

Nalash.G

Rs : 90/-



**VIJAY Technologies**

**MS - BI**

By  
**Mr. B. RAMA RAO**

3rd Floor, Manjeera Plaza, Opp. Aditya Trade Centre, Ameerpet, Hyderabad

Tel : 9704015525, 9550647461



## MS - BI

MS-BI : Microsoft Business Intelligence

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 analysed 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 analyse your business data and it also evaluates the performance of the organisation.

For example:

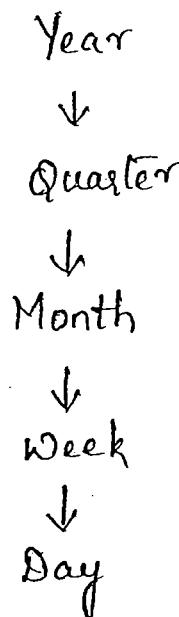
Sales amount would be a measure. This measure stores in fact table.

(3) Hierarchy:

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

For example:

One possible hierarchy in the Time dimension is,



#### (4) Dimension (Attribute):

A Unique level within in a dimension is known as Attribute.

For example: P.ID, P.Name are Attributes or dimensions in Dim product.

#### Data Warehouse:

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

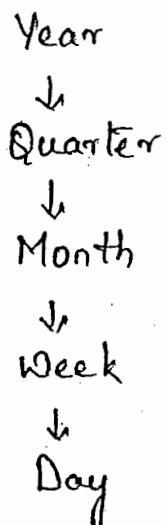
- \* Time Variant
  - \* Integrated Data Base
  - \* Non-Volatile
  - \* Subject oriented
- } TIDS

#### 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 period.

Example:

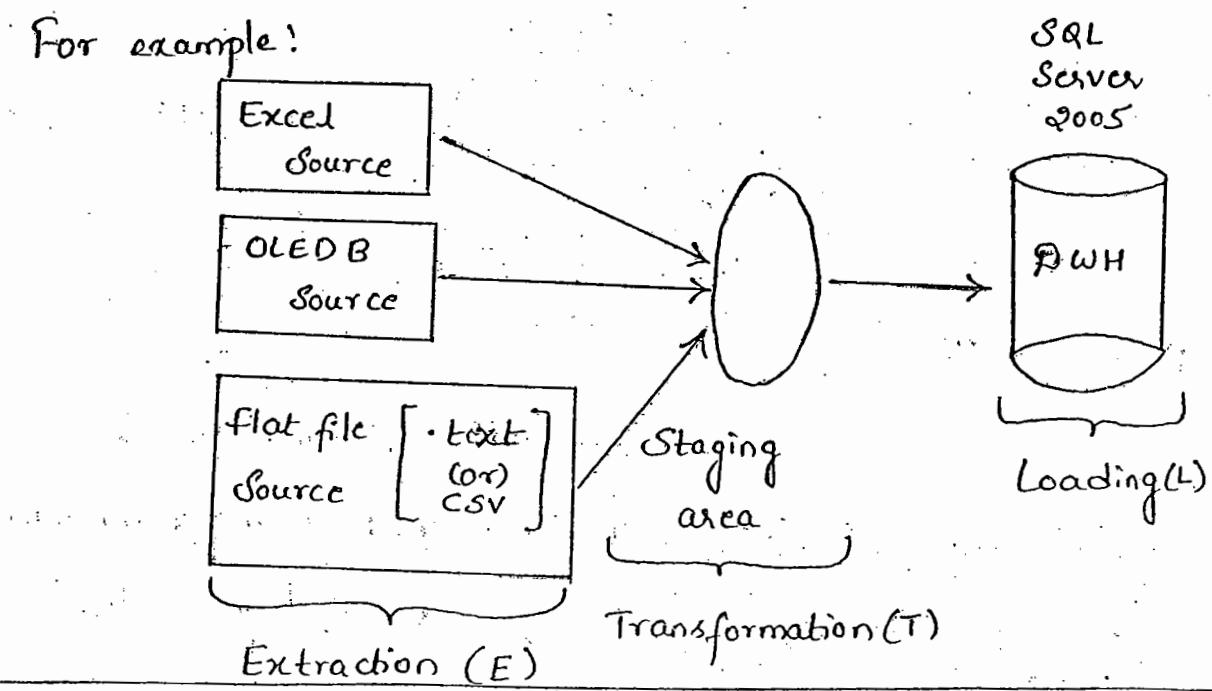


From the above example the business users or business analytics can analyse their business information with respect to year wise or quarter wise or month wise or week wise or day wise as well.

## ② Integrated Data base:

Data Ware house is built by integrating 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 dataware house.

Simply dataware house can be build by integrate the data from various operational sources.

### ③ Non-Volatile:

Once the source data is inserted into the dataware housing it doesn't reflect the changes since it is static or read only data.

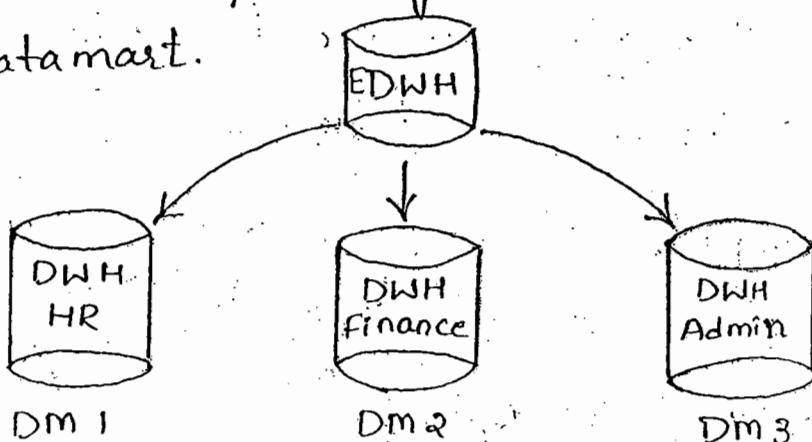
For example:

Policy ID	Policy Name	Customer Name	Address	Modified date
123	ULP	Rama	LIG-13	2009-12-12
123	ULP	Rama	LIG-14	2010-10-12
123	ULP	Rama	LIG-15	2010-10-13

### ④ Subject Oriented:

DataWare house is a subject oriented data base and it stores specific data about specific department in the complete organisation. It is also known as Datamart.

For example:



Note:- Data Mart is also known as HPQS  
(High Performance Query Structures)

### Data Warehousing Architectures:

In designing data models for dataware houses or datamarts, the most commonly used schema types are,

- ① Star schema
- ② Snowflake schema.

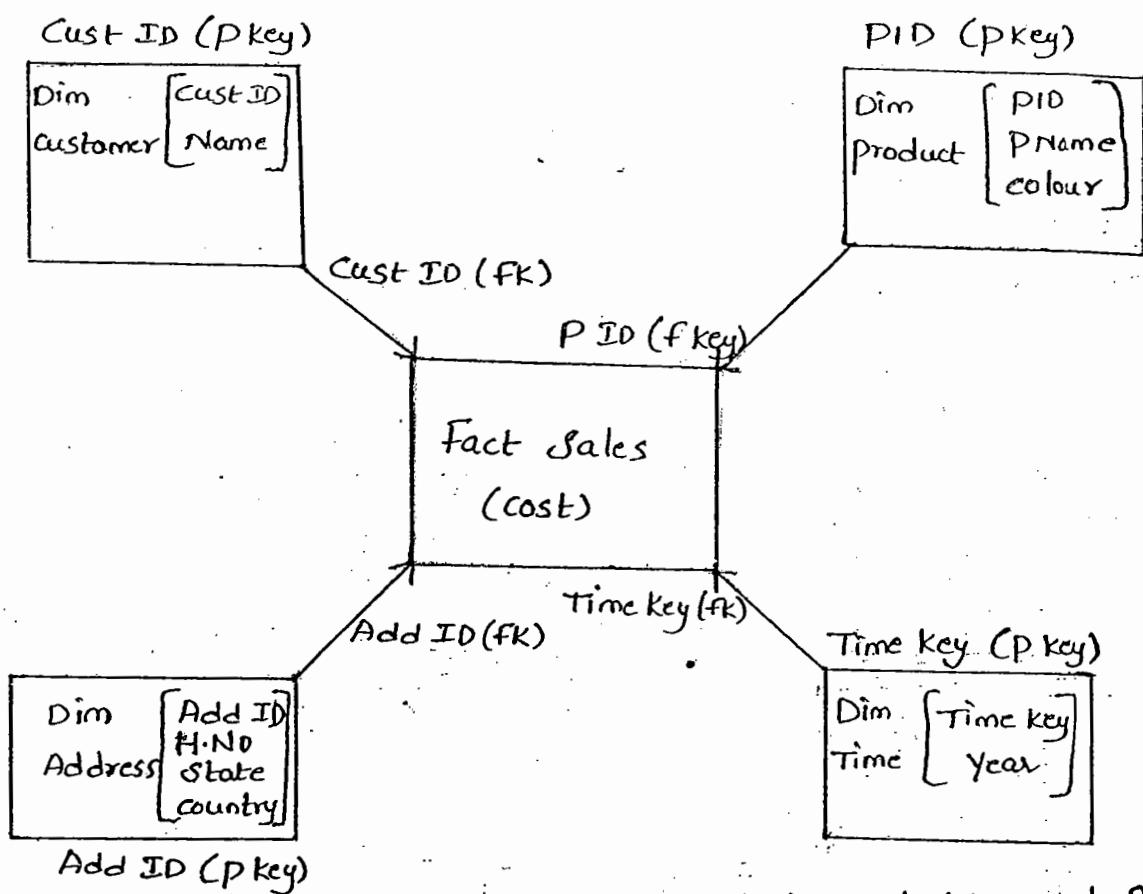
#### \* Star Schema :

The star schema dataware housing design contains atleast 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:

- ④ A simple star schema consists of one fact table and a complex star schema have more than one fact table.
- ⑤ All measures in the fact table are related to all the dimension tables.

#### Example:

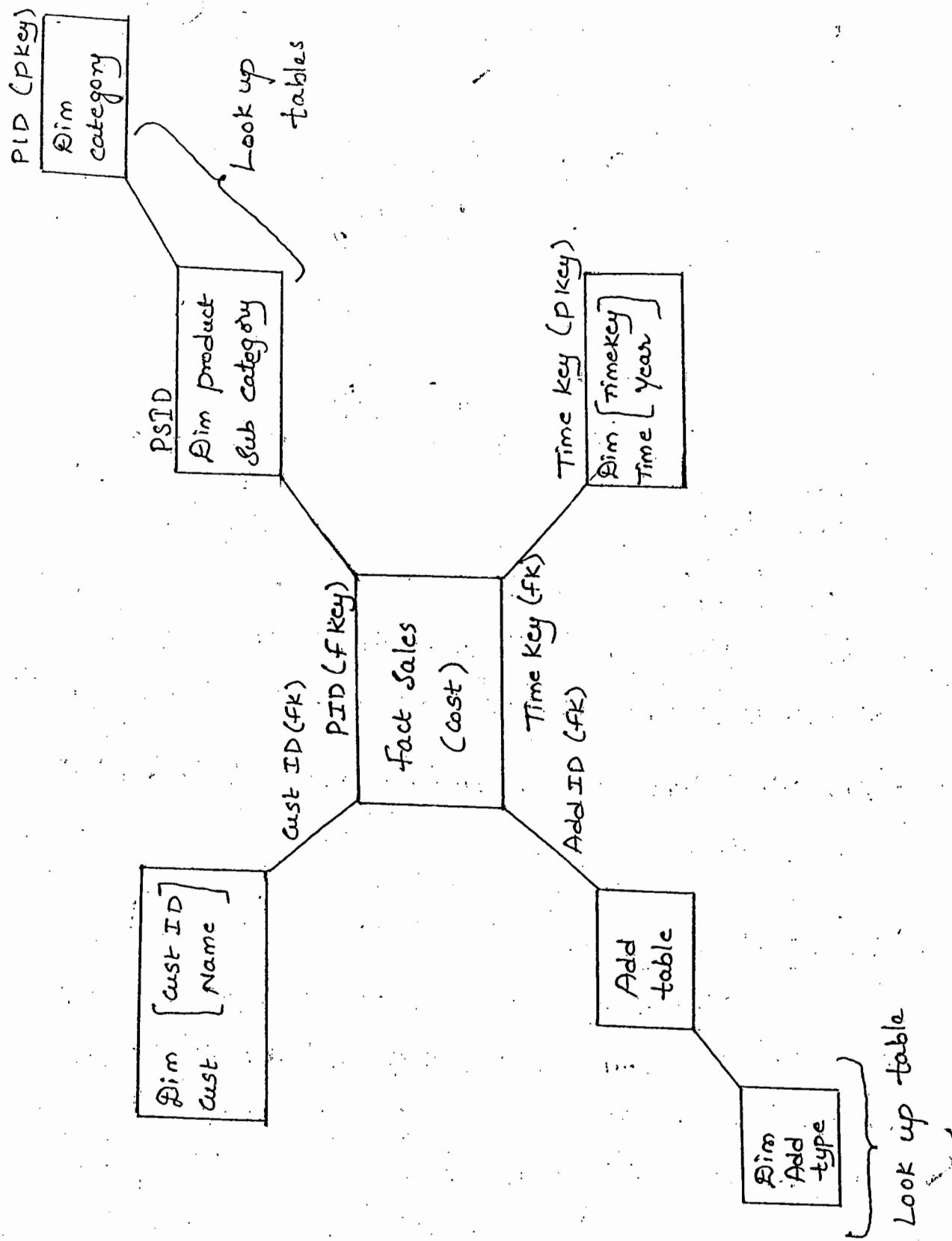


From the above diagram the fact table contains customer ID, PID, Address ID, cost and time key attributes.

Note: The different dimensions are not related to one another.

## ② Snowflake Schema:

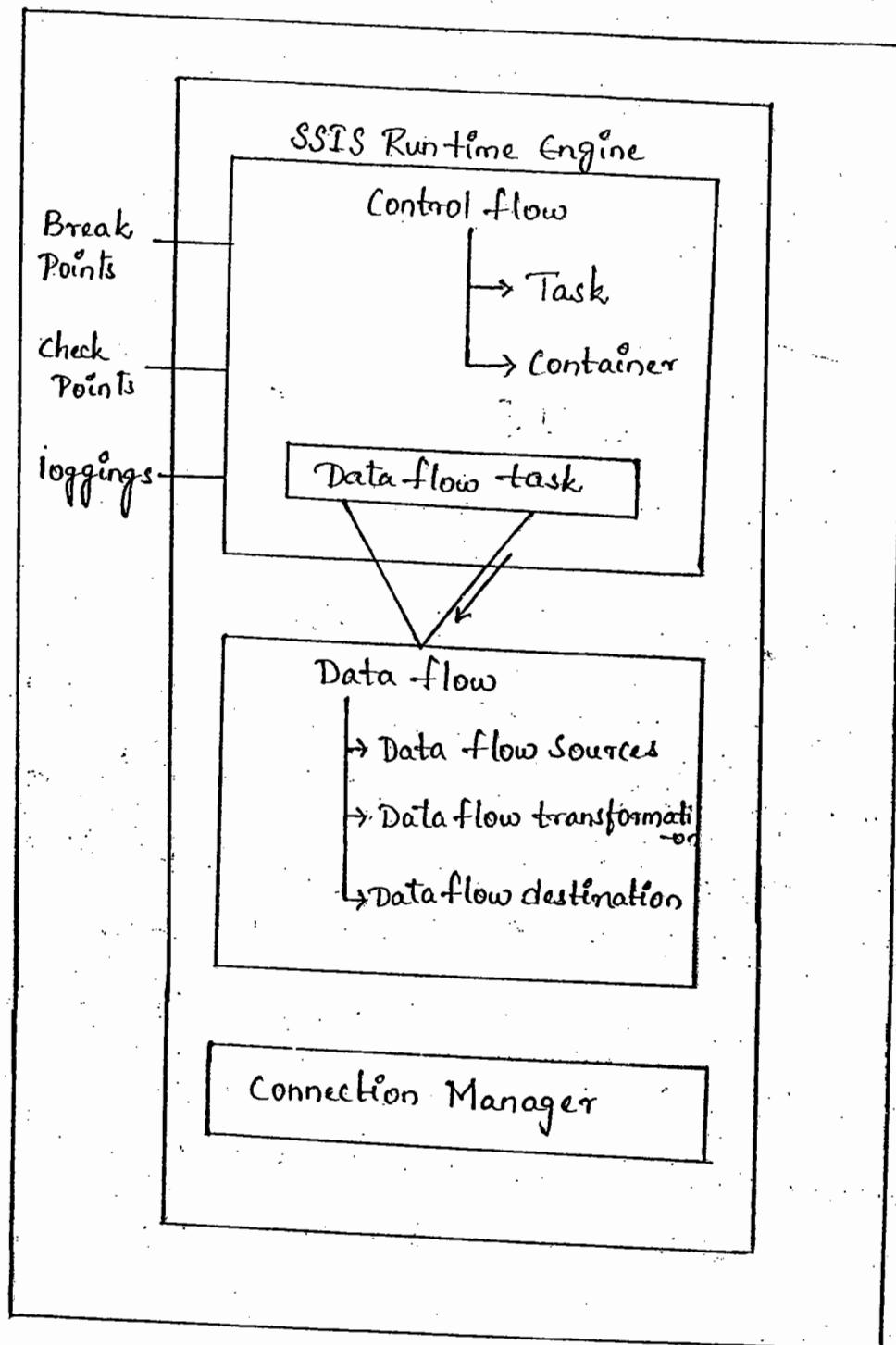
The Snowflake Schema is an extension to star schema, where each point of the star schema explodes or divides into more points. In star schema each dimension is represented by a single dimension table, whereas in a Snowflake schema the dimension table is normalised into multiple look up tables, each representing a level in the dimensional hierarchy.



In the above dataware housing 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.

## SSIS (SQL Server Integration Services)

SQL Server Integration Service is an ETL 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 database or file system.



Package .dtsx

SSIS Architecture is categorised into two components

### 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 runs. The control flow contains various tasks and containers as well.

Task: A unit of work in a workflow.

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 loop container, sequence container 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, Look up 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.

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 dataset.

Steps to configure data conversion transformation

Start



All Programmes



Select Microsoft SQL Server 2005



Select SQL Server Business Intelligence Development Studio

(to implement all the BI activities)

↓  
Select file menu

↓  
Select New

↓  
Select project

↓  
Business intelligence project options under  
Project types

↓  
Integration services project under templates section

↓  
change the project name as [Evening 7.30 Batch]

↓  
change the project location

↓  
click ok

↓  
select package.dtsx in solution explorer and  
rename it as data conversion.dtsx

↓  
In control flow, drag and 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 tab drag and drop OLEDB 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 [Local host (or) .(or) Server name]

↓  
Select data base name [Adventure works from the  
drop down list]

↓  
click test connection to evaluate the connection

↓  
click ok

↓  
click ok

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

↓  
Select human resources, employee table name  
from the drop down list

↓  
Select columns option from left pane

↓  
click ok

↓  
Drag and 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 OLEDB 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 data base 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 Marital status and rename the table as converted data.

↓  
click ok

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

## Steps to execute SSIS package:

In Business Intelligence Development Studio, Alt + Control + L for solution explorer, 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 use of Derived column transformation is to create or derive new columns by using the existing source columns or by using variable or by using available functions.

## Steps to Configure Derived Column Transformation:

Start  
↓  
All programmes  
↓  
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 dataflow, 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) Servername]

↓  
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,
- [Rowguid] Unique Identifier,
- [Modified date] DATETIME,
- [Ref Date] DATETIME )

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

Derived Column Name	Expression	Data type	Length
Ref date	(DT_DBDate) getdate()	DT_DB date	

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

## 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.

Open Business Intelligence Development Studio

↓  
Create a new package and rename it as execute SQL.dtsx

↓  
In control flow drag and drop Execute SQL Task onto design area

↓  
Double click on Execute SQL Task to edit it

↓  
Provide the following steps

↓  
Select New Connection

↓  
Click New

↓  
Provide Server Name (Local host or) (or) Server name

↓  
Select Adventure Works data base from drop down list.

↓  
Click test connection to evaluate the connection between data base and SSIS

↓  
Click OK

↓  
Click OK

\*

SQL Source type - Select Direct Input (default)

Note: Here, we have 3 options

- ① Direct input - User can provide or enter SQL command directly on available notepad.
- ② File Connection - We can pass the query through any file system by providing path and file name.
- ③ 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 (and 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.

Variables: In SSIS the variables are categorised into two parts

① System Variables

② 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 by System Scope resolution operator (::). That means all system variables should start with System::

User Defined Variable: 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 .dtsa

↓  
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  
checkbox 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

Source code	SBAT Type	Partner Type	Funded Amount
81818	MSA	Builder	60000
81540	B1	Realtor	40000
12345	MAP	Realtor	9000

## Merge Transformation:

The Merge Transformation combines two sorted data sets into single output based on values in their key columns. This transformation requires that the inputs or sources are sorted and then Merged columns must have same data type.

## Steps to configure Merge Transformation

Open Business Intelligence Development  
↓ studio

In Integration Services project, create a new package and rename it as Merge.dtsz.  
↓

In control flow drag and drop Data Flow Task and rename it as Data Flow Task merge.  
↓

In Data Flow drag and drop OLEDB source rename it as source 1  
↓

Double click on source 1 to configure it.  
↓

Select provide connection Manager if exists  
↓

Select Human Resources. Employee table from drop down list  
↓

- Select columns from left pane and click ok to save changes  
↓

Right click on source 1 and select show advanced editor and set the following properties,  
↓

Select input and output properties Tab,  
↓

Select OLEDB source output and set,

Is Sorted — True

↓  
Expand OLEDB source output and also expand output columns.

↓  
Select 1<sup>st</sup> column through which we are going to make a mapping between two sources (Employee ID) and set,

Sort key portion - 1

↓  
click Refresh

↓  
click Ok

↓  
Drag and Drop another OLEDB source and rename it as source 2.

↓  
Double click on source 2 provide connection Manager if exists

↓  
Select Human Resources Employee Address from drop down list.

↓  
Select columns

↓  
click Ok

↓  
Right click on source & select advanced editor to sort the data and set the following properties

↓  
Select input and output properties Tab.

Select OLEDB source output and set,

IS sorted - True



Expand OLEDB source output and also expand output columns.



Select 1<sup>st</sup> column through which we are going to make a mapping between two sources (EmployeeID) and set,

Sort key portion - 1



click Refresh



click ok



Drag and drop merge transformation

Make a connection from Source 1 to Merge and select Merge input 1. option in input, output selection editor



click ok



Make a connection from source 2 to Merge



Double click on Merge to make sure that all columns are mapped.



Drag and Drop OLEDB destination, make a connection from merge to OLEDB destination



Double click on OLEDB destination



Provide destination connection Manager and click new to create destination table and rename the table as merged data.



click ok



select mappings



click ok



In solution explorer select the package and select execute package

### Merge Join Transformation:

The merge join transformation combines two sorted data sets into single output using inner join (default join), left outer and full outer joins.

### Steps to configure Merge Join Transformation:

Configure 2 OLEDB sources (source 1, source 2) from the previous example.



Drag and drop merge join transformation



Make a connection from source 1 to merge join and select merge join left input option from input output selection editor



Make a connection from source 2 to merge join transformation



Double click on Merge Join and select left outer join as join type and select the following columns from both source 1 and source 2

↓  
click ok

Drag and drop OLEDB destination and make a connection from merge join to destination.

↓

Double click on OLEDB destination

↓

Provide destination connection manager if exists and click new to create destination table.

↓

Rename the OLEDB destination as merge join data.

↓

click ok

↓

Execute package

Union All Transformation: It combines multiple inputs into single output. It differs from Merge and MergeJoin transformation because Union all doesn't require sorted input. However the first input is the reference input that all subsequent inputs must match the following criteria.

1. No. of columns
2. Data type
3. Length
4. Precision (Decimal point)

conditional split Transformation! It routes the input data to different outputs based on case conditions if no case (or) conditions are met the data must be routed to default output. The implementation of conditional split is similar to case decision structure in general programming languages [switch case]

Scenario: To test the package whether it is successfully executed (or) fail using Conditional split, derived column transformation and Union all.

open Business Intelligence Development Studio.



Create a new package and rename it as Test.dtsx



In control Flow, drag and drop Data Flow Task.



In Data Flow, drag and drop OLEDB Source



Double click on OLEDB source to configure it.



Provide Connection Manager if exists



Select Human Resources. Employee table from the ddl.



Select columns from left panel



click ok



Drag and drop Rowcount Transformation to

find out SRC Row count.



Make a connection from SRC to Rowcount



Double click on Rowcount 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 Rowcount to Derived  
column



Double click on Derived column to define Execution  
Date provide the following expression.



Execution Date - get date( )

click ok



Drag and drop rowcount Transformation and  
rename as RCDest.Count



Make a connection from Derived column to  
Rowcount.



Double click on Rowcount and Provide the  
following property

## custom properties

Variable name - UNDstCount

↓  
click Refresh

↓  
click Ok

↓  
Drag and drop OLEDB Destination and make a connection from Rowcount 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

Name	Data type	Value
UNSrc count	Int 32	
UNDst count	int 32	
UNSolution Name	String	Morning 8.30 batch
UNTable Name	String	Tested Data

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::;vsolution Name]

Package Name @ [sys\_];:: Package Name]

Table Name @ [user :: UV Table Name]  
Source Count @ [user :: UV SrcCount ]  
Destination Count @ [user :: UV DstCount]



click ok



Drag and Drop, conditional split Transformation  
to check the condition



Make a connection from Derived column to  
conditional split.



In condition split Transformation editor, provide  
the following condition.

Output Name

Condition

Case 1

[source count] :



[Destination count]

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.

<u>Derived column Name</u>	<u>Expression</u>
Status	"success"

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.

<u>Derived column Name</u>	<u>Expression</u>
Status	"Failure"

click ok

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 OLE DB 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 if exists



click New to create new destination table and rename it as SSIS-Log



click OK



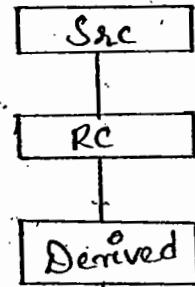
Select Mappings



click OK



Execute package



### Flat File Source:

Scenario: Remove duplicate records from file systems using either Sort transformation or 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



Select columns in flat file source editor



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 drop 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.

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 section right click and select new flat file connection.

↓  
Provide connection manager name and description

↓  
Click Browse and select student details.txt and click open

↓  
click ok

↓  
Drag and drop flat file source

↓  
Double click on flat file source 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 Managers 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

Bulk Insert Task: Bulk Insert Task can be used to insert 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 with in 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, Vijay, BTech, 71
- 3, Siva, MCA, 70
- 4, Laxmi, MCA, 80
- 5, Aarpana, BSC, 90

Open Business Intelligence Development Studio



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



In control flow drag and drop the 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

Column 0	- SNO
column 1	- SName
column 2	- Qualification
column 3	- 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

- ure work.

Destination table - Select [Adv. works].  
[dbo].

[Bulk Insert] from dropdown 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 at 1 transaction. Server commits or rolls 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 input 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<sup>st</sup> row in the specified data file.

Tips to improve the performance of bulk insert task.

① It is advised not to perform any sort operation while loading the data from text file using bulk insert task.

② The data will be loaded faster when there are no indexes created on staging table or destination table.

V. Jind

Data Viewers: Data Viewers used to debug the package and also user or developer can monitor the data which is flowing from source to other stages (transformations or destinations). In ssis Data Viewers are categorised into 4 parts

(1) GRID: Select Grid type and also select Grid tab.

Select the column 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) Scatterplot (x,y) : Select scatterplot type and also click select scatterplot (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 Viewers.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 viewers option



Click Add to add any data viewer



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 implemented only on development environment for debugging the package.

1. Jmp

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.

Create 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

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 data base



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 data base



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 fileName - 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



Execute package



The package execution starts at task 2

### For Loop Container:

For Loop Container is used to loop through a specified task at a specified number of times whenever the evaluation expression becomes false automatically the control comes out of the loop

### Steps to configure For Loop Container

Open Business Intelligence Development Studio



Create a new package and rename it as ForLoop  
dtsx

In Control flow drag and drop data-flow task

Double click on Data Flow Task to configure it

Drag and drop Excel source and configure it with  
any valid excel file

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

Double click on OLEDB destination provide Connection  
Manager if exists

Click new to create destination table and rename  
it as For Loop Container.

Select mappings and click ok

Note: If you execute the package the Data Flow  
Task will be executed only once. If you want to  
execute a task or a container multiple number of  
times place a specified task or container on For  
Loop Container.

In control flow drag and drop For Loop Container  
and place the Data Flow Task on for Loop Container

Right click on Control flow and select variables  
declare the following variable of type integer  
UV Counter

Q1

Double click on ForLoop container and provide the following expression

Int Expression → @ UV Counter = 0

Eval Expression → @ UV Counter < 0

Assign expression → @ UV Counter = @ UVCounter + 1

↓

Click Ok

↓

Create package

### For Each Loop Container:

For Each Loop Container is used to loop through multiple files, datasets, XML files, multiple sheets from a single Excel work book.

The above mentioned operations can be achieved by using 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 Rowset enumerator:

It is used to enumerate through schema information about a data source.

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

For Each Node List Enumerator: It is used to enumerate 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 such as copy, create, delete and move directories.

Scenario: Execute multiple files and move (or) copy executed files to new directory.

Open Business Intelligence Development Studio

↓  
Create a new package and rename it as For Each Loop .dtsx

In Control flow, select variable from rig click context menu and define the following variables

UV Source path      string      D:\ssis\packages\package

UV Packages To Run      string

UV Fullpath      string

↓  
In variables window, select UV Fullpath

Press F4 for properties, set the following settings

Evaluate As Expression - True

Expression - provide the following Expression

@ [USER::UVsourcepath] + " " + "executed on - " +  
(DT\_WSTR,20) (DT\_DBDATE) @ [SYSTEM::StartTime]  
↓

close properties window; copy any .dtsx files at  
the specified locations,

D:\ssis packages\packages

↓  
In control flow, drag and drop file system  
task and rename it as FST

↓  
Create a new directory with today's date

↓  
Double click on file system task and provide the  
following settings

Operation - Create dictionary

Is Source path Variable - True

Source Variable - USER::UVfullpath

Use Directory if exists - True

↓  
Drag and drop For Each Loop Container and make a  
connection between file system task and for each loop  
container.

↓  
Double click on For Each Loop Container to edit it

In collection page,

Enumerator - Select foreach file enumerator

Folder - Specify path where the files are located,

D:\SSIS packages\packages

File - \* .dtsx

In variable Mapping page,

Select user:: uv package to run variable from

ddl.

↓  
Click ok

Drag and drop execute package task on to foreach  
loop container

↓

Double click on execute package task to edit it

↓

Location - File system

Connection - New Connection

↓

Click Browse

↓

Select any package (Test condition.dtsx) and  
click open and click ok.

↓

In connection manager section, select test condition  
dtsx.

↓

Press F4 for properties

↓

Expression - Click Browse

↓

Select connection string property  
name and click browse

↓

Expand variable in edit expression editor.

Drag and drop USER::UV package to Run Variables  
on to expression section

Click Evaluate expression to check the syntax of  
expression.

Click ok Twice

Drag and drop file system task onto foreach loop  
container

Make a connection between Execute package task  
and File system Task.

Double click File System Task, produce the following  
Properties.

Operation - copy file

Is source path variable - False

Source Connection - Select Test Condition.dtsx

Is Destination path Variable - True

Destination Variable - USER::UV fullpath

Over write Destination - True

click ok

Execute package

## Loggings :-

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

## Steps to Configure Loggings!

Open Business Intelligence Development Studio and configure a package

In control flow, right click and select logging

Check containers check box

In provider and log tab,

Provider type - Select ssis.log provider for Text File

Click Add

Check newly added log provider check box

In configuration header, select new

connection

Usage Type - Select Create File Option

File - click Browse

Provide file Name (logged information .txt) and check open

click ok

Select Details tab,

Check events check boxes on which you want to capture log information. On Error, On Warning, On Query Failed, On Pre Execute.

↓  
Click Advanced button

↓  
click ok to save the changes done

### Different ways to execute SSIS package:

#### (1) Execute SSIS package Using BIDS:-

During the 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 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.

For example: DTExec.exe /F "C:\Packages\check point.dtsx"

(3) Execute ssis package using DTExecUI.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 DTExecUI.exe and press enter which will open up 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.

(or)

(4) The execute package utility is also used when you execute the ssis package from integration service node in SQL Server management studio.

Open SQL Server Management Studio

↓  
connect to integration services

↓  
expand stored packages

↓  
expand file systems

Select file systems and right click, select import  
Package editor.

Package location - File system

Package - Click Browse

↓ 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 foreach loop container



Select Steps page



Click New



Step Name - Load Foreach Loop Container

Type - SQL Server Integration Services



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 Foreach Loop

- container

Schedule Type - Recurring



Set the Frequency (When to start the execution  
of the specified Package)



Click Ok twice

Look Up Transformation: It is used to compare the data from source to reference dataset. Using reference dataset using simple equivalent.

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 look up.dtsx



In control flow drag and drop data flow task.



In data flow, drag and drop OLEDB Source



Double click on OLEDB Source to edit it



Provide Connection Manager if exists and select Production.product category



Select columns and click ok



Drag and drop Look up Transformation and make a connection from source to look up



Double click on Lookup to edit it



Provide connection manager and select production.product sub category



Click configure error output and set Redirat  
Rows under Error header,



click ok



click ok



Drag and Drop OLEDB destination



Make a connection from Look up to destination



Double click on OLEDB destination to configuire 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 and drop OLEDB destination to capture  
unmatched records from source to reference  
dataset.



Make a connection from Look up to OLEDB destination  
using error output ( Red data flowpath)



click ok in Look up error output editor



Double click on destination to configuire 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 the destination
- ③ In Look up transformation editor remove the mappings between all the columns except between Product Category ID.

Scenario:- To create Dynamic flat file destination.

In control flow drag and drop Data Flow Task ↓

Define the following variables with respect to Package.

Name	Data type	Value
UVSource path	string	D:\SSIS packages\packages
UV File Name	string	Vendor

In Dataflow drag and drop OLEDB Source

Double click on OLEDB source to edit it

Provide Source connection Manager and select Production, production sub category from drop down list.

Select columns

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

Double click on flat file to edit it.

Click new to configure new flat file connection Manager.

Select Delimited flat file format.

Click ok

Connection Manager Name - Dynamic Vendor flatfile

Description - Dynamic Vendor flatfile

File Name - Type the following path and file with extension which is not available at destination.

D:\sis\spackages\Packages\Vendor.txt

Select column page

Click ok

Select Mappings page



Click ok.

In Connection Manager,

Select dynamic Vendor flat file



Press F4 for properties



Expression - Click Browse



Select connection string and click

In expression Builder, provide the following  
expression which create a new flat file dynamic  
ally.

`@[user::UV Source path] + "1" + @ [user::UV File Name]  
+ "-" + (DT_WSTR, 10) (DT_DBDATE) @[system::start  
Time] + ".txt"`



Click ok twice



Close property window



Execute package

A custom Task

Script task: Script task is used to design custom  
interphases.

Scenario: Create a text file on every corresponding  
month dynamically using script task.

Define the following variables type string

UVSourcepath : D:\ssis\packages\packages

UVFileName : Product Category Details on

UVFullPath :



Drag and Drop script task



Double click on script task to edit or configure it.



Select Script from left panel and set,

Read Only Variables - UVSourcepath, UVFileName

Read write Variable - UVFullPath



Click Design Script, Opens MS Visual Studio for Applications IDE,



Provide the following VB.NET Script to create dynamic text file with the specified name.

Dim sSourcepath As String

Dim sfilename As String

Dim sFullPath As String

'D:\ssis\packages\packages

sSourcepath = Dts.Variables("UVSourcepath").

Value.ToString

'Product category Details On

sFileName = Dts.Variables("uvFileName").Value  
ToString

'D:\SSIS packages\packages\Product category Details on-

sFullpath = sSourcepath + sFileName + "-" + Month

(Now()), ToString() + Year(Now()).ToString() + ".txt"

Dts.Variables("uvFullpath").Value = & Fullpath.ToString()

↓

Select Debug Menu and select build to build the  
above Scripting

↓

Select file menu and select close and return

↓

click ok

Drag and Drop dataflowtask and make a connection  
from script task to Data Flow Task

↓

In Data Flow Drag and Drop OLEDB Source

↓

Double click on OLEDB Source to configure it

↓

Provide connection Manager if exists.

↓

Select product category table

↓

Select columns from left pane and click ok

↓

Drag and drop flat file destination and make a  
connection from OLEDB Source to flat file destination

↓

Double click on flat file destination



Click new to create new Connection Manager



Select Delimited flat file format



Click ok



Provide flat file connection manager name and description if any.



Type the following path

D:\SSIS packages\packages\product category Details on.txt



Select columns from left panel



Click ok



Select mapping from left panel



Click ok



In connection manager, select flat file connection manager.



Press F4 for properties and set,

Expression - click Browse



Select connection string and click  
Browse to build the expression in  
expression builder



Expand Variables

Drag and Drop USER::UV Fullpath on to expression section, i.e @ [USER::UV Fullpath]



click ok twice



close properties window



Create the package

### Providing Security for SSIS Package

Protection level is in SSIS packages 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.

Example: The sensitive information would be password to the Database.

### Steps to Configure protection level in SSIS

Open Business Intelligence Development Studio



Create OLEDB connection with server authentication and provide username and password of the server



Design package



Select package in control flow, right click



Select properties,

→ Security

Protection level - Don't Save

Sensitive

Don't Save Sensitive: When you specified Don't Save Sensitive as the protection level, any sensitive information is not written to the package XML file when you save the package. This could be useful when you want to make sure that anything sensitive is excluded from the package before sending it to someone. After saving the package with this setting, open the OLEDB Connection Manager, the Password is blank even though Save my password check box is checked.

Encrypt Sensitive with user key: Encrypt sensitive with user key encrypts sensitive information based on the credentials of the user who created the package.

There is a limitation with this setting, if another user (a different user than the one who created the package and saved it) opens the package the following error will be displayed, error loading encrypt sensitive with user key, failed to decrypt protection level XML load (dts Password)

Encrypt Sensitive with password:

The Encrypt Sensitive with password setting requires a password in the package and that password will be used to encrypt and decrypt the sensitive information in the package. To fill in the package password click on the button in the

Package password field of the package and provide Password and confirm password. When you open a Package with this setting you will be prompted to enter the password.

Note: The Encrypt Sensitive with Password Setting for the production level property overcomes the limitation of the encrypt sensitive with user key setting by allowing any user to open the package as long as they have the password.

Encrypt All With Password:- The Encrypt All With Password setting used to encrypt the entire content of the SSIS package with the specified password. You specify the package password in the package Property, same as Encrypt Sensitive with password settings. After saving the package you can view the package XML code that is already encrypted in between encrypted data tags in the package XML.

Encrypt All With User Key:- The Encrypt All With User Key setting is used to encrypt the entire contents of SSIS package by using user key this means that only the user who created the package will be able to open it, view or modified it and run it.

Server Storage:- The Server Storage setting for the Production level property allows the packages to retain all sensitive information when you are

Saving the package to SQL Server. SSIS packages saved to SQL Server use the MSDB Data base.

### 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

Year	Category	Jan	Feb	March	April
2008	Bikes	100	200	300	400
2008	Accessories	200	270	300	320
2009	Components	100	120	300	150
2009	Phones and Components	400	800	400	300

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 drag and drop Excel source

Double click on Excel Source to configure it



click New



click Browse



Select Unpivot.xls file and click open



click ok



Select Sheet 1 from the dropdown 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 the below columns to Unpivot them,  
Jan, Feb, March, April, May, June.



Rename pivot key value column Name as -  
Months



Specify sales amount as a Derived or Destination column for all the selected pivoted key values or input columns.



click ok



Make sure that Excel source file is closed.

Drag and 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 the following Excel sheet for source data.

Year	Quarter	Sales Amount
2009	Q1	100
2009	Q2	200
2009	Q3	300
2009	Q4	400

Rename the Excel Pivot.xls

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 Drag and Drop Excel source  
↓

Double click on Excel source to configure it.  
↓

Click New  
↓

Click Browse  
↓

Select pivot.xls and click open.  
↓

Click ok  
↓

Select sheet 1 from the dropdown list  
↓

Drag and drop pivot transformation and  
make a connection from 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 Quarters input column and set,

Pivot Usage - 2

Select Sales Amount input column and set,

Pivot usage - 3

Pivot usage: Pivot usage tells sets 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 rows with the same set key are considered into 1 output row.

2 - The column is pivot column

3 - The values from this column are placed in columns that are created as a result of the pivot.



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

Year, Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>, Q<sub>4</sub>



Copy or Note the Lineage ID of input column  
Year

Select year output column and set,

Pivot key value - Year

Source column - 58 (Lineage ID of Year  
input column)

Select Q<sub>1</sub> and set,

Pivot key value - Q<sub>1</sub>

Source column - 64

Select Q<sub>2</sub> and set,

Pivot key value - Q<sub>2</sub>

Source column - 64

And follow the same process for Q<sub>3</sub> and Q<sub>4</sub>  
output columns ↓

Click Refresh ↓

Click OK ↓

Drag and drop Excel destination ↓

Double click on 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

### Event Handlers on SSIS

It provides the ability to handle any type of event associated with creation of its task and containers. The following list of events are 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 type of Event Handler.

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 the component is completed.

On Pre-Validate: Marks the beginning of the component validation stage. The main purpose of validation is detection of potential problems.

that might prevent execution from completing successfully.

### Steps to configure Event Handler

Open Business Intelligence Development studio

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

In Control flow drag and drop Data Flow task and rename it as Data flow Task Event Handler.

In Data flow drag and drop OLEDB source

Configure it (provide connection manager and select any table from the list)

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

Configure the destination (Provide destination connection manager and create a new table named as Event Handler)

Select Event Handler tab and set,

Executable - Select Data Flow Task Event Handler option

Event Handler - Select on pre-validate option.

Drag and drop Execute SQL Task on to Event Handler section and it will be executed whenever On pre-Validate Event is raised on the selected executable (DFT Event Handler)



Double click on Execute SQL Task to configure it



Provide connection [Local Host, Adventure works]



SQL Command - Truncate Table [Event Handler]



Click Ok



Execute package

Send Mail Task: Send Mail Task is used to send an Email message to the recipient's via SMTP connection.

Steps to configure Send Mail Task

Drag and Drop Send Mail Task (In controlflow)



Double click on send Mail Task to configure it



Provide the following information

SMTP Connection - Select New connection



Provide SMTP server



Name

click ok

From - Provide from email Address

To - Provide To email Address

Cc - Provide Carboncopy email address (if any)

Bcc - Provide Blind Carboncopy email Address  
(if any)

Subject - Text 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

↓  
click ok

↓  
execute package

Example for SMTP connection - Asithxee.Vijayit  
.com

Migrate DTS 2000 package to 2005 dtspack package

Open Business Intelligence Development Studio  
2005 and 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

↓  
Select any package (.dts) and  
click open

↓  
Click Next

Source - File system

Click next

Click finish

Finally the migrated (dtspack) package arrives  
in the specified project under ssis packages

section

↓

Execute DTS

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 and drop execute DTs 2000 package task



Double click on it and set,

storage location - Select structured storage

File

file

- click Browse



Select any package (.dtc) and click open

Package Name

- Specify the package name, which you would like to execute it on SQL Server 2005 / 2008 environment



click ok



Execute package

## Slowly Changing Dimension (SCD):

In SSIS slowly changing Dimension (SCD) is categorised into 3 parts:-

① Change the Attribute (Type I - in terms of Data ware housing)

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 are saved as expired records.

③ Fixed Attributes: (Type III in terms of Dataware housing)

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 SCDiter

↓  
In Control flow drag and drop Data Flow Task and rename it as DFT SCD

↓  
Open SQL Server management studio and run the following SQL statements to create temporary tables for source and destination.

Table for source

Select into Human Resources.Employee  
Address from Human Resources.Employee



In Data flow drag and drop OLEDB source



Double click on OLEDB source to configure it.



Provide connection manager if exists



Select Human Resources.Employee Address from  
the dropdown list



Drag and 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  
Human Resources.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 expired records radio button.



Column to indicate current Record -

Select Nation ID Numbers



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 attribute (update [Human Resource].[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.

Note: The main usage of this transformation would be to select a fixed size subset of data. This subset can be used to test the packages with the limited data set.

For example: If you are running a package against a multi million rows; you could just run the package with 100 rows to check if your process works.

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 drag and 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 and 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 outputs (Sampling and 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 drag and drop OLEDB source and configure it.  
↓

Drag and 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 and 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, Task ID 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 operations in the selected dataset.

Steps to Configure Audit, copy column and  
Character Map Transformations.

Create a New package



In Data Flow drag and 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 and 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 and drop copy column and make a  
connection from Audit to Copy Column



Select Gender from available input columns  
and click ok. ↓

Drag and drop character Map transformation  
and make a connection from Copy column to  
character Map. ↓

Double click on character Map and check Copy  
of Gender available input column and set,

Operation → Lower Case

↓  
click ok

Drag and 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 table and  
rename it as Audit data

↓

click ok twice

#### OLEDB COMMAND TRANSFORMATION:

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

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.dtsr.  
↓

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 ID = 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 Martial Status = 'D' Where employee ID = 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 lookup 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 GTI will contain Rama and Rao. By using Token delimiters.

### Steps to configure Fuzzy Look up

Open Business Intelligence Development Studio

↓  
Create a new package and rename it as Fuzzy  
Lookup.dtsx.

↓  
In control flow drag and drop Data Flow Task

↓  
In Data Flow drag and drop OLEDB Source

↓  
Double click on OLEDB Source to configure it

↓  
Provide Source connection manager. If exists

↓  
Select production.product category table from  
the dropdown list.

↓  
Select columns

↓  
click ok

↓  
Drag and drop Fuzzy Lookup 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

## 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:

Open Business Intelligence Development Studio

↓  
Create a new package and rename it as Import column.dtsx.

↓  
In Control flow drag and drop data flow task and it as DFT, import column.

↓  
In Data flow drag and drop OLEDB source

↓  
Double click on OLEDB source to configure it

↓  
Provide Source connection manager if exists

↓  
Select production, product photo

↓  
Select columns and click ok

↓  
Drag and drop import column transformation and make a connection from source to import column.

Double click on Import column transformation to configure it.

Select Input columns and check thumbnail photo file name. Input column (the input column must be of type DT\_STR or DT\_WSTR)



Select Input and output properties tab and expand input column output and expand output column.



Click add column to add image output column and rename it as image file and note down the ID of image file output column.



Expand Import Column Input and Expand Input Column.



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 written on Microsoft Visual basic .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.dtsa.

↓  
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 foll. Code to add new rows by  
adding AddRow() method.

Output OBuffer.AddRow()

Output OBuffer.SNO = 123

Output OBuffer.SName = "Rama"

Click ok and close vs for applications 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  
checkbox



click test 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  
softwares list from Microsoft Manufacturer:

Select \* from Win32 - Pnppentity where  
Manufacturer = 'Microsoft'

### Deploy project (.sis) 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 systems deployment radio button  
and also check validate packages after installation  
checkbox.



click next



click next



click next



click Finish



Open SSMS



Connect to Integration Services Expand stored Packages and also expand file system you would observe that the morning 8:30 project deployed to server.

### 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 Error:- 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 OnError 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: Indicates that an executable component is about to be launched.
- \* On Pre Validate: 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: cleanup 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 dataflow 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 Eventhandler for package executable]



Double click on Execute SQL task provide  
connection if exists



Provide the following SQL command, to cleanup,  
the data from destination table



Truncate Table Employee details



Click OK



Execute package

## DBA Tasks in SSIS 2005

Database Administrators can create database maintenance plans either by using Maintenance Plan Wizard or by using SSIS designer. By using the Maintenance plan Wizard a very basic maintenance plan can be created for all the system and user databases. However, to create an enhanced workflow it is advised to create maintenance plan using SSIS designer.

Backup Database Task: Backup database task different types of databases backups to be performed; such as Full, 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 refer to the article titled "Database Recovery Models In SQL Server".

Check Database Integrity Task: Check Database Integrity Task can be used to check the allocation and structural integrity of all the user and system tables 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 CHECKDB Statement.

Execute SQL Server Agent Job Task: Execute SQL Server Agent Job Task 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 some Transact SQL queries against database. This task is only available when creating Maintenance Plan 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 faster, but a disadvantage with this mode is that the source database will not be available during Copy and move operation. Also, the person executing the package with this mode must be sysadmin on both source and destination instances.

\* Online: In this mode, the task uses EMO to transfer the database objects to the destination server. In this mode, the database is online during the copy and move operation, but it will take longer as it has to copy each object from the database individually. Some one executing the package with this mode must be either sysadmin or database owner of the specified database.

### 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 Database 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 File(s) - 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 File(s) - 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 these 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 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 Objects Task

The Transfer SQL Server Objects 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 procedures, 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 Task" to the Control Flow. Right click on the task and then Select Edit. In the "Transfer SQL Server

"Objects Task Editor" click on Objects to set the different properties for this task as shown below.

Similar to the way you configured 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 database

Destination Connection - the destination instance

Destination Database - name of the new database

#### Destination

Drop Objects First - Drop selected objects on the target before copy

Include Extended Properties - while copying operation 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 or replace data

Copy Schema - Copy the schema of the objects being copied

Use Collation - Make sure collation of the columns are appropriately set on copied tables

Include Dependent Objects - include all the objects in copy operations 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 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 permission.

#### \* 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 Scripts In Unicode - 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 the source server to the destination server.

Open BIDS. Drag a Transfer Job 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 TransferAllJobs 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 Job 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 Toolbox to the Control Flow. Right click on the task and select Edit to configure the task's 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.

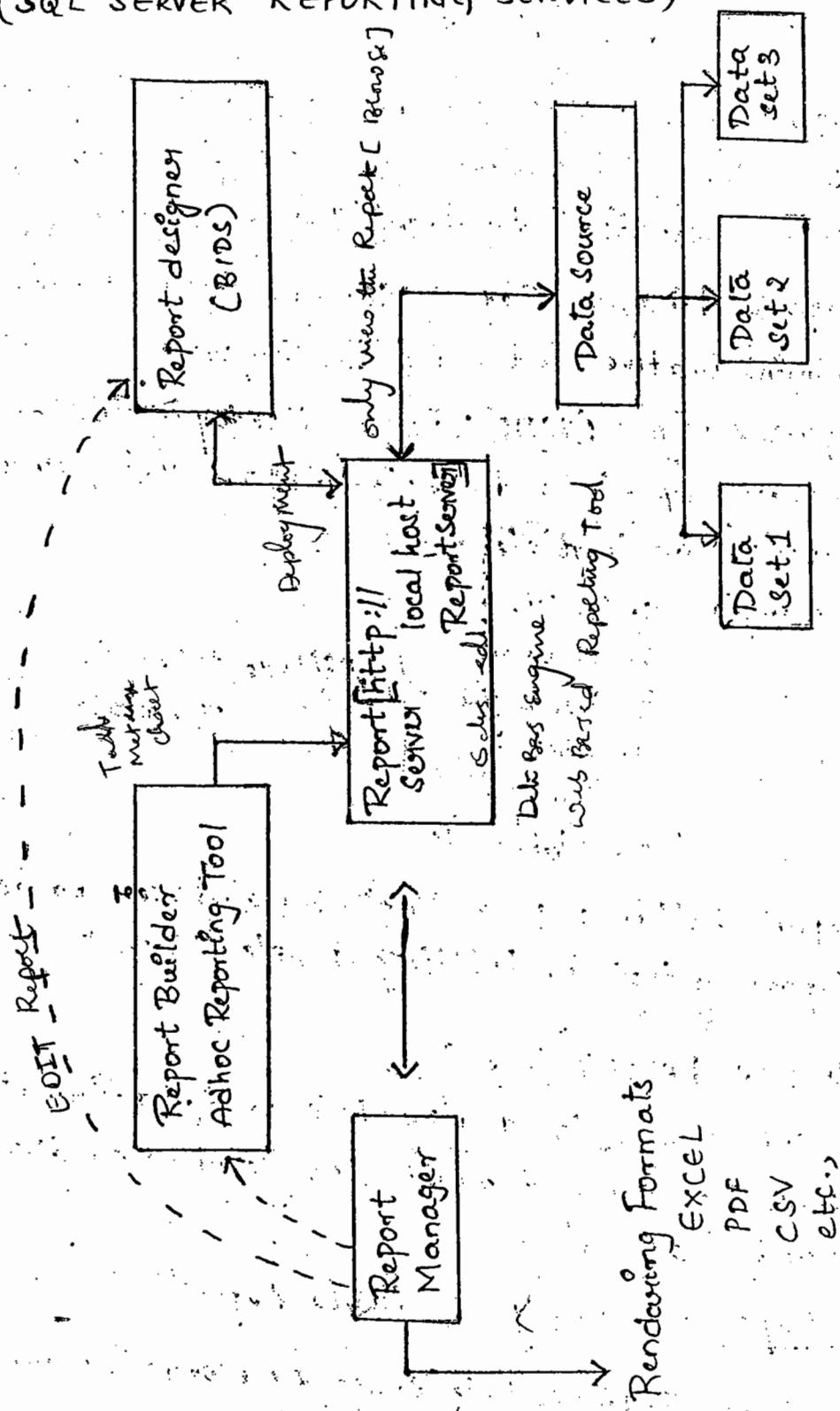
- Login List - this will allow you to select specific logins if you select Selected Logins for Logins To Transfer.
- Database List - this will allow you to select the databases if you select All Logins From Selected Databases for Logins To Transfer.

#### - Options

- \* If Object Exists - if the logins already exist 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.
- \* Copieside - If you set it to True then security identifiers (SIDs) associated with logins are also copied to the destination.

ROL → Report definition language.

### ARCHITECTURE OF SSRS



Data Source : Logical connection between application (SSRS) and database

Report Server : 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 templates.

- Example :
- 1. Table
  - 2. Column
  - 3. Rectangle
  - 4. List
  - 5. Image
  - 6. Matrix etc.
- } Different Chart Template

Report Builder : Is a simplified end user or adhoc reporting tool for building simple Reports

`http://localhost:8080/ReportingServices/ReportBuilder/ReportBuilder` application.

#### 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 :-

Open Business Intelligence Development Studio

Select File Menu



Select New



Select Project



Under Business Intelligence projects Select Report Server project option to build simple and complex Reports on development environment.



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 (localhost) (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 dataset option and set,

Dataset Name -> Dataset product information

Data Source

- Select Datasource product details from dropdown list.

Command Type

- Text (., SQL Script)



click ok



Select Query Designer option and Add Tables icon on the large 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 as Category Name ,  
Product . Product ID ,  
Product . Standard cost ,  
Product . Size ,  
Product . Sell start Date ,  
Product . Reorder point

From Product category Inner Join product  
subcategory on Product Category . Product  
category Id = Product subcategory . Product  
category ID .

Inner Join product on Product . product  
subcategory . Product = subcategory ID  
subcategory ID

Inner Join product cost history ON  
Product . product ID = 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 Items on body of  
the report

↓  
Drag and drop Category Name field onto  
Detail section of the table report [Building  
(or) Mapping the field (or) columns from dataset  
to Report items]

↓  
Map other field as well

↓  
In table Report items select 1 Row and 2 Row

↓  
Press F4 and set

Borderstyle — Solid

↓  
Select 1 Row [Header] and Press F4 for properties  
and set

Background color — Maroon

Text align — Centre

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 parameter 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 - Disc 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 ID's from the selected Dataset)

Dataset — 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 dropdown 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!StandardCost.value, "N 0")  
 ↓ (Zero)

② Expression to check Null Records in the specify 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 centre else left justified. Select size text box press F4 from properties and set Text Align ---  
 $=IIF(Fields!Size.Value > 0, "center", "left")$

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

Select size text box  
↓  
Press F4 for properties  
↓  
set, color  
↓  
 $=IIF(Fields!Size.Value < 30, "Red", IIF(Fields!Size.Value > 30, "Blue", "Green"))$

⑤ Expression to handle negative values for example  
 $(\text{Blue} - 79) \Rightarrow (-79)$  in Red color.

In table Report item add one column and Rename the header as Total and provide the following expression

$=IIF((Fields!StandardCost.Value - Fields!ReorderPoint.Value) < 0,$

replace (Fields! StandardCost.Value - Fields! ReorderPoint.Value), "-","(" ) & ")", (Fields! StandardCost.Value - Fields! ReorderPoint.Value)

Select Totals text box (In Details Section)



Press F4 Properties



Set, color = = IIF (Fields! StandardCost.Value - Fields! ReorderPoint.Value) < 0, "Red", "Black")

⑥ Expression to display Report with alternative Row color.

= IIF (RowNumber(nesting) mod 2, "Silver", "Transparent")

Chart: Chart Report Item is used to display the data graphically as bar 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.



Right click on chart and select properties from context menu.

In General tab,

Title = "Product Details for

the month — " & monthName  
month (Fields! Sell.startDate.value))

Select Data tab:



Dataset Name — Select Dset

Product information from dropdown 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 [Field! Sell start Date.value]) .3) + "—" +

Right (year (Field! 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 — centre



Major tick lines — cross

Minor tick lines — cross



Check Side Margins check box to display the chart items in centre of the plotter area.

In Y-axis,

Title — size

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



Title align - Centre



Major tick line - Cross

Minor tick line - Cross



check Side Margins check box to display the chart items in centre of the plotter area.

In Legend Tab,

Check Show Legend (by default)



Lay out - Table



Check Legend style and set,

font size - 8pt



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 colour and click ok

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

Values — Select standard cast and click edit.



Select point labels Tab



Check show point labels check point,



Data label — = "&" and Fields! Standard  
cast. Value



click ok

Rectangle: Rectangle 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



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 export it to excel or pdf etc.,

## Steps

Select Rect Table (Rectangle 1)



Right click and select properties



set,

PageBreaks — check Insert Before Rectangle chart Box.

[∴ To maintain the page Break after Rect Table Rectangle]



Select preview Tab,



Select 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: A page 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 embeded 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 Embed (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: Text Box 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

= "page" & Globals! Page Number & "of" & Globals! Total pages

= iif (Parameters! Product Category ID . Value = 1,  
"Bikes", "Components") & "Details for the Month —"  
& MonthName (Month (Fields! Sell Start Date . Value))

List Report Item :- It is used to display a set of Report items that are Repeated for each group (or) Row of Data

Steps to Configure List Report Items:

In our previous Report Example, Remove the existing Where clause and add the below to implement multivalue Report.

Where;

(Production, Product Category, Product Category ID IN (@ Product category ID))

Select Report menu and Select Report Parameter option, Set check multivalue 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 — Set product information



click Edit Details Group



Select = Fields ! Product category ID . Value  
in Group on section



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 contains 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 Dataset 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 colour field on to newly added column.



Set border style as solid and set the font size



Go to preview 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 header and 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 Items - Select appropriate text box Name  
↓  
click ok  
↓  
click ok

How to display Group total or subtotal :-

From the above table report example add the below expression at group footer under standard cost header.

$$= \text{Sum}(\text{Fields! Standard cost.value})$$

How to display Grand total 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.

$$= \text{Sum}(\text{Fields! Standard cost.value})$$

How to Create template Reports in SSRS

Design basic Report template [Add a logo on header and display page numbers in page footer]

Rename it as Vijay IT Template.RDL



Go to project path



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

C:\Program Files\

Microsoft Visual Studio.8\

Common7\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 datasource Properties and layout of its base report.

Steps to configure Linked Report:

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



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



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. Since linked report always points to data source and layout of its base or its parent report.

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

Steps to Configure Subreport

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 SubCategory  
ON Product Category . Product Category ID = Product  
SubCategory . 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 . product 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 subreport item



Select Subreport, 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 project)



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 subreport 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 like,

\\Corp\c|Rama

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. ReportServerTempDB

UNC - unified naming conventions

1. Report Server: 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: The 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 role  
↓

Select properties option



Select Security page



Click Edit Item security



Click ok



Click New Role Assignment



Provide user Name 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 datasource wizard, click next



Select the datasource if exists



Click Next, Rename the datasource as product  
Details



Click Finish



Select Data source view



Right click and select Add new data source view



Click Next



Select product Details datasource and click Next.



Select production.Product Subcategory table and  
click Add related tables.



Click Next



Rename the datasource 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 three



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 application.

Report Builder will load with the new report dialogue box and show 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 (<sup>Product</sup> 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.

Interactive 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! CategoryName.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

SMDL semantic model definition language

### 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 Parameters properties :

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

For example: While implementing Subreport hide the parameter in the embedded Report

Multivalue: If you want to display multiple elections then multivalue parameter is used. In order to implement multivalue parameter Report. IN 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 type]

Cascading Reports: In this Report one parameter is depending on another parameter.

For example: Country parameter and state parameter.

If the user select any country from the Country Parameter then state parameter can be bindable and displays (or) populate corresponding states.

#### Steps to Configure Cascading Report

Open Business Intelligence Development Studio and create a new Report Item and Rename it as Cascading.rdl.

In Databab create New Dataset and Rename it as Category Details.

Provide the following SQL Query  
Select productcategory ID, Name as category name

from production.product category.



click ok



Select New Dataset and set

Name : Sub Category

Data Source : Disc 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 : Disc product Category

Command type : Text

Query string : Select \* from production.product  
Where

Product Subcategory ID = @product Subcategory  
ID.

Select Report menu and select 'Report Parameters'



Select product category ID parameter and set In-  
available values; Check  from Query radio button  
and

DataSource : Category details

Value field : Product Category ID

Label field : Category Name  
↓

Select product Sub category ID parameter and set  
Inavailable values:

Check  from Query radio button and

DataSource : Sub category

Value field : Product Subcategory 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 off 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 analyse 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. Dimensions comes from dimension table. measures comes from Fact table.

Dimension Table: It contains hierarchical data by which you would 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 tables.

Fact table: It contains basic information that you want to summarise.

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 analyse (or) to evaluate the performance of organisation.

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) Invok a Wizard.

## Create a new analysis Service project:

Open Business Intelligence Development Studio



Select File Menu



New



Project



Select project Analysis Services



change project name and location



click ok

## Define a Data Source:

In Solution Explorer select Data Source



Right click and select 'New Data Source'



Welcome to the Data Source wizard



click Next



click New to create a new Connection Manager  
and set,

Server name - Local host

Database name - Adventure works DW



click ok



click Next



check default Radio button



click next



Data Source Name - product details



click Finish

### Define a Datasource View:

In solution explorer select Data Source View



Right click and select New Data Source View



In Welcome to the Datasource View wizard



click Next



Select the available Relational Data Source  
[Product details] ↓

Click Next

Sales

Fact Resellers, 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 datasource 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 Datasource View named as product information [Select the data source view, that will provide the data for the cube].



Click Next



Click Next



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 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 columns (or) Rows (or) Filter area.

## 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 SQL and MDX. MDX provides various ways 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  
MEMBER exp 3 and empty } calculated columns.  
Select

{ - (Set operator)  
<axis - specification 1>, → (column to be displayed  
<axis - specification 2>, on column)  
} ON columns  
<axis specification > on Rows  
FROM [cube Name]  
Where [slicer]

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

Members: Select dimension attributes 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 members 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

for example: Top count ([DimProduct].[English Product Name].allmembers, 5) on rows

Bottom Count: It is used to display a specified no. of Records from the Bottom.

for example: Bottom count ([DimProduct].[English Product Name].allmembers, 5)

Comma Operator: It is used to construct a set by enumerating tuples. for example:

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.

For example:

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 rows.

WITH Operator: It is used to specify calculated measures. for ex:-

WITH  
Member [Measures].[total amount]  
as  
[Measures].[sales amount]\*  
[Measures].[Order Quantity]

NON EMPTY: It is used to Remove null Records either from columns (or) Rows.

for ex: Select non empty  
{[measures].[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

{ [measures]. [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 Returns 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,

X non empty

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

The above MDX Query New measures are calculated  
[measures].[total amount] and its formatted all  
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 Information]

Where [Fact Internet Sales - Due Date].[Time Key].

+ [1]: [Fact Internet 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 15<sup>th</sup>]

Format-string: It is used to format the measures in WITH Operator [format the calculated measures]  
for ex:-

Format-string = "\$ ##,###+.##"

" " n = "currency"

" " I = "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

drop down list.

Expression - Define the following Expression by drag and drop the measure from the measure 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 be available.

### KPIs (Key Performance Indicators)

In Analysis Services, Represents 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 KPI's 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.

[Measures].[Order Quantity]

Goal Expression: Here our new created measure will come into picture. We'll 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.

[Measures].[Total Product Cost]

State: It's checks the state of KPI at any point.

State's MDX Expression should return value between

-1 and +1

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

$$\text{IIF}(\text{KPI VALUE}\left(["\text{Measures}].[\text{Order Quantity}]\right) > \text{KPI GOAL}\left(["\text{Measures}].[\text{Total Product Cost}]\right), -1, 1)$$

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:

Open BIDS  
↓

In Analysis Service Create a valid cube and deploy it to analysis Service Server.

Open BIDS Create Report Server project  
↓

In Solution Explorer set Shared Data Sources.  
↓

Right click and select Add new Data Source and Set,  
Name - Dsrc product sales details

Type - Select Microsoft SQL Server Analysis Services option from dropdown list  
↓

Check Edit  
↓

Server Name - Local host (or) Actual Server name  
(or) IP Address

Database Name - Cube Name (Product Details  
morning 8:30)

↓  
Test connection

↓  
click ok thrice

↓  
Select Report option

↓  
Right click and select Add

↓  
New Item

↓  
Select Report template. Rename it as Product  
details from cube. RDl.

↓  
click add

↓  
In Data tab add new Dataset 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

[Fact and dimension tables]

From Dim Product,

[English ProductName],

[Class],

[color],

[DealersPrice],

[List Price]

From measures,

[Measures].[Fright]

[measures].[Unit Price]

Press Alt + Ctrl + D for Dataset



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 adventureworks DW database.

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 tables Dimproduct, Dimproduct SubCategory , DimProductCategory ; DimCustomer and FactInternetSales.
4. Create a cube with default settings but selective measures as Order Quantity , UnitPrice , Sales Amount and Internet Sales Count. Give Cube name as DemoCube.
5. Go to the properties of the project (ProductDetails) > Select Deployment tab and enter server name. (If you are working with default instance, you can keep the name provided by default i.e localhost)
6. After successful deployment, Open SSMS and connect to Analysis Services to work with MDX queries.
7. Create a new query, connect ProductDetails database  
Enter following MDX query:

```
SELECT FROM DemoCube
```

In this case default member for Order Quantity is displayed as 60938

Enter the query as:

```
SELECT [Measures].[Order Quantity] ON 0  
      FROM DemoCube
```

Will also give the same result. (0 can also be replaced by COLUMNS) which is the axis.

8. Let us add one more measure 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 DemoCube.
```

OR

```
SELECT [Product].[Product Category Key].&[1] ON  
      COLUMNS FROM DemoCube.
```

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

```
SELECT [Product].[Product Category Key].&[1]  
      ON COLUMNS,  
      [Customer].[Customer Key].MEMBERS ON ROWS  
      FROM DemoCube
```

With this we get Sales Amount for Bikes for all different customers

This can be achieved by using ordinals as 0 and 1 instead of 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 values)
12. Let us see how to add a calculated member  
WITH MEMBER Measures.[calculated Member] 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 follows  
SELECT NON EMPTY {Measures.[Calculated Member],  
[Measures].[Sales Amount]} ON COLUMNS,  
NON EMPTY [Product].[Product Category Key].CHILDREN  
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: Simply using bottomcount will return customers with null sales. You will have to combine it with NONEMPTY or FILTER.

```
SELECT {[Measures].[Internet Sales Amount]} ON COLUMNS,  
BOTTOMCOUNT(  
NONEMPTY (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 "OrderBy" and "OrderByAttribute". 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  
,[Date].[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 userfunction 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].[TimeName] AS
IIF ([Date].[calendar].Level IS [Date].[calendar].[Calendar Quarter], 'Qtr', 'Not Qtr')
SELECT [Measures].[TimeName] 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 months with sales

A: Function Count(, Exclude Empty) counts number of non empty set members. So if we crossjoin 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
COUNT(CROSSJOIN([Measures].[Sales Amount]
, DESCENDANTS({[Date].[calendar].[Calendar Year].&
[2003]:,
[Date].[calendar].[Calendar Year].&[2004]}, {[Date].[calendar].[Month]}))
, Exclude Empty
)

```

```
SELECT {[Measures].[Sales Amount], [Measures].[Months with NonZeroSales]} ON 0  
, [Product].[Product Model Lines].[Product Line].Members  
ON 1  
FROM [Adventure Works]  
WHERE ([Date].[Calendar].[Calendar Year].[&[2003]:  
[Date].[Calendar].[Calendar Year].[&[2004]])
```

Q: How can I setup default dimension member in calculation script?

A: You can use ALTER CUBE statement. Syntax:

```
ALTER CUBE CurrentCube | YourCubeName UPDATE  
DIMENSION,  
DEFAULT_MEMBER = ";
```

Q: I would like to create MDX calculated measure that instead of summing children amounts, use 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. Let's first see what data values are at Calendar Semester level:

```
SELECT {[Measures].[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::MyMonthCount AS  
    SUM(YTD([Date].[Calendar]), COUNT([Date].[Month  
    of Year])))  
MEMBER Measures::MyYTDAVG AS Measures.  
    MyYTD / Measures.MyMonthCount
```

```
SELECT {Measures.MyYTD, Measures.MyMonth  
    Count, [Measures].[Internet Sales Amount],  
    Measures.MyYTDAVG} ON 0,  
    [Date].[Calendar].[Month] ON 1  
FROM [Adventure Works]  
WHERE ([Date].[Month of Year].&[7])
```

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 Crossjoin 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 Line].[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 Newsgroup.

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].[CaseResult 1] AS  
Aggregate (Filter ([Product].[category].[All]).  
children,  
[Product].[category].currentmember.properties  
("key") < "3")  
MEMBER [Product].[category].[CaseResult 2] AS  
Aggregate (Filter ([Product].[category].[All]).  
children,  
[Product].[category].currentmember.properties  
("key") = "3")  
MEMBER [Product].[category].[CaseResult 3] AS  
Aggregate (Filter ([Product].[category].[All]).  
children,  
[Product].[category].currentmember.properties  
("key") > "3")  
SELECT NON EMPTY {[Measure].[OrderCount]}  
ON COLUMNS  
, {[Product].[category].[CaseResult 1], [Product].[category].[CaseResult 2], [Product].[category].[CaseResult 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 day?

A: You can use Filter function and Compare member keys using Properties function:

```
SELECT {[Measures].[Internet Order Count]} ON 0  
, FILTER(NonEmpty(CrossJoin([Ship Date].[Date].  
Children, [Delivery Date].[Date].Children  
)  
, [Ship Date].[Date].CurrentMember.Properties  
(`key'))  
= [Delivery 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.ProductKey AS [Product].[Product  
Categories].CurrentMember.Member_key  
SELECT {[Measures.ProductKey]} ON axis(0),  
[Product].[Product Categories].Member ON axis(1)  
FROM [Adventure Works]
```

Q: How do I create a Rolling 12 Months Accumulated Sum (InternetSalesAmtR12Acc) that can show a trend without seasonal variations?

A: Here is query example

```
WITH MEMBER [Measures].[InternetSalesAmtYTD] AS  
SUM(YTD([Date].[calendar].CurrentMember),  
[Measures].[Internet Sales Amount]),  
Format_String = "# ## # ## # ## #"  
  
MEMBER [Measures].[Internet Sales AmtPPYTD] AS  
SUM(YTD(ParallelPeriod([Date].[calendar].[calendar Year], 1, [Date].[calendar].CurrentMember)),  
[Measures].[Internet Sales Amount]), Format_String  
= "# ## # ## # ##"  
  
MEMBER [Measures].[InternetSalesAmtPY] AS  
SUM(Ancestor(ParallelPeriod([Date].[calendar].[calendar Year], 1, [Date].[calendar].CurrentMember),  
[Date].[calendar].[calendar Year]),  
[Measures].[Internet Sales Amount]), Format_String  
= "# ## # ## # ##"  
  
MEMBER [Measures].[InternetSalesAmtR12Acc] AS  
([Measures].[InternetSalesAmtYTD] + [Measures].[InternetSalesAmtPY]) - [Measures].[InternetSalesAmtPPYTD]
```

Select {[Measures].[Internet Sales Amount]}, Measures.  
[Internet Sales Amt YTD],  
[Measures].[Internet Sales Amt PYPYTD], [Measures].[Internet Sales Amt R 12 Acc]} 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 CURRENTCUBE 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(NONEMPTY([Customer].[Customer Geography]).  
[Customer].MEMBERS  
*[Date].[Date].[Date].MEMBERS
```

, [Measures]. [Internet Sales Amount])

AS MYSET

+ MYSET. CURRENTORDINAL = 1 or

NOT (MYSET. CURRENT. ITEM(0) IS MYSET. ITEM  
(MYSET. CURRENTORDINAL - 2). ITEM(0)))

MEMBER [Measures]. [customers w/ FirstSales] AS  
COUNT (NonEmpty ([First Sales], [Measures]. [Internet  
Sales Amount])),

FORMAT\_STRING = '#,##'

SELECT { [Measures]. [Internet Sales Amount],  
[Measures]. [customers w/ FirstSales] } ON 0,  
[Product]. [Product Categories]. [Category] ON 1  
FROM [Adventure Works]

WHERE ({ [Date]. [Calendar]. [Calendar Year]. &  
[2002], [Date]. [Calendar]. [Calendar Year]. &  
[2003] }, [Customer]. [Customer Geography]. [City].  
& [Calgary] & [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 Date 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 level, ratio will be calculated to level [All]:

WITH MEMBER [Measures].[Order Count Ratio To Parent] AS  
TIF( ([Measures].[Order Count], [Date].[Calendar].  
Current Member. Parent) = 0,  
, NULL  
, [Measures].[Order Count]  
/ ([Measures].[Order Count], [Date].[Calendar].  
Current Member. Parent)  
)  
, FORMAT\_STRING = "Percent"

SELECT {[Measures].[Order Count]}, [Measures].[  
Order Count Ratio To Parent]} ON 0,  
. {DESCENDANTS ([Date].[Calendar].[All Periods],  
. 1), [Date].[Calendar].[All Periods]} ON 1  
FROM [Adventure Works]

#### MDX Date Functions

How do you get Last month in the time dimension

SELECT Closing Period ([Date].[Calendar].[Month],  
[Date].[Calendar].Default Member) ON 0 FROM  
[Sales Summary] WHERE ([Measures].[Sales Amount])

Need a MDX query that returns list of months from start of year up to specified month.

```
SELECT YTD([Date].[calendar].[Month], & [2003]&[8])  
ON 0 FROM [Sales Summary];
```

I Need an MDX statement to show the first day of the last month in the cube.

```
SELECT Opening Period ([Date].[calendar].[Date],  
closing Period ([Date].[calendar].[Month],[Date].[  
calendar].Default Member)) ON 0 FROM [Sales  
Summary]
```

How in the report can I order date dimension members in descending order?

```
SELECT {[Measures].[Retail Order Quantity]} ON 0,  
ORDER (Tail([Date].[calendar].[calendar year].  
Members,3),[Date].[calendar].CurrentMember.  
Member_key,DESC) ON 1 FROM [Adventure Works]
```

I Need an MDX statement to get the last Month loaded into a cube. SELECT

```
Closing Period ([Date].[calendar].[Month],[Date].[  
calendar].Default Member) ON 0 FROM [Sales  
Summary];
```

I Need an MDX statement to get the first month of the last year loaded into a cube. SELECT  
Opening Period ([Date].[calendar].[Month],

Closing Period ([Date].[Calendar].[Calendar Year],  
[Date].[Calendar].DefaultMember) ) ON 0 FROM  
[Sales Summary];

How do you write MDX query that uses execution  
date / time as a parameter?

```
SELECT {[Measures].[Internet Order Count]} ON 0,  
{StrToMember("[Date].[Date].[" + Format(  
now(), "MMMM dd, yyyy") + "]")} ON 1  
FROM [Direct Sales];
```

How to create calculated member for AVG Sales  
over last 3 year based on NOW()?

```
CREATE MEMBER CurrentCube.Measures.[AVG  
3 Years] AS Avg ( {ParallelPeriod([Date].[  
Date].[Date Yr], 3, StrToMember("[Date].[  
Date].[" + Format(now(), "yyyyMMdd") + "  
"]")}, [Measures].[Sales Qty]);
```



closing period ([Date].[Calendar].[Calendar Year],  
[Date].[Calendar].DefaultMember) ) ON 0 FROM  
[Sales Summary];

How do you write MDX query that uses execution date/time as a parameter?

```
SELECT {[Measures].[Internet Order Count]} ON 0,  
{StrToMember("[Date].[Date].[" + Format(Now(), "MMMM dd, yyyy") + "]")} ON 1  
FROM [Direct Sales];
```

How to create calculated member for Avg sales over last 3 year based on Now()?

```
CREATE MEMBER CurrentCube.Measures.[Avg  
3 Years] AS Avg ( {ParallelPeriod([Date].[  
Date]. [Date Yr], 3, StrToMember("[Date].[  
Date]. & [" + Format(Now(), "yyyyMMdd") + "  
"]"))}, [Measures].[Sales Qty]);
```

① BIDS

② SSMS → Connect to IS → right click → Run package

③ OTEXEC.EXE

④ OTEXECUT.EXE

⑤ SSMS → select source agent → Tasks