **Logins List** – This will allow you to select specific login if you select. Selected logins for login transfer.

- **Database list** – This will allow you to select the databases if you select All Logins From selected databases for login to transfer.

**Options:**

- **If Object Exists** – If the logins already exists on the destination you have three choices:
  
  First Fail task execution, Second overwrite the destination logins and Third skip the existing login and continue with the others.

- **Copy Sides** – If you set it true then security identifiers (SIDS) associated with logins are also copied to the destination.
ARCHITECTURE OF SSRS

Report Builder Adhoc
Reporting Tool

Report Designer (BIDS)

Report Manager

Report [http://localhost]
Report Server

Data Source

Rendering Formats

- EXCEL
- PDF
- CSV Etc

Data Set 1
Data Set 2
Data Set 3
**Data Source:** Logical connection between application (SSRS) and database.

**Report Servers:** Report server is the core engine that drives Reporting services. The URL for report server is http://localhost/report server.

**Report Manager:** Report Manager is a web based administrative interface for reporting services.

**Report Designer (BIDS):** is a developer tool for building simple and complex reports using existing report item template.

**Example:**
1. Table
2. Columns
3. Rectangle
4. List
5. Image
6. Matrix, etc.

**Report Builder:** is a simplified end user or adhoc reporting tool for building simple reports.

**Table Report Item:**

It is used to display data in grid format that has a fixed number of columns and variable number of rows.

**Steps to configure Table Report Item:**

1. Open Business Intelligence Development Studio
2. Select File Menu
3. Select New
4. Select Project
5. Under Business Intelligence projects select Report server project option to build simple and complex reports on development environment.
6. Change the location and name of the project
Click OK

Press Alt+Ctrl+L for solution explorer

Select shared data source and right click. Select add new data source option from context menu or right click popup menu.

Set,

In share Data source editor,
- Data source – Rename it as Dsource product details
- Type – Microsoft SQL server
- Connection string – Click Edit

Provide server name (local host) (or).(or)host name or IP Address

Select Adventure works

Click OK

Click OK

In Solution Explorer, select reports, right click

Select Add and also select New Item

In Data Tab, select New data set option and set,
- Data set name – Dset product information
- Data source – Dsource product details from drop down list
- Command Type – Text (i.e., SQL Script)
Click OK

Select Query Designer option and Add Tables icon on the rare right and select Tables (or)

Write the Query in SSMS and copy and paste in Query Designer.

Select, product category. Product category ID,

Product category name. Name as category name, Product. Product ID,

Product. Standard cost,

Product. size, product.

sell start date, product.

reorder point

From product category Inner Join product

Sub category on product category . product

Category ID = product sub category . product

Category ID.

Inner Join product on product . product

Product sub category . product = sub category ID

Sub category ID

Inner Join product cost history ON

Product . product = product cost history . product ID

Provide the above query to the created data set and click Refresh.

Click OK

Select View Menu and select Dataset option or (Alt+Ctrl+D)

Select Layout tab to design the report

Select tool box (or) (Alt+Ctrl+X)
Drag and drop table report item on body of the report

Drag and drop Category Name field on to Detail section of the table report [Building (or) Mapping the field (or) Columns from dataset to Report item]

Map other field as well

In table Report item select 1 Row and 2 Row

Press F4 and set

Borderstyle – solid

Select 1Row [Header] and press F4 for properties and set,

Background color – Maroon

Text align – center

Font size – 9pt

Color – white

Parameterized Report:

In parameterized report, the report will be populated data based on the user input.

Steps To Configure Parameterized Report:

In previous Report example – In data tab, add the following where clause to set the parameters

Where,

(production . product category . product category ID = @product category ID)

In Data tab select New Dataset and rename it as Dset product category ID name Data source – Dsrc product details

Common type – Text

Query string – Provide the following SQL command to populate all the product category ID, name as well.
Select,
Product category ID,
Name as category name
From,
Product . product category
Select Report Menu and select Report parameters
Select product category ID parameter and set,
Prompt – select category

Check from Query Radio Button (to populate all IDs from the selected Data set)
Data set – Dset product category ID Name
Value field – Product category ID
Label field – Category name

Click OK

Multi Valued Parameterized Report:

In Multi Valued Parameterized Report, the user can select multiple values from the specified parameter.

Steps to configure Multivalued Report:

From the previous report example, select Data tab

Select Dset product information dataset

Append the following where clause

WHERE,
[production . product category . product category ID IN (product category ID)] Refresh

Select Report Menu and Report parameter option

Check multivalue label check box
Click OK

Go to preview tab, select multiple values from the parameter drop down list and click

Preview Reports

1) **Expressions in SSRS:** Expression to format the data using format function (or) format codes.

   Select standard cost text box
   
   Right click and select expression,

   = Format [Fields!standard cost.value.NO(zero)]

2) **Expression to check null records in the specified field**

   If any field value is Null then place “- -“, otherwise display the actual value of the fields.

   = IIF (Is nothing (fields size.value), “- -“,Fields!size.value)

3) **Expression to align the text:** If the text is Null then place (or) display the “- -“ in center else left justified. Select size text box, press F4 from properties and set text align - IIF [is nothing (Fields!size.value), “center”, “left”]

4) **Expression to display the values in various colors based on the conditions**

   Select size text box
   
   Press F4 for properties
   
   Set,

   Color ___

   = IIF [Fields! size value < 39, “Red”, IIF (Fields! size.value > 30, “Blue”, “Green”)]
5) **Expression to handle negative values**

For example,

\[(\text{Blue} - 79) \Rightarrow (79) \text{ in Red color}\]

In table report item, add one column and rename the header as Totals and provide the following expression.

\[= \text{IIF} \ [(\text{Fields! standard cost. value} - \text{Fields! Reorder point. value}) < 0, \ \\
\text{Replace} \ [(\text{Fields! standard cost. value} - \text{Fields! Reorder point. value}), "-", "(" & ")", \ \\
(\text{Fields! standard cost. value} - \text{Fields! Reorder point. value})]\]

Select Totals text box (in details section)

\[\downarrow\]

Press F4 properties

\[\downarrow\]

Set, \(\text{color} = \text{IIF} \ [\text{Fields! standard cost. value} - \text{Fields! Reorder point. value} < 0, \ \\
"Red", "Black"]\]

6) **Expression to display Report with alternative Row color**

\[= \text{IIF} \ [\text{Row number (nothing) mod2, "Silver", "Transparent"}]\]

**Chart:**

Chart Report item is used to display the data graphically as bass pie, column, area and other chart types. In our previous Report example, Drag and drop chart Report item after (or) below the table Report item.

\[\downarrow\]

Right click on chart and select properties from Context Menu.

In General tab,

\[\text{Title} = \text{"product details for the month"} \text{ and month name [month}}\)

(\text{Fields! sell start date. value})]\]

Select Data tab

\[\downarrow\]

Data set name – select Dset

Product information from drop down list value_ click Add to add column (or) fields that would be plotted on Y-axis.
In Edit chart value,

Series Label – standard cost

Value = Fields! Standard cost. value

Click OK

Values – click Add to add column (or) fields that would be plotted on Y-axis

In Edit chart value

Series label – size

Value – Fields! Size. value

Click Ok

Category group – click Add to add column (or) Fields that would be plotted on X-axis

Select expression and provide the expression to display the records like JAN-01.

Format,

= Left (Month Name ( ) (Month[Fields! Sell start date.value]).3) & “_” &
Right (year(Fields! Sell start date.value).2)

Click OK twice

Select X-axis tab and set,

Title_sell standard date

Check Major Grid Line and specify the scale (or) interval [0.5]

Title align - center

Major tick lines - cross
Minor tick lines - cross

Check side margins, check box to display the chart items in center of the plotter area.

In Y-axis,

Title - size

Check Major Grid Line and specify the scale (or) interval

0.5

Title align - center

Major tick lines - cross

Minor tick lines - cross

Check side margins check box to display the chart items in center of the plotter area.

In Legend Tab,

Check Show Legend (by default)

Layout - Table

Check Legend style and set,

Font size – 8pz

Click OK

How to set or fill custom colors to the chart? (Series)

In Data tab,

Values – select standard cost and click Edit

Select Appearance Tab

Click series style
Select Fill tab

↓

Click Browse to select a color from the list

↓

Select any color and click OK

How to set point labels in chart (Display the exact value on plotted area)?

Values – select standard cost and click Edit

↓

Select point labels tab

↓

Check show point labels check point

↓

Data label _ = “&” and Fields! Standard cost.value

↓

Click OK

Rectangle:

It is a container Report item for other report items.

Steps to configure rectangle report item:

In our previous report example drag and drop rectangle

↓

Select rectangle and right click, properties,

In General Tab, rename it as Rect Table.

↓

Click OK

↓

Drag and drop the designed table report item on the rect table rectangle

↓

Drag and drop another rectangle and rename it as Rect chart in rectangle properties editor

Page 77
Drag and drop the configured chart onto Rect chart rectangle

**Scenario:**
Display Table report item in 1 page and matrix report item in another page when you report it to excel or pdf etc.

**Steps:**
- Select Rect Table (Rectangle 1)
- Right click and select properties
- Set, Page breaks – check Insert before Rectangle chart box (since to maintain the page breaks after Rect tab Rectangle)
- Select preview Tab
- Selects bikes as a parameter and click View Report
- In export option, select excel, save it to desktop
- Open the exported excel file. You will observe that sheet 1 contains Table Report and sheet 2 contains Chart Report.

**Image:**
Image Report item is used to display a bit mapped images such as photos or logos in a report. The selected or configured image will be embedded to the report or to the selected project.

**Scenario:** Display company logo on each and every report page.
- Select Report Menu and select page header and also select page footer.
- Press Alt+Ctrl+X
Drag and drop image report item onto left corner of the page header

In welcome to the image wizard, click Next

Select Emberd (the selected image is stored as a part of the Report)

Click Next

Click New Image

Select any jpg file and click Open

Click Next

Click Finish

Resize it based on the requirement.

**Text Box:**

It is used to display labels or fields or values calculated from expressions. From our previous report example, drag and drop text box report item onto page footer.

Select Text box and right click and select expression option. Compose the following expression.

\[ = \text{“page”} \& \text{Globals! Page number} \& \text{“of”} \& \text{Globals! Total pages} \]

\[ = \text{iif (parameters! Product category ID.value = 1, “Bikes”, “components”)} \& \text{“Details for the Month ____”} \& \text{month name} \]

\[ (\text{Month (Fields! Sell start date.value)}) \]

**List Report Item:**

It is used to display a set of report item that are repeated for each group (or)

Row of data.

**Steps to configure List Report Item:**

In our previous Report example, remove the existing where clause and add the below to implement multi value report,
where,

(production.product category.product category ID IN (@product category ID) Select

Report Menu and select Report parameter option, set check multi value check box

↓

Click OK

↓

drag and drop list report item and place table report item and chart report item onto list

report item

↓

Select List right click and select properties

↓

In list properties editor, select General tab,

Data set name – Dset product information

↓

Click Edit Details Group

↓

Select = Fields! Product category ID.value in group on selection

↓

Check page break at end check box (to maintain the page breaks after each group)

↓

Click OK twice

↓

Go to preview tab, select Accessories and bikes and click View report

↓

Export the report to excel format. Save it on desktop.

↓

Open the exported excel. You would observe that sheet 1 contain accessories details

and sheet 2 contains bikes data

Matrix:

Matrix report item displays the aggregated data in a grid format that has

variable number of rows and variable number of columns as well.
Steps to configure Matrix Report item:

In our previous report example, drag and drop matrix report item Go to the properties of matrix

In Matrix properties editor, Dataset name – Select Dset product information Click OK

Press “Alt+Ctrl+D” for dataset.

Drag and drop category name on a row text box Drag and drop sub category name on column text box

Drag and drop standard cost on data text box

Select data text box

Right click and select add column. To add a new column under sub category base column Drag and drop color field onto newly added column.

Set, border style as solid and set the font size Go to previous tab

Drill Down Reports:

Select configured table report item Go to table properties
Select groups tab, click add to add a new group list

In grouping and sorting properties editor,
select = Fields! Category name.value

From the drop down list, make sure that the include group footer check boxes are checked

Click OK

Click Details group

In details grouping editor, select
= Fields! Category name.value

In details grouping editor, select visibility tab and set,
Initial visibility – check hidden radio button

Check visibility can be tagged by another report item check box

Report item – select appropriate text box name

Click OK

Click OK

How to display Group totals or sub totals?
From the above table report example add below expression at group footer under standard cost header.

= sum (Fields! Standard cost.value)
How to display Grand totals of the report?

From the above table report item, provide the following expression to display grand total of the entire report. In report footer area, under standard cost,

\[= \text{Sum} \left( \text{Fields! Standard cost value} \right)\]

How to create Template Reports in SSRS?

Design basic report template [Add a logo on header and display page numbers in page footer.

\[\downarrow\]

Rename it as Corporate IT Template.RDL

\[\downarrow\]

Go to project path

\[\downarrow\]

Copy Corporate IT Template.RDL and paste it in the specified (or) below mentioned path.

\[C:\Program Files|\]

Microsoft Visual Studio 8|\]

Common 7|IDE|\]

| Private Assemblies|\]

Project Items|Report project

Linked Reports:

Linked report is the report with the settings and properties of its own. A linked report can have its own name, description, location, parameter properties, report execution properties etc.

Note: A linked report must use data source properties and layout of its base report.

Steps to configure Linked Report:

Open report manager (http://localhost/reports)

\[\downarrow\]

Select a report to which you would like to create a linked report

\[\downarrow\]
Select properties of the report

↓

In General tab, click create linked report

↓

Provide linked report name, description and change the location etc.

↓

Click OK

Note:

In properties of the linked report, data source, edit and update option are not available.

Linked report is always points to data source and lay out of its base or its parent report.

Sub Report:

Sub report item is used to display an embedded report within the current report.

Steps to configure sub report:

Open Business Intelligence Development Studio

↓

Create a new report and rename it as Master report.rdl

↓

Create a new dataset and provide the following query, select *

↓

From product category INNER JOIN product sub category ON product
category.product category ID = product sub category.product category ID

WHERE,

(product category.product category ID = @product category ID)

↓

Create dataset 2 and provide the following query

Select product category ID, name as category name from production,production
category

↓

Select Report menu and select report parameters
Set,

Select product category ID parameter,

Name – product category ID  
Data type – string  
Prompt – select category

Select “From Query” radio button to populate all the values in a parameter drop down list and set,

Dataset – Dataset 2 Value  
field – product category ID  
Label field – category name

Click OK

In layout tab, drag and drop sub report item

Select sub report, right click and select properties

In General tab,

Sub report – select any report from the drop down list (the ddl contains only the reports from the current projects)

In parameter tab,

Parameter Name – product category ID  
Parameter Value = parameters! Product category ID value

Click OK

Select preview tab

Pass all the parameters which are in the embedded report to sub report whatever the value it may be, either original value or dummy value. Dummy value must be like = “Bikes”.
Globals:

Global parameters or fields applies to a set of variables built into reporting services, details while configuring report service in SQL server configuration tools.

Path – provide path in uniform naming convention format.

Credentials – provide user name and password (given by client)

Render format – select pdf or excel

Click select schedule to specify time to execute the report

↓

Select once radio button and specify the time when to execute the report

↓

Click OK

↓

Select parameter values

↓

Click OK

The report server uses two databases to store the data after successful completion of report configuration.

1. Report server
2. Report server Temp DB

1) Report Server: is the main database to store reporting services data like report definition (rdl file details) report model, data sources, schedule information, security details.

2) Report Server Temp DB: Stores temporary reporting services data like user sessions, cache information.

Expression to show and hide a column based on parameter:

Select any table column

↓

Press F4 for properties of the selected table column and set, Expand visibility

Hidden – select expression

↓

= IIF (parameters! Product category ID.value = 1, false, true)
How to provide role based security on Report Manager:

Open Report Manager

↓

Select any report to which you want to provide security with respect to a role

↓

Select properties option

↓

Select security page

↓

Click edit item

↓

security Click Ok

↓

Click new role assignment

↓

Provide username and also select one or more roles to assign to the specific group or user

↓

Click OK

Report Data Model:

Open Business Intelligence Development Studio

↓

Select File Menu

↓

New and select project

↓

Select Business Intelligence project option

↓

Select report model project under template folder

↓

Change the name and rename it as product details
Click OK

In solution explorer, select data source Right
click and select Add new data source In
welcome to data source wizard, click Next
Select the data source if exists
Click Next, rename the data source as product
details Click Finish
Select data source view

Right click and select Add new data source
view Click Next
Select product details data source and click Next
Select production. product sub category table and click Add related
tables Click Next
Rename the data source view as product
details Click Finish
Select report models in solution explorer

Right click and select Add new report model

Click Next

Select product details.DSV

Click NEXT thrice

Rename it as product details

Click Run

Click Finish

At this point, the report model is available but it is not yet available to the end user. Once we publish or deploy a report model to the report server, so that it will be available at the report builder.

**Report Builder:**

Report Builder is a windows applications. Report builder will load with the new report dialogue box and shows all the available report models.

**Publish a report from the Report Builder:**

Open report builder from report manager

Double click on the available data model (product Details)

Design the report by drag and drop or double click on the fields

Select File Menu and select save

The default save location is the Home folder on the reporting services server that hosts the report model.
Click ok

Detractive sort – sort the report data dynamically
In our previous report example, select category Name header text box. Right click and select properties

In properties Editor, select Interactive sort tab
Check Add on interactive sort action to this text box check box and set
Sort expression - = fields! category Name.value

Click ok

Drill through Reports (or) Hyperlink Reports (or) Navigational Reports:

Jump to Report:

In our previous report example select subcategory details selection text box
Right click and select properties
In text box properties editor, select navigation tab and set,
Check Jump to Report radio button
Select any report from the ddl
Click ok

Jump to Url:

In our previous report example, select Name detail text box.
Right click and select navigation tab
Check Jump to Url radio button
Click fx and provide the following expression= “Java script: Void window. Open (URL of child report from report Manager)”
**Report parameter properties:**

**Hidden:**

By using hidden properties we can hide the parameter on the parameter input area of the published Report.

*For example:* While implementing sub report hides the parameter in the embedded Report.

**Multivalued:**

If you want to display multiple elections then multivalued parameter is used. In order to implement multivalued parameter Report, an operator must be used in the dataset.

**Internal:**

If you set a parameter to internal it is not exposed in any way except in the Report definition. Internal parameter is a parameter that can’t be changed at Runtime. A consumer of a published report will never see this as a parameter.

**Allow Null values:**

It indicates that the parameter can have null values.

**Allow Blank value:**

If we want to allow an empty string as a valid value then we have to set this parameter.[Allow Blank value doesn’t support integer data types].

**Cascading Reports:**

In this Reports one parameter is depending on another parameter.

*For example:* Country parameter and state parameter. If the user selects any country from the country parameter then state parameter can be enable and displays (or) populate corresponding states.

**Steps to configure cascading Reports:**

Open Business Intelligence Development Studio and create a new Report item and Rename it as cascading. rdl.

In data tab create New dataset and Rename it as category Details.

Provide the following SQL Query:

Select product category ID, Name as category Name from production. product category.

Click ok

Select New Data set and set Name: Sub category.
Data source: Dsrc product category
Command type: text
Query string:
Select product sub category ID, Name as sub category Name.
From production. product category where product category ID= @ product category Id.

Click ok

Select New Dataset and set
Name : product_details
Data source : Dsrc product category
Command type : Text
Query string : Select * from production.product

Where
Product sub category Id = @ product sub category Id.
Select Report menu and select ‘Report parameters’

Select product category Id parameter and set in available values; check from Query radio button and
Data source : category details
Value field : product category ID
Label field : category Name

Select product sub category ID parameter and set in available values; Check from Query radio button and
Data source : sub category Value
field : product sub category ID
Label field : sub category Name

Click ok

In layout tab design the report
Go to preview tab and Run the Report.
**SSAS**

*(SQL Server Analysis Services)*

The main usage of SSAS is to build OLAPS/online analytical processing. The basic idea to create OLAP is to trade of increased storage space for speed of Querying. OLAP does this by pre – calculating and storing aggregates when you identify the data that you want to store in an OLAP Database, analysis services analyze it in advance.

The basic concepts of OLAP

1) **Cube:**

   The basic unit of storage and analysis in analysis services is the cube.

   A cube is a collection of data that is been aggregated to allow Queries to return data quickly.

   Cubes are ordered into Dimensions and measures comes from fact table.

**Dimension Table:**

It contains hierarchical data by which like to summarize.

For example:

An order table that you might group by year, month, week and day of receipt

**Dimension:**

Each cube has one (or) more dimensions. Each dimension based on one (or) more dimension table.

**Fact table:**

It contains basic information that you want to summarize

**Note:**

Any table that you have used with SUM, Average functions in a total Query is a good bet to be a fact table.

**Measure:**

Each cube will contain one (or) more measures. Measure is a key value to analyze (or) to evaluate the performance of organization.

**Schema:**

There are two types of schemas in OLAP. 1. Star schema.

2. Snowflake schema.
In star schema, every dimension table related directly to fact table.
In snowflake schema some dimension tables are related indirectly to the fact table.

**Steps to Configure (or) Create a Data Cube:**

To build a new data cube using business intelligence development studio. We need to perform the following steps:
1) Create a new analysis service project
2) Define a Data Source
3) Define a Data Source View
4) Invoke a cube Wizard

**Create a new analysis service project:**

1. Open Business Intelligence Development Studio
2. Select File Menu
3. New
4. Project
5. Select project Analysis Services
6. Change project Name and location
7. Click ok

**Define a Data Source:**

1. In solution explorer select data source
2. Right click and select ‘New Data Source’
3. Welcome to the Data Source Wizard
4. Click next
5. Click new to create a new connection manager and set,
   Server name – local host
   Data base name – Adventure works DW
6. Click ok
7. Click next
Check default radio button
Click next
Data Source Name – product details
Click Finish

Define a Data source View:

In solution explorer select Data source view
Right click and select New Data source view
In welcome to the Data source view wizard
Click next
Select the available Relational Data source [product details]
Click Next
Fact reseller’s sales table from available objects list
Click add related tables to include all the tables which are referenced to
fact reseller sales table
Click next
Set, Name – fact resellers view
Click finish

Cube:

In solution explorer select cube
Right click and select new cube
In welcome to the cube wizard, click next
In select build method page, check . build the cube using a data source radio button and
also make sure that auto build check box is checked to create attributes and hierarchy’s
from the specified dimension table
Click next
In select Data source view page, select the available Data source view named as product information [select the data source view, that will provide the data for the cube]

Click next
Click next
Click next
Click next
Click next
Click next
Click next
Click next
Click next
Click finish

In identify facts and dimension tables page select a dimension table from the dropdown list
Click next

In select time period page select any column [time key] to create hierarchy’s on time dimension
Click next
Click next
Click next
Click next
Click next
Click next
Click finish

**Process the cube:**

Select project [In solution explorer]
Right click and select process option
Click yes
In process Data base wizard, click run
Click close
Open SQL server management studio (SSMS)
Click connect and set,
Server type – Analysis Server
Server Name – Local host
Click connect
Exploring the Data Cube:

In BIDS create a cube and process (or) deploy it to server. BIDS includes a built-in cube browser that lets you explore the data in any cube that has been deployed (or) processed. To open the cube browser, in solution explorer double click on cube.

[Product information. Cube]

Select Browse tab

Drag and drop the column on column (or) Rows (or) Filter

Introduction to MDX:

Multi Dimensional expression (MDX) is a language created to allow the users to work with multi-dimensional data in Microsoft SQL Server Analysis Services. It works as a query language with OLAP (online Analytical Processing) cubes. We can also use it to create calculated members or use the various functions provided which can be used to query data.

Differences in SQL in MDX:

If you are a little familiar with SQL (structured Query Language), you may feel there is some similarity with MDX in some aspects. The SELECT clause, FROM clause and WHILE clause are similar with which a cube can be queried it provides different functions. With MDX we can even create, modify and delete cubes if required.

Syntax of MDX Query (Multi Dimension Expression):

WITH

MEMBER exp 1 and exp 2

Select

{ (set operator)
  < axis – Specification 1>,
  < axis – Specification 2>,
} ON columns

<axis specification> on Row

FROM [cube Name]

Where [slicer]
**Axis Specification:**

A collection of members from different dimensions organized as a set of tuples [columns]

**Members:**

Select dimension attribute that are included in output cube.

**Slicer:**

It is a filter that selects cells in output cube.

**Set operator [{}]:**

Set operator is used to display a collection of member on single axis. In MDX set operator is {} (or) Generate ()

**Note:** Generate MDX function is a set of members.

**Functions in MDX:**

**Order function:**

It is used to sort the data either in ascending or descending order.

**Top count:**

It is used to display a specified number of rows from the top.

*Ex:* Top count ([dim product].[EnglishproductName].allmembers,5) on rows

**Bottom Count:**

It is used to display a specified no. of records from the bottom

*Ex:* Bottom Count ([Dim product].[EnglishproductName].allmembers,5)

**Comma operator:**

It is used to construct a set by enumerating tuples.

*Ex:* select {[measures].[Order Quantity],
[Measures].[Sales Amount],
[Measures].[Total amount],
} on columns

(:) **Colon Operator:** It is used to specify a range with discrete or continuous data.

*Ex:

Where [Fact Internet Sales-Due date].[Time Key] . & [1]:[fact Internet sales Due date].[Time key],&[15]
**Dot operator:** It is used to return a set of all the members from the specified dimension.

*For Ex:* [Dim Product].[English Product name].all members on row.

**WITH Operators:** It is used to specify calculated measures.

*For ex:*

Member [Measures].[total amount] as [Measures] . [Order Quantity]* [Measures.[order Quantity]]

**NON EMPTY:**

It is used to remove null records either from column or Rows.

*For Ex: Select non empty

{[measure].[order Quantity]} on columns, Non empty

[Dim Product].[English Product Name].all members on rows.

**Comments:**

-- =>

|**|= for multi

**MDX notation for schema:**

```
[Dim time]. [Fiscal]. [2011].[Q1].[April]
[Dim Table]-Dimension
[Dim time]. [Fiscal]-Hierarchy
[Dim time]. [Fiscal]. [2011].[Q1]-Level
[Dim time]. [Fiscal]. [2011].[Q1].[April]-Member
Select non empty
{[Measure]. [Order quantity]. [Sales amount]}
} ON Columns
Non empty
[Dim Product].[English Product Name]. all members
On rows
```

From [Product information] the above MDX Query returns all the English product names on Rows and its corresponding order quantity and sales amount displayed on Rows from the specified cube called product information and it Return non empty (NOT NULL) records

**WITH**
MEMBER [Measures].[Total amount] as [measures].[Sales amount] * [measures].[order quantity]

Format - String = "$##, ###.##"

Select non empty

{(measures).[order quantity],[sales amount],[total amount]} ON Columns,

Non empty

[Dim product].[English product Name]. all members on rows from (product informations) The above MDX query New measures is calculated [measures].[total amount] and its formatted as with two decimal digits.

Select non empty

{(measures).[order quantity],[measures].[sales amount]} on columns,

Non empty

[Dim product].[English product Name]. all members on rows from [product informations]

Where [each interest sales - Due date].[Time key].&[1]:[fact interest sales - Due date].[time key].&[15]

In the above MDX example

Returns all the English product names and its corresponding order quantity and sales amount based on the where clause [time key start from 1 to 15th]

Format – String:

It is used to format the measures in WITH operator [format the calculated measures]

For ex:

Format – String = "&##, ###.##"

Format – String n = "currency"

Format – String l = "percent"

Calculations

KPI (Key Performance Indication)

How to calculate calculated measures using BIDS wizard

Open BIDS

Create a new cube

In cube, select calculations tab
Click on new calculated members and set Name – [total amount]

Parent Hierarchy – Select MEASURES from dropdown list.

Expression – Define the following expression by drag and drop the measures from the measures section (from fact table)

[Measures]. [Sales amount]

[Measures]. [Tax amount]

Format String – Select “currency” option

Color – Click and Select any color

Background color – click and select any color

Click ok

In calculating tab click process to deploy (or) process into analysis services server.

Select browser tab expand measures you would observed that the total amount calculated measure is available.

KPIS (Key Performance Indicators)

In analysis services, represent Data to measure Business Success at very high level (mostly at organization level (or) regional level)

Steps to configure KPIS in a cube:

Open BIDS

Create a valid cube

Select KPIS tab in a cube

Click on new KPI and it will open one template for filling the desired information

Name – Provide any name to new KPI like fact interest sales KPI

Associated measure group – KPI belongs to one particular

Here you can specify a particular measure group (or) you can specify all.

Value Expression:

This is a actual value of KPI this may be a row measure (or) MDX numeric Expression.

Drag and drop the following Expression to KPI value.

[Measure] . [order Quality]
**Goal Expression:**

Here our new created measure will come into picture. We will get value Expression from measure group (order Quantity).

But the goal value will be target measure groups. Provide the following Expression as target (or) goal Expression.

\[ \text{[Measure]. [Total product cost]} \]

**Status:**

Its checks the state of KPI at any point. Status MDX Expression should return value between -1 and +1.

Provide the following Expression to find out the status of the KPI.

\[ \text{IIT (KPIVALUE ("[Measure]. [Order Quantity]"> KPI GOAL ("[Measure] . [Total product cost]") , -1, 1)} \]

In the above Example if KPI value greater than KPI Goal then the status is negative (Bad performance) (or) else positive (good performance).

**How to generate a Report from Analysis services cube:**

1. Open BIDS
2. In Analysis services create a valid cube and deploy it to analysis Service server.
3. Open BIDS create Report server project
4. In solution Explorer set shared Data sources.
5. Right click and select Add new Data source and set,
6. Name – Dsrc product sales details
7. Type – Select Microsoft SQL server Analysis services option from drop down list
8. Check Edit
9. Server name – Local host (or) Actual Server name (or) IP Address
10. Data base Name – Cube Name (product Details morning 8:30)
11. Test connection
12. Click ok thrice
13. Select Reports option
Report click and select Add

Select Reports template Rename it as product details from cube. RD1.

Click add

In Data tab add new Data set and set,

Name – Dset product details

Data source – Select Dsrc product sales details

Command type – text

Click ok

Drag and drop the following fields from the available Meta data section

[Facts and dimension tables]

From Dim product,

[English Product Name,] [Class],

[Color],

[Dealer price],

[List price]

From measure,

[Measure]. [Fright]

[Measure]. [Unit price]

Press Alt + ctrl + D for Data set

Go to layout tab design any Report item (table, matrix, chart etc.,)

Now the Report is displaying all the Reports from the specified cube. How to set parameterized Reports from cube: - From the above Report example.

In a cube,

Let us start by creating a cube on adventure works DW data base.

1. Create Analysis Services project and name it product details.
2. Add a new data source in it, which directs to adventure Works DW database namely ‘Adventure Works DW’.

3. Add a new Data source view (named Adventure Works DW) which consists of table’s Dim product, Dim product sub category, Dim product category, Dim customer and Fact Internet sales.

4. Create a cube with default settings but selective measures as order Quantity, Unit price, Sales Amount and Internet sales count. Give cube name as Demo cube.

5. Go to the properties of the project (project details)>select Deployment tab and enter server name. (If you are working with default instance, you can keep the name provided by default i.e. local host).

6. After successful deployment, open SSMS and connect to Analysis Services to work with MDX queries.

7. Create a new query, connect Product Details database Enter following MDX query: SELECT FROM Demo cube

   In this case default member for order Quantity is displayed as 60938.

   Enter the query as:

   SELECT [Measure] . [Order Quantity] ON 0 FROM Demo cube

   Will also give the same result. (1 can also be replaced by COLUMNS) which is the axis.

8. Let us add one more measures as follows:

   In this case we can see two measures order quantity and sales amount on columns.

9. Let us give query for a specific product category bikes as follows

   SELECT [product].[product category key].[bikes] ON COLUMNS FROM Demo cube OR
   SELECT [product].[product category key].&[1] ON COLUMNS FROM Demo cube.

10. Let us add rows now (add another axis)

    SELECT [product].[product category key].&[1] ON COLUMNS, [customer].[customer key].MEMBERS ON ROWS FROM Demo cube with this we get sales amount for bikes for all different customers.

    This can be achieved by using ordinals as 0 and 1 instead or COLUMNS and ROWS.

11. SELECT NON EMPTY[product].[product category key].MEMBERS ON COLUMNS, [customer].[customer key].MEMBERS ON ROWS FROM Demo cube.

    to get all categories (we added NON EMPTY to get rid of NULL value)
12. Let us see how to add a calculated member WITH MEMBER Measures.[calculated members] AS ([measures].[order quantity]*[measures].[unit price])
   SELECT NON EMPTY{Measures.[calculated member],[measures].[sales amount]} ON COLUMNS, NON EMPTY[product].[product category key].MEMBERS ON ROWS FROM Demo cube.
   First we calculate the member and use it with select to view the result

13. If we replace members after product category key with CHILDREN we get the result as follow SELECT NON EMPTY{Measures.[calculated member],[measures].[sales amount]} ON COLUMNS, NON EMPTY[product].[product category key].MEMBERS ON ROWS FROM Demo cube.
   We get only the (non empty) children for categories, all categories are not displayed.

**MDX EXAMPLES:**

Q). How do I find the bottom 10 customers with the lowest sales in 2003 that were not null?

A). simple using bottom count will return customers with null sales. You will have to combine it with NON EMPTY or FILTER

SELECT {[Measures].[Internet sales amount]}ON COLUMNS, BOTTOM COUNT (NON EMPTY (DESCENDANTS([customer].[customer geography].[all customers],
   [customer].[customer geography].[customer]),
   [measures].[internet sales amount]))
   ,10 , [measures].[internet sales amount]) ON ROWS FROM [Adventure Works]
WHERE ([date].[calendar].[calendar year] .&[2003]);

Q). How in MDX Query can I get top 3 sales years based on order quantity?

A). By default analysis services returns members in an order specified during attribute design Attribute properties that define ordering are “order by” and “order By Attribute”.

Let’s say we want to see order counts for each year. In Adventure Works MDX query would be:
SELECT {[Measures].[Reseller Order Quantity]} ON 0, [Data].[Calendar].[calendar year].Members ON 1 FROM [Adventure Works];

Same query using Top count; SELECT {[Measures].[Reseller Order Quantity]} ON 0, Top count ([Date].[calendar].[calendar year]. Members, 3, [Measures].[Reseller Order Quantity]) ON 1 FROM [Adventure Works];

Q). How do you extract first tuple from the set? A). Use could use function set. Item (0)

Example:

SELECT {{[Date].[Calendar].[Calendar year].Members}.Item(0)} FROM [Adventure Works]

Q). How do you compare dimension level name to Specific value?
A). Best Way to compare if specific dimension is at certain level is by using ‘IS’ operator:

Example:

WITH MEMBER [Measures].[Time Name] AS IIF ([Date].[Calendar].level IS [Date].[calendar].[calendar Quarter], 'Qtr', 'Not Qtr')
SELECT [Measures].[Time Name] ON 0 FROM [Sales Summary]
WHERE ([Date].[Calendar].[Calendar Quarter].& [2004] & [3])

Q). MDX Query to get sales by product line for specific period plus number of month with sales.
A). Function Count (, execute empty) counts number of non empty set members. So if we cross join Month with measure we will get set that we can use to count members.

Query Example:

WITH Member[Measures].[Months with Non zero sales] AS

Syntax:
ALTER CUBE current cube| your cube name UPDATE DIMENSION,
“DEFAULT_MEMBER= “;

Q). I would like to create MDX calculated measure that is instead of summing children amounts, uses last child amount?

A). Normally best way to create this in SSAS 2005 is to create real measure with aggregation function last child. If for some reason you still need to create calculated measure, just use function. Last child on current member of date dimension, and you will always get value of last period child.

Example: We want to see last semester value for year level data. Lets first see what data values are at calendar semester level.

SELECT {[measure].[Internet Order Count]}ON 0,DESCENDENTS([Date].[calendar].[All periods], [Date].[calendar].[calendar semester]) ON 1 from [Adventure Works]

Q). How to calculate YTD monthly average and compare it over several years for the same selected month?

A). MDX Query:

WITH MEMBER Measures. MYYTD AS
SUM(YTD([Date].[calendar]),[Measures].[Internet sales Amount])
MEMBER Measures.MyMonth count As
SUM(YTD([Date].[calendar]),(COUNT([Date].[Month of year])))
MEMBER Measures.MyYTDAVG As Measures.MyYTD/Measures.MyMonthcount
Q). MDX query to get sales by product line for specific period plus number of months with non empty sales?
A). You can use COUNT() function with exclude empty option. For count function you specify set that is cross join of date members at the month level and measure that you are interested in.

WITH MEMBER [Measures].[Months with Above zero sales] AS
COUNT(
DESCENDANTS({[Date].[calendar].[calendar year].&[2003]: [Date].[calendar]).[calendar year].& [2004]},
[Date].[calendar].[MONTH])*[Measures].[Sales Amount],Exclude empty
)

SELECT{[Measures].[sales amount],[Measures].[Months with Above zero sales]} ON 0,[product].[product model lines].[product line].Member on 1
FROM [Adventure Works]
WHERE([Date].[calendar]).[calendar year].& [2003]: [Date].[calendar]).[calendar year].& [2004])

Q). How do I group dimension members dynamically in MDX? Source: MSDN SSAS New group?
A). You can create calculated members for dimension and then use them in the query.
Example below will create 3 calculated members based on filter condition:

WITH MEMBER [product].[category].[case result 1] AS’
Aggregate(Filter([product].[category].[All].children,
[product].[category].current member. properties("key")<”3”))’
MEMBER [product].[category].[case result 2] AS’
Aggregate(Filter([product].[category].[All].children,
[product].[category].current member. properties("key")="3”))’
MEMBER [product].[category].[case result 3] AS’
Aggregate(Filter([product].[category].[All].children,
[product].[category].current member. properties("key")>"3")'

SELECT NON EMPTY{[Measures].[order count]}
ON COLOMNS
, {[product].[category].[case result 1], [product].[category].[case result 2],
[product].[category].[case result 3]} ON ROWS
FROM[Adventure Works]

Q). How can I compare members from different dimensions that have the same key values?

Let’s say I have dimensions [Delivery Date] and [ship Date]. How can I select just records
that were Delivered and shipped the same way?

A). You can use Filter function and compare member keys using properties function:

SELECT{[Measures].[Internet Order count]} ON 0, FILTER(Non Empty cross join([ship
date].[date].children,[Delivery Date].[Date].children),
[ship date].[Date].properties('key')) ON 1 FROM[Adventure Works]

Q). How can I get attribute key with MDX

A). To do so, use member_key function:

WITH MEMBER Measures.product key as [product].[product categories].current
member.member_key
SELECT{ Measures.product key} ON axis(0),
[product].[product categories].member on axis(1)
FROM[Adventure Works]

Q). How do I create a Rolling 12 months accumulated sum (Internet sales Amt R12ACC)
that can show a trend without seasonal variations?

A). Here is Query example

WITH MEMBER [Measures].[Internet sales AmtYTD] AS
SUM(YTD([Date].[calendar].current member), [Measures].[Internet sales Amount]),
Format_string= "# ######" "#
MEMBER [Measures].[Internet sales AmtPPYTD] AS
SUM(YTD(parallel period([Date].[calendar].[calendar year],1,[Date].[calendar].current
member)),
[Measures].[Internet sales Amount]),
MEMBER [Measures]. [Internet sales AmtPY] AS
SUM(Ancestors(parallel_period([Date].[calendar].[calendar year],1,[Date].[calendar].[current member]), [Date].[calendar].[calendar year],
[Measures].[Internet sales Amount]),
Format_string= "# # # # # # # # #"

MEMBER [Measures]. [Internet sales AmtR12ACC] AS
(([Measures]. [Internet sales AmtYTD] + [Measures]. [Internet sales AmtPY]) - [Measures]. [Internet sales AmtPPYTD])
Select{[Measures].[Internet sales Amount], [Measures].[Internet sales AmtYTD],
[Measures].[Internet sales AmtPPYTD], [Measures].[Internet sales AmtR12ACC]} on 0,
[Date].[calendar].[Month].Members on 1
FROM[Adventure Works]
Where (([Date].[calendar year].& [2004])

Q). How to setup calculated measures as default measure for a cube?
A). Use Alter cube statement on measures dimension Example:

ALTER CUBE CURRENT CUBE UPDATE DIMENSION
Measures,
DEFAULT_MEMBER = [Measures].[Profit]

Q). How can I write MDX query for the count of customers for whom the earliest sale in the selected time period (2002 and 2003) occurred in a particular product category.
A). Example of such Query:

WITH SET [First sales] AS
FILTER(Non Empty([customer].[customer geography].[customer].MEMBERS
[Date] .[Date].[Date].MEMBERS, [Measures].[Internet sales Amount])
AS MYSET
MYSET,CURRENT ORDINAL=1 or
NOT(MYSET,CURRENT.ITEM(0) IS MYSET.ITEM
(MYSET,CURRENT ORDINAL-2).ITEM(0)))
MEMBER [Measures].[customers W/First sales] AS COUNT (NON Empty([First sales],
[Measures].[Internet sales Amount]),
FORMAT_STRING = ‘# , #’
SELECT { [Measures].[Internet sales Amount], [Measures].[customers W/First sales] } ON 0,
[product].[product categories].[category] ON
FROM[Adventure Works]
WHERE(( [Date].[calendar].[calendar year].&[2002], [Date].[calendar].[calendar year].&[2003]), [customer].[customer geography].[city].&[category]&[AB]);

Q). How do you write MDX query that returns measure ratio to parent value?
A). Below is example on how is ratio calculated for measure [order count] using Data dimension
Using parent function, your MDX is independent on level that you are querying data on. In
example below, if you query data at year at year level, ratio will be calculated to level [ALL]:
WITH MEMBER [Measures].[order count ratio to parent] AS
IIF( ([Measures].[order count],[Date].[calendar].current Member.parent)=0,
NULL,
[Measures].[order count],
([Measures].[order count],[Date].[calendar]. current Member.parent)
)
;
FORMAT_STRING=”parent”
SELECT{ ([Measures].[order count], [Measures]. [order count ratio to parent])
ON 0, [DESCENDANTS([Date].[calendar].[All periods])] ON 1
FROM[Adventure Works]