1.1 - Schemas

**SSAS** enables analysis in real time, a process called online analytical processing (OLAP).

**Snowflake schema** – star schema with normalised dimensions. Doesn’t perform as well because more joins. Is an appropriate when dimensions share the same attribute.

Star schemas are joined through shard dimensions

1.2 – Designing dimensions

**Types of fields in a Dimension**

|  |  |
| --- | --- |
| Keys |  |
| Name columns | Used for names of entities |
| Attributes | Used for pivoting in analyses. Continuous values should be discretized according to business rules. |
| Member properties | Used for labels in a report such as company name, address etc. Not pivoted on |
| Lineage columns | Used for auditing, and never exposed to end users |

**Hierarchies**

Dimension can have natural hierarchies that provide drill-down path. Hierarchies have levels which have members eg Jan, Feb, March etc.

In a snowflake schema, hierarchies are implemented as joined look-up tables.

In a star schema, they are implemented in terms of attributes

**Slowly Changing Dimensions**

|  |  |
| --- | --- |
| Type 1 SCD | update the single row. Used when there is no concept of history for that attribute |
| Type 2 SCD | create a new row. Used when there is a concept of history eg someone moves from one address to another. If doing this then need a surrogate key because there will be duplicates. |
| Type 3 SCD | Hold previous values in additional columns |

1.3 – Designing facts

**Types of fields in a Fact table**

|  |  |
| --- | --- |
| Measure | normally numerical, can be aggregated and what is being measured eg sales value. |
| Foreign keys to dimensions | all foreign keys together usually uniquely identify a row |
| Linage |  |
| Business keys |  |

**Additivity of Measures**

|  |  |
| --- | --- |
| Additive | measure can be summed across any dimension eg sales amount |
| Non-Additive | can’t be summed across any dimension. Normally averaged eg unit price |
| Semi-Additive | can be summed across any dimension except time. Eg balance |

**Related fact tables**

Fact tables may be related and so one may be on ‘many’ side. This won’t work for SSAS so can put in an intermediary Dimension table. Not derived from source, is a construct of the data warehouse.

2.1 – Implement dimension and fact tables

**Properties of data warehouse**

Simple recovery model

Prevent auto-shrinking and auto-growing as caused fragmentation. Allow enough space in log and data files so they don’t have to auto-grow.

**Calculate space** required for data file by extrapolating from a small period and add 25% for rebuilding indexes without fragmentation.

For log, find your biggest transaction

**Files Groups** have one group per partition and have one file per physical disk.

**Sequence** – are independently created from a table and can be used by multiple tables. Can perform better and offer more functionality

**Computed columns** – can be used to discretized columns

**Fact tables – foreign keys** – can be disabled for loading into but ETL should be robust enough to error if violations occur. Or it can add a row when it doesn’t exist. This is called **inferred member**

2.2 – Managing the Performance of a Data Warehouse

Every table should have a **Clustered index** – means table is stored as a B-Tree rather than heap. Advantages

* Index (as so table) can be rebuilt to defrag
* You can move table to a new file groups without having to drop and recreate the whole table

ALTER table DimCustomerJoe

drop constraint PK\_DimCustomer\_CustomerKeyJoe

ALTER table DimCustomerJoe add

constraint PK\_DimCustomer\_CustomerKeyJoe PRIMARY KEY CLUSTERED (CustomerKey ASC)

on [Primary]

* By default, a non-clustered index will be created which will probably be bigger than a clustered index.
* Can create covered queries that run only off the index.

Key for a clustered index should be short. Clustered index is part of non-clustered indexes. If it is long, then non-clustered indexes will be bigger and not perform as well. So opt for auto-incrementing integer.

Non-clustered index, won’t be used much on dimension attributes. Could be used on name columns and member property columns, used in parametrised reports eg company name.

**Filtered indexes** – index with where clause. Use for rare values in a column. Common can be handled with a table scan.

Do not index foreign keys of fact tables unless queries performing merge or nested loops joins.

**Indexed Views -** Putting index on a view will materialize it. Will be automatically maintained. Should be dropped and recreated during ETL.

Window functions - Make sure running-total queries use “window” functions

**Compression** – useful when data is mostly read and rarely updated

Types of compression

|  |  |
| --- | --- |
| Row compression | Storing fixed data type columns in a variable-length format. Small impact on CPU so good for OLTP as well |
| Page compression | Includes row compression but also   * Prefix compression - stores repeated prefixes of values from a single column in a special compression information (CI) * Dictionary compression – stored repeated values from any column on the page to the CI |
| Unicode compression | Stores Unicode strings as single bytes where able to |

**Columnstore index**

Stores indexes by column rather than row. Each column stored in a different page. Can be more efficient on wide tables because doesn’t have to access the whole row. Not good for very selective queries that touch a few rows.  Not suitable for OLTP workloads.

Limitations

* One columnstore index per table
* Table becomes read-only. Has to be dropped to update
* Must be aligned to partition

**Batch mode processing**

Retrieves data in batches rather than individual rows. Lowers CPU. Useful for bitmap filtered hash join and scan operators

2.3 – Loading and Auditing Loads

**Partitioning**

Tables and indexes can be partitioned. Partitions can be stored in different file groups on different drives thus enabling parallel querying

|  |  |
| --- | --- |
| Partition function | An object that maps rows to partitions by using values from specific columns (partitioned columns) |
| Partition scheme | Maps partitions to filegroups |
| Aligned index | indexes that have the same portioning as base table |
| Partition elimination | query optimizer only access partitions it needs to |
| Partition switching | This is a process that switches a block of data from one table or partition to another table or partition |

If you want to switch content from a nonpartitioned table to a partition of a partitioned table, what conditions must the nonpartitioned table meet?

* It must have the same constraints as the partitioned table.
* It must have the same compression as the partitioned table.
* It must have a check constraint on the partitioning column that guarantees that all of the data goes to exactly one partition of the partitioned table.
* It must have the same indexes as the partitioned table.

**Data Linage**

Many in-built functions in SSIS and T-SQL for logging linage eg

1. APP\_NAME()
2. DATABASE\_PRINCIPAL\_ID()
3. USER\_NAME()
4. SUSER\_SID()
5. SUSER\_SNAME()
6. CONNECTIONPROPERTY('net\_transport')
7. CONNECTIONPROPERTY('client\_net\_address')
8. CURRENT\_TIMESTAMP
9. @@ROWCOUNT

3.1 – Using the SQL Server Import and Export Wizard

Use input/export wizard for simple data transfer ie where data is not being transformed

3.2 – Developing SSIS Packages in SSDT

**Things SSIS can do**

* Access remote locations eg FTP
* Call processes that are external to SSIS
* SQL Server Administration operations eg backups, integrity checks
* Operating system inspection - Windows Management Instrumentation (WMI) data is accessible to SSIS
* Send mail
* SQL Server Analysis Services processing
* Data profiling
* Data mining queries

4.1 – Connection Managers

Types

|  |  |
| --- | --- |
| ADO connection manager | Connect to active X. Used for backward compatibility so don’t use |
| ADO.NET connection manager | Connects to data stores using .net. Can be used for SQL Server |
| Analysis Services connection manager | connects to SSAS database |
| File connection manager/Multiple Files connection manager | connects to SSIS data files |
| Flat File connection manager/Multiple Flat Files connection manager | connects to flat files |
| FTP connection manager | SFTP not supported and windows integrated authentication not supported |
| HTTP connection manager | Connects to web service. Again basic authentication is supported not windows integrated |
| MSMQ connection manager | Access to Microsoft Message Queuing (MSMQ) message queues |
| ODBC connection manager | Provides access to database management systems that use the Open Database Connectivity (ODBC) specification. |
| OLE DB connection manager | Access to database management systems that use the OLE DB provider include SQL Server |
| SMO connection manager | Access to SQL Management Object (SMO) servers. Used for maintenance tasks |
| SMTP connection manager | The SMTP connection manager provides access to Simple Mail Transfer Protocol (SMTP) servers and is used by the Send Mail task to send email messages. |
| SQL Server Compact Edition connection manager | Used to access SQL Server Compact Edition (whatever that is) |
| WMI connection manager | Connect to Windows Management Instrumentation ie OS stuff |

ADO should be used when using parametrized queries in Execute SQL task because

* can use name of parameter rather than question mark

EG select count(1) from [dbo].[DimCustomer] where firstname = @FirstName

* Supports additional datatypes eg VARCHAR(MAX) and VARBINARY(MAX) and Binary

At design time, connection managers are used by the SSIS developer to configure a connection to a data source.

At run time, connection managers are used by the SSIS engine to establish connections to data sources.

Not every provider exists in both 64-bit and 32-bit versions.

4.2 – Control Flow Tasks and Containers

**Complex Data Movement** has these features

|  |  |
| --- | --- |
| Data cleansing |  |
| Data normalization | conversion of complex data types into primitive data types eg by parsing XML |
| Data type conversion |  |
| Data translation | eg changing F to Female |
| Data validation | This is the verification and/or application of business rules against individual values |
| Data calculation and data aggregation | For example, “net price” and “tax” exist at the source, but “price including tax” is expected at the destination |
| Data pivoting and data unpivoting | Source data might need to be restructured or reorganized in order to comply with the destination data model (for example, data in the entry-attribute-value (EAV) might need to be restructured into columns or viceversa). |

**Control flow tasks**

|  |  |
| --- | --- |
| **Data Preparation Tasks** | |
| File System task | operations on task eg copying moving etc. |
| FTP task | typically used to download from or upload to an FTP |
| Web Service task | access web services |
| XML task | XML manipulation, validation and data retrieval |
| Data Profiling task | used for determining data quality and cleansing |
| **Workflow Tasks** | |
| Execute Package task | call another SSIS package |
| Execute Process task | execute process outside SQL |
| Message Queue task | used to send and receive messages to and from Microsoft Message Queuing (MSMQ) queues on the local server |
| Send Mail task | allows the sending of email messages from SSIS packages by using the Simple Mail Transfer Protocol (SMTP). |
| WMI Data Reader task | provides access to Windows Management Instrumentation (WMI) data which gives access to information about the environment |
| WMI Event Watcher task | Typically, the WMI Event Watcher task would be used to trace events in the environment, and based on them to control the execution of SSIS processes |
| Expression Task | Typically, the Expression task is used to assign values to variables without the overhead of using the Script task for the same purpose |
| CDC Control task | This task controls the life cycle of SSIS packages that rely on the Change Data Capture (CDC) functionality. |
| **Data Movement Tasks** | |
| Bulk Insert task | Task loads data from text files to SQL server database tables. No transformation options so is loaded quickly. Requires Sysadmin rights. |
| Execute SQL task |  |
| Data flow task | allows complex data movements |
| **SQL Server Administration Tasks** | |
| Transfer Database task | online – slower but database is available during backup, offline – faster but database is not available |
| Transfer Error Messages task | Use this task to transfer user-defined error messages from one SQL Server instance to another |
| Transfer Jobs task | Use this task to transfer SQL Server Agent Jobs from one SQL Server instance to another |
| Transfer Logins task | Use this task to transfer SQL Server logins from one SQL Server instance to another |
| Transfer Master Stored Procedures task | Use this task to transfer user-defined stored procedures (owned by dbo) from the master database of one SQL Server instance to the master database on another SQL Server |
| Transfer SQL Server Objects task | Use this task to transfer objects from one SQL Server instance to another |
| **SQL Server Maintenance Tasks** | |
| Back Up Database task |  |
| Check Database Integrity task | automate data and index page integrity checks |
| Execute SQL Server Agent Job task |  |
| Execute T-SQLStatement task | more basic version of “Execute SQL task”, can’t send parameters for eg |
| History Cleanup task | Use this task in your maintenance plan to automate the purging of historical data about backups and restore operations |
| Maintenance Cleanup task | Use this task in your maintenance plan to automate the removal of files left over by maintenance plan execution |
| Notify Operator task | Use this task in your maintenance plan to send email messages to SQL Server Agent operators |
| Rebuild Index task |  |
| Reorganize Index task |  |
| Shrink Database task | avoid shrink because might cause fragmentation |
| Update Statistics task |  |
| Analysis Services Tasks | |
| Analysis Services Execute DDL task | This task provides access to SSAS databases for creating, modifying, and deleting multidimensional objects or data mining models |
| Analysis Services Processing task | This task provides access to SSAS databases to process multidimensional objects, tabular models, or data mining models |
| Data Mining Query task | This task provides access to Data Mining models, using queries to retrieve the data from the mining model and load it into a table in the destination relational database |
| script Task | Used to provide functionality not available from the inbuild features |

**Custom Tasks**

Use instead of script task if reusability is required eg different SSIS packages uses same script. Can be developed independently of SSIS.

**Containers**

|  |  |
| --- | --- |
| For-Loop Container | executes control flow until evaluation condition is false eg using a variable |
| Foreach Loop Container | executes control flow over an enumerate eg all files in a folder or number of rows in a table. Assign enumerator value to a variable to make available to tasks inside the loop |
| Sequence Container | Controls structure and determine precedents |

4.2 – Precedence Constraints

|  |  |
| --- | --- |
| Success constraint | Allows following task to begin when successfully completes |
| Failure constraint | Allows following task to begin when errors |
| Completion | Always allows following task to be begin |

5.1 – Defining Data Sources and Destinations

Types of Data Flow Source Adapters

|  |  |
| --- | --- |
| ADO.NET source | ADO.NET provider |
| CDC source | ADO.NET provider |
| Excel source |  |
| Flat File source | Used a Flat File connection manager |
| ODBC source |  |
| OLE DB source | OLE DB providers |
| Raw File source | reads data from a native SSIS data file |
| XML source | requires an XML schema |

Types of Data Flow Destination Adapters

|  |  |
| --- | --- |
| ADO.NET destination | ADO.NET provider |
| Data Mining Model Training |  |
| DataReader destination | Lets you pass data in a ADO.NET recordset that can be programmatically referenced. |
| Dimension Processing | Loads and processes a SQL Server Analysis Services dimension. |
| Excel destination |  |
| Flat File destination | Used a Flat File connection manager |
| ODBC destination |  |
| OLE DB destination | OLE DB providers |
| Partition destination | Allows an SSAS partition to be processed directly from data flowing through the data flow. |
| Raw File destaint | native SSIS data file |
| Recordset destination | Creates recordset in object variable |
| SQL Server Compact destination | Lets you send data to a mobile device running SQL Mobile. |
| SQL Server destination | Provides a high-speed destination specific to a local SQL Server database. |

**Fast Parse -** Very fast loading of flat files only works on certain data types. Is specified as column level.

**Support for delimited files with varying numbers of columns** eg ragged-right

**Things to remember**

* Use appropriate data source or data destination adapters.
* Always extract only the columns you need.
* Use Fast Load or Batch mode when inserting data by using an ODBC or OLE DB destination adapter.
* Use a Raw File destination if you have to temporarily store data to be used by SSIS later.

5.2 – Working with Data Flow Transformations

**Blocking in transformations**

|  |  |
| --- | --- |
| non-blocking transformations | each row goes through transformation without waiting |
| partial-blocking transformation | waits until a sufficient number of rows have been stored then proceeds |
| blocking transformation | all rows must read into the transformation before they are be processed |

**Logical Row-Level Transformations**

Performs operations at row level – use eg calculated columns from multiple sources and conversion of data types

|  |  |  |
| --- | --- | --- |
| Audit | Adds additional columns to each row based on system package variables such as ExecutionStartTime and PackageName | N |
| Cache Transform | Allows you to write data to a cache with the Cache connection manager. The data can then be used by the Lookup transformation. This is useful if you are using multiple Lookup transformations against the same data, because SSIS will cache the needed data only once and not for each Lookup component. | N |
| Character Map | Performs common text operations such as Uppercase and allows advanced linguistic bit-conversion operations. | N |
| Copy Column |  | N |
| Data Conversion |  | N |
| Derived Column | Can be used to create new columns or replace existing | N |
| Export Column | Exports binary large objects (BLOB) columns, one row at a  time, to a file. | N |
| Import Column | Loads binary files such as images into the pipeline; intended for a BLOB data type destination. | N |
| Row Count | Tracks the number of rows that flow through the transformation and stores the number in a package variable after the final row | N |

**Useful expressions for dealing with nulls**

REPLACENULL (col1, 0) – If Col1 is null then replace with 0

NULL(DT\_I4) – puts a null into a 4 byte integer column

**Multi-Input and Multi-Output Transformations**

|  |  |  |
| --- | --- | --- |
| CDC Splitter | Splits a single flow of changed rows from the CDC source  component into multiple data flows based on the type of the  source data change | N |
| Conditional Split | Rereoutes rows according to condition | N |
| Lookup | Performs a lookup operation between a current row and an external dataset on one or more columns. | N |
| Merge | Combines the rows of two similar sorted inputs, one on top of the other, based on a defined sort key | P |
| Merge Join | Joins the rows of two sorted inputs based on a defined join column or columns, adding columns from each source. Works like T-SQL join but each source has to be sorted on joining columns | P |
| Multicast | Generates one or mode identical outputs, from which every row is sent out every output. This transformation creates a logical copy of the data | M |
| Union all | Combines the rows of two similar sorted inputs, one on top of the other, based on a defined sort key. Doesn’t need to be sorted | P |

**Multi-Row Transformations**

|  |  |  |
| --- | --- | --- |
| Aggregate |  | B |
| Percent Sampling | Allows only defined number to go through | N |
| Pivot |  | P |
| Row Sampling | Generates a fixed number of rows, sampling the data from the entire input, no matter how much larger than the defined output the input is. | B |
| Sort | Used for merge joins and for removing duplicates | B |
| Unpivot |  | P |

**Advanced Data-Preparation Transformations**

|  |  |  |
| --- | --- | --- |
| DQS Cleansing | Validates rows by automatically performing data cleansing using an existing knowledge base in Data Quality Services (DQS). | P |
| OLE DB Command | Performs database operations such as updates or deletions, one row at a time, based on mapped parameters from input rows. | N |
| Slowly Changing Dimension | Generates transformations necessary to support loading dimension tables | N |
| Data Mining Query | Applies input rows against a data mining model | P |
| Fuzzy Grouping | Performs de-duplication based on similarity of selected columns | B |
| Script Component | Applies custom .NET scripting capabilities against rows, columns,  inputs, and outputs | N |
| Term Extraction | Analyzes text input columns for English-language nouns and noun phrases | P |
| Term Lookup | Analyzes text input columns against a user-defined set of words for association | P |

**Resolve References editor**

use to quickly resolve the mapping of input and output columns between components.

5.3 – Determining Appropriate ETL Strategy and Tools

**ETL Architecture**

ETL solution using SSIS should have multiple packages dedicated to each process. Means it is more modular and can be better developed

**Lookup Transformation**

Can be used on table, query or cached file

|  |  |
| --- | --- |
| Full cache | Source queried once and start of execution of the task and results cached. But you need enough memory. Will use to match with every row. Will NOT swap memory so if not enough memory will fail. Because the whole table is being put in memory, write SQL so only the required columns are included |
| Partial cache | If the match is found at the database, the values are cached so they can be used the next time a matching row comes in. So cache is populated one row at a time |
| No cache | Database will be re-queried for each row |

Lookup transformation are case sensitive. Can you Character Map transformation to convert case

Best approach for missing lookup – (if you want the rows) set to “ignore failure” then add derived column task to replace eg ISNULL(CustomerDwKey) ? 0 : CustomerDwKey.

Lookup transformation doesn’t need to be sorted. Much more efficient than merge joins.

**Cache Transform transformation**

The Cache Transform transformation writes data from a connected data source in the data flow to a Cache connection manager which persists in memory whilst package is running (or whole project if set at that level.) Or persisted to disk. Benefits:

* Query the lookup once and used for multiple look ups eg for **role-playing dimension** (dimensions used for multiple looks eg Date)
* You can do lookups against other (non OLE-DB) sources.

**Sorting**

Is expensive in SSIS so can be done in SQL using custom SQL then show the source is sorted with SortKeyPosition and IsSorted

**Set-Based Updates**

Don’t do in SSIS. Push back to SQL

6.1 – SSIS Variables

**Parameters v Variables** – parameters are exposed to the caller, variables are not. And parameters are read-only. Parameters collects values that should be determined outside SSIS process.

**Variables** Can store row sets but if large may cause package to run out of memory.

Some of the data types…

|  |  |
| --- | --- |
| Object | general type not represented by another type |
| Byte | integers with values between 0 and 255 |
| DBNull | used to assign an explicit null |
| Int16 | 16-bit integers with values between –32,768 and 32,767 |
| Int32 | signed 32-bit integers with values between –2,147,483,648 and 2,147,483,647 |
| Single | Floating point with precision of seven digits |
| DateTime |  |
| String | Unicode characters |
| Boolean | A simple type representing Boolean values of True or False |

Two types of variables can store row sets. If rowset comes from results of query then Object. If rowset is XML can also be stored in a string and handled Data Flow XML

Source Component.

Namespace – two User and System

**Property Parameterization** allowing specific SSIS object properties to be set dynamically, can be implemented in several ways:

* Set from calling environment
* Explicit assignment of a property from a variable eg SQL from Variable
* Assignment through expressions

6.2 – Connection Managers, Tasks, and Precedence Constraint Expressions

**What should be typically parameterized?**

|  |  |
| --- | --- |
| Connection managers |  |
| Tasks and components | If component relies on values determined by additional programmatic logic |
| Data flow tasks | Large data movements are typically resource intensive; therefore, in order to prevent them from running out of resources, you could adjust their behavior in accordance with the actual availability of resources at run time by using appropriate programmatic logic eg setting batch size |

**Expressions**

SSIS expression **not** part of the .net framework

Some operators

|  |  |
| --- | --- |
| || | Logical OR |
| && | Logical AND |
| ! | Logical Not |
| == | Equals |
| GETDATE | Current date |
| MONTH | Integer that represents current month |
| ISNULL | Returns Boolean if expression is null |

6.3 – Using a Master Package for Advanced Control Flow

**Execute Package task**

Two methods for parameterizing child package.

|  |  |
| --- | --- |
| Package configuration | For every property that should be exposed to the caller (the parent package), a parent package variable configuration must be prepared in the child package. The name of the parent package variable must match the name of the corresponding variable in the parent package |
| Parameters | Variables, project parameters, or package parameters of the parent package can be mapped to the parameters of the child package belonging to the same project as the master package. This method is only available in the Project Deployment  model. |

7.1 – Slowly Changing Dimensions

**Inferred Dimension Members**

When a fact arrives with a business key not in the dimension. Deal with as follows:

* Create a new dimension record with the business key (flag it in some way)
* Create the fact record
* Don’t treat inferred dimension as type 2 SDT because, when it does come in, it will generate new records. Instead, when it comes in, update it and remove flag.

**hash function** - algorithm or subroutine that maps large data sets of variable length, called keys, to smaller data sets of a fixed length. Useful for performing deltas.

7.2 – Slowly Changing Dimensions

**Passing dynamic SQL to source**

* Can be done with a parameterized SQL command
* Build SQL command in an expression of a variable

ODBC or ADO Net can’t accept parameters for their SQL so have to done via their assigning an expression to the SQL command property.

**Change data capture** – can be set up on SQL database on a table-by-table basis that will identify records that have changed. SSIS can then pick this up using CDC control.

Mandatory that SQL Agent is running when you apply CDC. It will do the following:

* Generate cdc schema
* System tables
* Two jobs

CDC components

|  |  |
| --- | --- |
| CDC Control task |  |
| CDC source adapter | Reads a range of change data from CDC change tables |
| CDC splitter | Splits a single flow of change rows from a CDC source component into different data flows for insertion, update, and deletion operations. |

The CDC source offers five possibilities of supplying data:

|  |  |
| --- | --- |
| All | row will be supplied for every change. So if record gets updated 3 times, 3 rows will be supplied |
| All with old values | as All but two rows for every change, one before, one after. |
| Net | One row per unique row. Best for DW. |
| Net with update mask | as Net with additional Boolean column for each column updated |
| Net with merge | as Net but no distinction between update and insert |

**Loading Fact tables**

Partition fact table with incremental data on one partition so can be easily switched.

Use this strategy:

* Load incremental data to a table that has the same structure as the destination fact table, without compression or indexes.
* Apply the necessary indexes and compression.
* Switch the loaded table with the partition in the destination fact table.

Use fully cached lookups to get appropriate surrogate keys.

7.3 – Error Flow

Three options for handling errors in the data flow components

|  |  |
| --- | --- |
| Fail Transformation | whole transfer fails |
| Ignore Failure | value is nullified and transfer continues |
| Redirect Rows |  |

Individual columns can have different settings.

8.1 – Package Transaction

Transactions in SSIS use the Microsoft Distributed Transaction Coordinator, allows distributed transactions providing it has been turned on. Eg could work over SQL and Oracle database.

To enable transactions in SSIS, you must start the MSDTC service, and the tasks that

you want to include as part of the transaction must work with MSDTC services natively.

To enable transactions set TransactionOption. This exists at package, container and control flow task level.

Can be set to the following:

|  |  |
| --- | --- |
| Required | If a transaction already exists, join it; if not, start a new transaction |
| Supported | If a transaction exists, join it (this is the default setting) |
| NotSupported | The package, container, or task should not join an existing transaction. |

If a series of tasks must be completed as a single unit in which either all the tasks are

successful and committed or an error occurs and none of the tasks are committed,

place the tasks within a sequence container and set the TransactionOption property of

the container to Required.

A task can inherit the transaction setting of its parent when the TransactionOption

property is set to Supported,

If you want to exclude a specific task from participating in a transaction, set the

TransactionOption property to NotSupported.

If you set the TransactionOption property of a Foreach Loop container or For Loop

container to Required, a new transaction will be created for each loop.

Running tasks in transactions has a performance overhead

Configuring the MSDTC across multiple hosts and environments in complex and sometimes fragile

**Transaction Isolation Levels**

Set for SQL task or data flow task

|  |  |
| --- | --- |
| Unspecified | A different isolation level than the one specified is being used, but the level cannot be determined |
| ReadUncommitted | Does not lock the record being read ie dirty read |
| Chaos | Same as ReadUncommitted, but checks the isolation level of other pending transactions during a write operation so that transactions with more restrictive isolation levels are not overwritten |
| ReadCommitted | Locks the record being read but then immediately releases it |
| RepeatableRead | Locks the records being read and keeps the lock until the transaction completes |
| Serializable | Locks the entire data set being read and keeps the lock until the transaction completes. Default |
| Snapshot | The data read within a transaction will not reflect changes made by other simultaneous transactions. The transaction uses the data row versions that exist when the transaction begins. No locks are placed on the data when it is read |

RetainSameConnection – connection manager will join existing connection rather than start a new one.

8.2 – Checkpoints

1. Set the SaveCheckpoints property at the package level to True.
2. For the CheckpointFileName property, provide a valid path and file name to the checkpoint file.
3. Set CheckpointUsage to IfExists
4. set checkpoints at the various tasks within your package by setting FailPackageOnFailure to True

CheckpointUsage = Always – checkpoint file will have to exist or the package won’t run. Also using checkpoints is not allowed if you have set the TransactionOption of the package to Required.

8.3 – Event Handlers

|  |  |
| --- | --- |
| OnError | Runs when an executable component reports an error |
| OnExecStatusChanged | Runs for all tasks and containers when the execution status changes  to In Process, Success, or Failed |
| OnInformation | Runs when SSIS displays information messages during the validation and execution of a task or container |
| OnPostExecute | Runs after a container or task successfully completes |
| OnPostValidate | Executes after the task or container has been successfully validated |
| OnPreExecute | Runs before an executable component is executed |
| OnPreValidate | Runs before a component is validated by the engine |
| OnProgress | Executed when a progress message is sent by the SSIS engine, indicating tangible advancement of the task or container |
| OnQueryCancel | Invoked when an Execute SQL task is canceled through manual intervention, such as stopping the package |
| OnTaskFailed | Similar to OnError, but runs when a task fails rather than each time an error occurs |
| OnVariableValueChanged | Runs when the value changes in a variable for which the RaiseChangeEvent property is set to True |
| OnWarning | Runs when a task returns a warning event such as a column not being used in a data flow |

Event Handlers can be turned off for a particular task with DisableEventHandlers property

8.1 – Package-Level and Project-Level Connection Managers and Parameters

**Parameters**

Parameter can have 3 possible values

|  |  |
| --- | --- |
| Design default value | The default value assigned when the project is created or  edited in SQL Server Data Tools (SSDT). This value persists with the project |
| Server default value | The default value assigned during project deployment or later,  while the project resides in the SSIS catalog. This value verrides the design default. |
| Execution value | The value that is assigned in reference to a specific instance of package execution. This assignment overrides all other values but applies to only a single instance of package execution |

9.1 – Package-Level and Project-Level Connection Managers and Parameters

Build configurations provide a way to store multiple versions of solution and project properties.

9.2 – Package configuration

To use package configuration, have to be in package deployment mode. By default package is in project deployment mode.

Configuration types:

* XML file
* Environment variable
* Registry Entry
* Parent Package Variable – can be used to send a variable from a parent package to its child
* SQL Server

Indirect file location approach – using a environment variable as the file location pointer. Useful when the location of the file will change from one environment to another.

10.1 – Logging Packages

**Auditing**

Two ways of capturing logging

|  |  |
| --- | --- |
| Integration Services Logging | Logs information about execution in Integration Services |
| Integration Services Auditing | Provided by the Audit data flow component |

Package logging, five providers. Can be used to log any executable task in package.

|  |  |
| --- | --- |
| Text file |  |
| SQL Server Profiler | writes log to a trace file |
| SQL Server | Writes log entries to the sysssislog system |
| Windows Event Log | Writes logs to Application log that can be viewed Windows Event Viewer |
| XML file |  |

**How will log be used?**

* Low volume and only need for error detection – consider application log or text file
* Need for performance monitoring – SQL server or profiler

Should implement more than one logging, for example, SQL server and Text file in case SQL itself goes down.

**Determining what events need to be logged**

Same as event handler. Can be divided into the following

* execution boundary events – start and end of execution task
* execution progress events – eg variable changes value
* execution exception

exception and boundary events needed for high-level monitoring

**Event properties that can be captured**

|  |  |
| --- | --- |
| Computer | name of computer event occurred |
| Operator |  |
| SourceName | Name of container or task event occurred in |
| SourceID | identifier of package, container |
| ExecutionID | GUID of the execution instance |
| MessageText | most useful for trouble-shooting |
| StartTime |  |
| EndTime |  |
| DataCode | 0 – Success  1 – Failure  2 – Completed  3 – Canceled |

**LoggingMode = UseParentSetting**

Task inherits it’s logging setting from its parent object

Log Configuration Templates – logging setting can be saved as template to use in other packages.

10.2 - Logging Packages

Can be divided into 2 types:

|  |  |
| --- | --- |
| Elementary auditing | one-to-one with the record being audited. Changes in data, who made it, when etc. Usually stored in same place at the aduit eg database |
| Complete auditing | many-to-one with the record being audited eg before and after values. Usually stored in different location |

Auditing is less important in DW than in OLTP

**Audit Transformation** – following system variables are available

|  |  |
| --- | --- |
| Execution instance GUID | The GUID that uniquely identifies the execution instance of the SSIS package |
| Package ID | The GUID that uniquely identifies the SSIS package |
| Package name | Package name The SSIS package name |
| Version ID | The GUID that uniquely identifies the version of the SSIS package |
| Execution start time | The time at which SSIS package execution started |
| Machine name | The name of the computer on which the SSIS package was launched |
| User name | The login name of the user who launched the SSIS package |
| Task name | The name of the data flow task that contains the Audit transformation |
| Task ID | The GUID that uniquely identifies the data flow task with which the Audit transformation is associated |

Audit info and logs can be correlated based on shared properties eg TaskID and ExecutionID

11.1 - Installing SSIS Components

SQL Server Integration Services (SSIS) is an optional element of the SQL Server platform and can either be installed as part of the initial SQL Server installation or added to an existing SQL Server installation later.

SSIS Service used before 2012 and is still available but now referred to as SSIS Legacy Service,

Now we should use SSIS server

**SSIS server** an instance of SQL Server hosting the SSISDB catalog. This database stores the following

* SSIS solutions (projects, packages, and parameters)
* Complete operational history

Production server – limit to features for SSIS storage and execution

Development server – development features

|  |  |  |
| --- | --- | --- |
| **Feature or component** | **Development** | **Production** |
| SQL Server Database Engine | Optional; used to host development  and testing data stores | Recommended; used to host the  SSIS package store (SSIS server or  the msdb database) |
| SQL Server Agent | Not needed | Recommended; used to automate  and schedule SSIS execution |
| Integration Services | Not needed | Required |
| SQL Server Development Tools | Required | Not needed |
| Management Tools – Complete | Recommended; used for SSIS  management | Optional; used for SSIS  Management |
| Client Tool SDK | Required | Not Needed |
| SSIS Server | Not Needed | Recommended; used as the SSIS  package store |

Database Engine Services, installed as part of SQL Server Database Engine, provides limited SSIS functionality; import and export wizard and support for maintenance plans.

**Security Considerations**

Virtual accounts - Virtual Accounts can access the network by using the computer identity in a domain environment.

Managed service accounts - Functionally equivalent to other domain accounts, even though they only exist on the local server. Typically this is used for an SSIS service

Domain account

**64-bit v 32-bit**

SQL Server 2012 is available in 32-bit as well as 64-bit editions; SSIS is included in both, but

some SSIS features are available only in 32-bit editions,

Development tools (SSDT) are available only as 32-bit applications,

To enable the SSIS packages to be used in 64-bit environments, you must have the appropriate editions of the data providers; if you don’t, they will have to be executed using the 32-bit run time.

**SSIS Tools**

|  |  |
| --- | --- |
| SQL Server Import And Export Wizard | Used to copy data between supported data stores |
| SQL Server Integration Services Deployment Wizard Used | Used to deploy SSIS projects to an instance of SQL Server. |
| SQL Server Integration Services Project Conversion Wizard Used | Used to generate a project deployment file from a set of SSIS package files and accompanying configuration files. |
| SQ L Server Integration Services Package Upgrade Wizard | Used to upgrade SSIS packages created in previous versions of SQL Server |
| SQL Server Integration Services Package Installation Utility | Used to deploy SSIS packages to an instance of SQL Server using the deployment manifest. Legacy reasons |
| SQL Server Integration Services Package Utility | Used to manage SSIS packages (for example, to copy, move, or delete them, or to verify their existence) from the command line. |
| SQL Server Integration Services Execute Package Utilities | Two utilities used to execute SSIS packages, either from the command line or from a user interface. |

**What are SQL Server Integration Services (SSIS)?**

SSIS is a feature of SQL Server that hosts SSIS deployment, maintenance, execution, and monitoring.

**How can SSIS be installed?**

SSIS can be installed together with other SQL Server features, added to an existing SQL Server installation, or used to upgrade an earlier version of SSIS.

**What tool should be used to change or modify the SSIS service account?**

SQL Server Configuration Manager – but check, this may have changed

11.2 – Deploying SSIS Packages

**SSISDB Catalog**

Specialized database dedicated to SSIS. The SSISDB catalog serves as the SSIS project and package repository and is the recommended deployment target for SQL Server 2012 SSIS solutions. Holds metadata and definitions: projects, packages, parameters, and environment properties.

Data in the SSISDB catalog is encrypted.

**SSISDB settings**

Can be set from SSMS or by using catalog.configure\_catalog stored procedure.

|  |  |
| --- | --- |
| Encryption Algorithm Name | The type of encryption algorithm that is used to encrypt sensitive data. |
| Clean Logs Periodically | When the value is True (default), operation details and operation messages older than the number of days specified by the Retention Period property are deleted from the catalog. When the value is False, all operation details and operation messages remain stored. |
| Retention Period (days) | The number of days (365 by default) that operation details and operation messages are stored in the catalog. |
| Server-wide Default Logging Level | The default logging level for the Integration Services server. |
| Maximum Number of Versions per Project | The number of new project versions that will be retained for a single project. |
| Periodically Remove Old Versions | When the value is True (default), only the number of project versions specified by the Maximum Number of Versions per Project property are stored in the catalog. |
| Validation Timeout | Validations will be stopped if they do not complete in the number of seconds specified by this property. |

Settings can be retrieved by using catalog.catalog\_properties

**SSISDB Objects**

SSISDB objects represent SSIS projects and packages that have been deployed to the SSIS server, including SSIS project-scoped and package-scoped parameters and environment settings used in various SSIS configurations.

**Folders**

Folder metadata can be retrieved from catalog.folders

**Projects and Packages**

Project and package metadata can be retrieved from catalog.projects, catalog.packages,

and catalog.object\_versions catalog views

**Parameters**

catalog.object\_parameters and catalog.execution\_parameter\_values

**Server Environments, Server Variables, and Server Environment References**

Parameter values can be assigned directly at execution time. Or can be bound to an Environment variable

Environment references represent the association of server variables with project and package

parameters.

catalog.environments, catalog.environment\_variables, and catalog.environment\_references

**Operations** actions performed against the SSISDB catalog (deployments and

catalog-level configurations) and against SSISDB projects and packages (validations and executions).

Accessible from catalog.operations

**Project Deployment** - Under the project deployment model, SSIS project deployment is integrated into SQL Server Data Tools (SSDT), as well as SQL Server Management Studio (SSMS) (ie can be pushed or pulled.) Both use SQL Server Integration Services Deployment Wizard. Can also be accessed by Integration Services project deployment files which have a .ispac file name.

Remember, the SSIS server hosts the SSISDB catalogue

And SSIS projects can only be stored in SSISDB folders

SQL-CLR must be enabled on the SQL Server instance to create and use the SSISDB catalog

12.1 – Executing SSIS packages

User interaction by running the package from SSMS

Programatically by Transact-SQL, Windows PowerShell, the SSIS Managed API, and DTExec (but not DTExecUI)

Execute through T-SQL…

EXEC catalog.create\_execution

EXEC catalog.set\_execution\_parameter\_value

@execution\_id,

@object\_type = 50,

@parameter\_name = N'LOGGING\_LEVEL',

@parameter\_value = @logging\_level;

EXEC catalog.start\_execution

Through DTExec

DTExec /Server localhost /ISServer "\SSISDB\TK 463 Chapter 11\TK 463 Chapter 10\Master. dtsx" /Par $ServerOption::LOGGING\_LEVEL(Int32);1

Through PowerShell (shows how to Instantiate the package)

$catalog = $ssis.Catalogs["SSISDB"]

$folder = $catalog.Folders["TK 463 Chapter 11"]

$project = $folder.Projects["TK 463 Chapter 10"]

$package = $project.Packages["Master.dtsx"]

Every action performed against the SSISDB catalog, or against an SSISDB object, is represented by an operation in the SSISDB catalog—a new row reported by the catalog.operations catalog view. Possible operation status include; Created, Running and Failed

The DTExecUI utility can be used to execute SSIS packages that are accessible through the older SSIS service (stored in the msdb system database or in the managed file system).

**Validation**

Remember, project validation cannot succeed if default values are not set for all required

parameters. And metadata must match.

**Logging Levels** – supported by SSISDB

|  |  |
| --- | --- |
| None |  |
| Basic | Default All events are logged except custom and diagnostic events. |
| Performance | Only performance statistics, errors, and warnings are logged. Used for performance benchmarking |
| Verbose | All events are logged |

Execution can be monitored using **Active Operations viewer** in SSMS

12.2 – Securing SSIS Packages

SSISDB catalog security is determined based on information about which SSISDB principal (SSISDB user) is allowed access to which SSISDB securables (folders, projects, and environments) and what the nature of this access is (to view, to execute, to modify, or to remove them).

Only members of the ssis\_admin database role have full access to any of the objects stored in the SSISDB catalog.

Permissions on SSIS folders, projects, and environments are managed explicitly; permissions on packages, environment references, and variables, are inherited from the objects they belong to.

Who can access SSISDB objects after they have been deployed?

By default, access to SSISDB objects is limited to the users who created them and the members of the ssis\_admin database role.

How can permissions on various SSISDB objects be controlled?

Permissions can be controlled explicitly on folders, projects, and environments, but not on packages, environment references, or variables. Permissions on the latter are inherited from the containing object (the project or the environment to which the object belongs).

Any user who has been granted access to the SSISDB database is permitted to deploy objects to the SSISDB catalog, unless, of course, the CREATE\_OBJECTS permission has been denied to that user on specific folders.

All SSISDB users be default can; deploy projects, modify projects they have created

13.1 – Troubleshooting Package Execution

See value of variables in the Local window. Can assign them to the watch list and change the value.

**Call Stack** debugging window to show a list of the tasks that have executed up to the breakpoint.

SSISDB catalog – Five main reports…

|  |  |
| --- | --- |
| All Executions | all package executions |
| All Validations | all package validations |
| All Operations | overview |
| All Connections | displays the connection context of execution errors |

Predefined reports are available at SSISDB, project and package level

**Data Tap**

Can be used to capture the data from a particular data type at run-time. Can’t be used in design. Create as follows

catalog.create\_execution

catalog.add\_data\_tap

catalog.start\_execution procedure

13.2 – Performance Tuning

**SSIS Data Flow Engine**

Data Buffer - The data flow manages data in groups called buffers. Data Buffers should get passed down from transformation to transformation but sometimes they can get held up because of transformations

|  |  |
| --- | --- |
| Synchronous | Happens in parallel |
| Asynchronous | Happens serially |

Transformations can be divided into three types:

|  |  |
| --- | --- |
| Non-blocking or row-based (synchronous) transformations | work row by row; in this case, the buffer is reused  and no memory is copied. |
| Partial-blocking (asynchronous) transformations | Partial-blocking transformations work with groups of rows; memory is copied, and the structure of the buffer can change |
| Blocking (asynchronous) transformations | store all rows from all  buffers before producing any output rows. |

Synchronous component reuses buffers and therefore is generally faster than an asynchronous component

**Execution Trees**

An execution tree is a logical group of transformations that starts at either a source adapter or an asynchronous transformation and ends at the first asynchronous transformation or at a destination adapter. Execution trees specify how buffers and threads are allocated in a package. Each tree creates a new buffer and may use a new thread so affect performance.

**Backpressure Mechanism**

One of the mechanisms that the SSIS data flow engine uses to achieve high performance is backpressure. If a source or an asynchronous transformation is too fast (compared to the next transformation or destination farther down the path), the source is suspended when its execution tree creates too many buffers (currently the limit is fixed at five buffers). If the source is slow (that is, if transformations and destinations can process data faster than sources generate it), the backpressure mechanism does not get involved and sources can run at full speed. Because of the backpressure mechanism, you can process fast sources without worrying about opening too many buffers and eventually running out of memory.

**Buffer Optimization**

At execution time, before the data is read from the sources, SSIS automatically tunes buffer

sizes to achieve maximum memory utilization based on a series of input parameters. SSIS applies the size of the buffers based on these input parameters:

|  |  |
| --- | --- |
| Estimated Row Size | Isn’t configurable. Calculated from metadata supplied at design time. Can be shrunk by identifying the smallest data type size for columns and selecting only required columns. |
| DefaultBufferMaxRows | Automatically set at 10,000 records. SSIS multiplies Estimated Row Size by DefaultBufferMaxRows to get a rough sense of your dataset size per 10,000 records. |
| DefaultBufferSize | Set to 10 MB by default. Can not change it above MaxBufferSize, |
| MinBufferSize | Isn’t configurable. SSIS uses this internal parameter to gauge whether you have set the DefaultBufferMaxSize too low. |

Objective is to get as many rows as possible in the buffer

You can configure the DefaultBufferMaxRows and DefaultBufferMaxSize properties and, depending on your configured values, SSIS will tune buffer sizes at execution time by using one of the following scenarios:

When Estimated Row Size \* DefaultBufferMaxRows exceeds MaxBufferSize, SSIS reduces the number of rows that will be stored in a buffer to manage the memory footprint.

When Estimated Row Size \* DefaultBufferMaxRows is less than MinBufferSize, SSIS

increases the number of rows that will be stored in a buffer to maximize memory utilization.

When Estimated Row Size \* DefaultBufferMaxRows is between MinBufferSize and

DefaultBufferMaxSize, then SSIS attempts to size the buffer as closely as possible to

the result of Estimated Row Size \* DefaultMaxBufferRow

**Data flow performance tips**

* No swapping to disk
* Push somethings down to the database eg sorts
* Always use only the columns you need
* Define data types as narrowly as possible
* Perform updates by using a staging table
* Use full-cache lookups
* Always use fast load

Change logging level for whole server - In the Operations Log area, find Server-Wide Default

Logging Level and change the property value to Performance.

**Parallel Execution in SSIS**

MaxConcurrentExecutables = number of processors plus 2

Which performance counters display information that the data flow has started swapping to disk storage? - The Buffers Spooled counter

14.1 – Data Quality Problems and Roles

**Master Data Management (MDM)**

In business, master data management (MDM) is a method used to define and manage the critical data of an organization to provide, with data integration, a single point of reference.

**Data Quality Dimensions**

Describes a data quality measure that can relate to multiple data elements including attribute, record, table, system or more abstract groupings such as business unit, company or product range

|  |  |
| --- | --- |
| Hard Dimensions | Can measure directly using SQL queries |
| Soft Dimensions | Subjective, measure through data users |

Hard Dimension

|  |  |
| --- | --- |
| Completeness | In a closed-world assumption, the presence of NULLs is what defines the completeness.   * Attribute completeness * Tuple completeness (row) * Relation completeness (group of rows) * Value completeness (complex eg XML |
| Accuracy | Difficult to measure. Can you statistical techniques to determine if an attribute contains accurate values ie values with low frequency may be incorrect. |
| Information | Defines entropy as the quantification of information in a system |
| Consistency | Compare the same value across multiple systems |

Soft Dimensions

|  |  |
| --- | --- |
| Timeliness | Degree to which data is current and available when needed |
| Ease of use | Relies on user perception. |
| Intention | Is the data the right data for its intended use? Unintended data, for example, would be holding the fact that someone has left the company in the name field. |
| Trust | You have to ask users whether they trust the data. |
| Presentation quality |  |

**Data Quality Schema Dimensions**

|  |  |
| --- | --- |
| Schema completeness | Extent to which the schema covers the business problem |
| Schema correctness | Concerned with the correct representation of real-world objects in the schema and the correct representation of  requirements. |
| Documentation | Tells you whether the schema is properly documented |
| Compliance with theoretical models | Is schema properly normalised? Is there not enough specialization? |
| Minimalization | Number of schema entities should be minimal and should not include objects that are not pertinent to the problem the schema is solving. |

Find root cause of data quality issues with 5 whys. Or by tracking data through the systems.

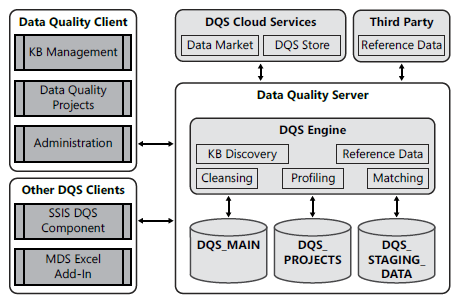
**SQL Server Master Data Services** – application that can implement an MDM solution

Data governance – activities involved in improving and maintaining data quality

Data stewards – people responsible for these activities

14.2 – Installing Data Quality Services

SQL Server 2012 Data Quality Services (DQS) is a knowledge-driven data quality solution. This means that it requires you to maintain one or more knowledge bases (KBs).



**Data Quality Server** of a DQS contains three databases

|  |  |
| --- | --- |
| DQS\_MAIN | Holds the DQS stored procedures which make up the actual DQS engine. Also contains published knowledge bases (ie a KB that has been prepared to be used in a cleansing project.) |
| DQS\_PROJECTS | Data for knowledge base management and data needed during cleansing and matching projects |
| DQS\_STAGING\_DATA | Provides an intermediate storage area where you can copy source data for cleansing and where you can export cleansing results. |

**Data Quality Client**

* Application to manage knowledge bases; execute cleansing, profiling, and matching projects; and administer Data Quality Services.
* SSIS DQS Cleansing transformation for batch cleaning inside an SSIS package
* With the free Master Data Services (MDS) Microsoft Excel add-in, you can perform matching of master data in an Excel worksheet.

**Installation**

What are the prerequisites for installing Data Quality Server? - SQL Server 2012 Database Engine Services

What are the prerequisites for installing Data Quality Client? - .NET Framework and Internet Explorer 6.0 sp 1

Can be installed using SQL Server set up.

Once installation is complete, you have to run the DQSInstaller application. It does the following:

* Creates the DQS\_MAIN, DQS\_PROJECTS, and DQS\_STAGING\_DATA databases.
* Creates two logins needed by Data Quality Server: ##MS\_dqs\_db\_owner\_login## and ##MS\_dqs\_service\_login##
* Creates three roles in the DQS\_MAIN database: dqs\_administrator, dqs\_kb\_editor, and dqs\_kb\_operator.
* In the master database, it creates the DQInitDQS\_MAIN stored procedure.
* Logs the installation in the DQS\_install.log file.

Users must be members of any of the three roles to connect to the Data Quality Server.

To restore the DQS database you will need to supply the original password with this command

EXECUTE [internal\_core].[RestoreDQDatabases] '<PASSWORD>'

14.3 – Installing Data Quality Services

With Data Quality Client you can

* Monitor DQS activities
* Configure reference data service settings
* Enable or disable notifications.
* Configure logging

**Thresholds**

Configure the threshold values for cleansing and matching

|  |  |
| --- | --- |
| Min Score For Suggestions | This is the confidence level that DQS uses for suggesting replacements for a value during cleansing. The default value is 0.7. Must be between 0 and 1. A higher value means fewer suggestions, and thus fewer replacements. This value should be less than or equal to the value for Min Score For Auto Corrections. |
| Min Score For Auto Corrections | This is the confidence level for automatically correcting a value during cleansing |
| Min Record Score | This is the threshold value for the matching policy. It denotes the minimum score for a record to be considered as a match for another record. The default value is 80 percent. |
| Enable Notifications |  |

**Logs**

DQS log - DQServerLog.DQS\_MAIN.log

Data Quality Client log file - DQLog.Client.xml

use Windows Explorer to delete old log files

**DQS\_MAIN roles**

|  |  |
| --- | --- |
| dqs\_admininstrator | Can create or edit a knowledge base, create and execute a DQS project, terminate a running project or other activity, stop a process, and change the Reference Data Services and other settings. |
| dqs\_kb\_editor | Can edit and execute a project and create and edit a knowledge base. Can monitor all DQS activity |
| dqs\_kb\_operator | Can edit and execute a project. |

15.1 – Defining Master Data

master data management (MDM) is a set of coordinated processes, policies, tools, and technologies used to create and maintain accurate master data

The following should be considered when decided if data is master data

* Cardinality has to be high (ie number of actual records)
* Complexity has to be high (ie number of attributes)
* Volatility has to be high (ie new records are frequently created or changed)
* Auditing is required
* Versioning
* Reusage which increases value

Data quality issues are mostly about master data. Master data is often held in more than one source system with different definitions

Goals of MDM are

* Harmonising across systems
* Maintaining multiple copies of the data in different systems
* Integrating master data for analytical and CRM systems
* Maintaining history for analytical systems
* Capturing information about hierarchies in master data
* Supporting compliance with government prescriptions
* Maintaining a clear CRUD process (Create, Read, Update, Delete)
* Maximizing return on investment (ROI) through reusage

**Data governance** sets the policies for master data.

Approaches to MDM

|  |  |
| --- | --- |
| Central metadata storage | Unified, centrally maintained definition for master data which systems should use |
| Central metadata storage with identity mapping | In addition to the above, holds the keys that map tables in different sources. |
| Central metadata storage and central data that is continuously merged | Holds all attributes and the keys. Often implemented in data warehouse |
| Central MDM, single copy | Master data, together with its metadata, is maintained in a central location, where all other system consume from |
| Central MDM, multiple copies | Master data centrally located but only common attributes are stored |

Challenges of MDM

* Different definitions of master metadata in source systems
* Data quality issues
* Authority
* Data conflicts
* Domain knowledge
* Documentation

15.2 – Installing Master Data Services

Master Data Service comprises of

|  |  |
| --- | --- |
| MDS database | Master data is stored along with MDS system objects (system tables and many programmatic objects such as system stored procedures and functions) |
| MDS service | Performs the business logic and data access for the MDS solution |
| Master Data Manager | Web application for MDS users and administrators. |

Data can be entered into the MDS database through these methods

* Master Data Manager web application
* Through staging tables in the MDS database
* Microsoft Excel 2010 with the MDS add-in
* Integrating the Master Data Manager web service in your application

Remember that the Master Data Services web application requires specific Windows Web Server roles and role services, and you must have specific Windows features enabled. It is not enough to use only SQL Server Setup to install the MDS web application.

How can you export data from your MDS database?

You can query the subscription views you create, or you can integrate your application directly to Master Data Services through the Master Data Manager web service.

15.3 – Creating a Master Data Services Model

A Master Data Services model is a container for all master data objects. One model typically covers one master data business area.

A model contains the following objects

|  |  |
| --- | --- |
| Entities | Can be many entities but one will be central called base entity |
| Attributes and attribute groups | Each entity must have two attributes; code and name.   * Free-form attributes – hold any value * Domain-based attributes – list of possible values exists * File attributes – can store documents |
| Explicit and derived hierarchies | * Derived hierarchies - Domain-based attributes form natural hierarchies * Explicit hierarchy - The hierarchy structure can be ragged, which means that the hierarchy can end at different levels. You create consolidated members for the purpose of grouping other members. * Collections - Not a hierarchical structure; it is, rather, a flat list of members |
| Collections |  |

Numeric free-form attributes are limited to the .NET SqlDouble data type.

**MDS business rules** ensure data integrity. Defined as if..then… statements. Can correct, send email or start workflow.

Applied in the following order

1. Generating a default value
2. Changing an existing value
3. Validation
4. External action, such as starting a workflow

16.1 – Importing and Exporting Master Data

**MDS model deployment package** is an XML file that contains MDS metadata and sometimes data as well. File extension is .pkg

* Entities
* Attributes
* Attribute groups
* Hierarchies
* Collections
* Business rules
* Version flags
* Subscription views

Used to transfer the MDS model

MDS packages can be created and deployed using

* Model Deployment Wizard, which is accessible from the Master Data Manager application (cannot include data)
* MDSModelDeploy command prompt utility

**Importing Batches of Data**

First step is to populate staging data. Three tables…

|  |  |
| --- | --- |
| stg.entityname\_Leaf | The staging table where you insert leaf-level members for an entity with the name entityname. |
| stg.entityname\_Consolidated | The staging table you populate with consolidated members. |
| stg.entityname\_Relationship | Use this table to move batches of members in an explicit hierarchy. |

Second step, initiate staging process. Each staging table has an associated stored procedure.

Third, validate against business rules.

Following can be done in the staging process

* Create, update, deactivate, and delete leaf and consolidated members.
* Update attribute values.
* Designate relationships in explicit hierarchies.

**Exporting Data**

Two ways of allowing applications to use MDM data

* MDS web service
* Subscription views.

16.2 – Defining Master Data Security

MDS security is based on Windows local users and groups and Active Directory users and

groups.

To give n user permission to use MDS, you have to assign:

|  |  |
| --- | --- |
| Functional area access | This determines which of the functional areas of the Master Data Manager application a user can access. |
| Model object permissions | This determines which objects in a model a user can access. |
| Hierarchy member permissions | This determines which members of a hierarchy a user can access. |

There are two levels of administrators in MDS:

* MDS System Administrator (only one)
* Model Administrators

To change MDS System Administrator:

1. Find the SID of the user in mdm.tblUser
2. Run EXEC mdm.udpSecuritySetAdministrator stored proc

**Model Administrators** have Update permissions on the complete model— that is, on the top-level object—and no other assigned permissions on lower levels, such as the entity level. And need to be given functional access

Functional access

* Explorer
* Version Management
* Integration Management
* System Administration
* User and Group Permissions

Levels of permission

|  |  |
| --- | --- |
| Model permissions | Applies to all entities, derived hierarchies, explicit hierarchies, and collections of that model. |
| Entity permissions | Applies to all attributes, including Name and Code, for leaf and consolidated members, all collections, and all explicit hierarchies and relationships |
| Leaf permissions | Applies to attribute values of leaf-level members of an entity |
| Consolidated permissions | Applies to the attribute values for consolidated members of entities that have explicit hierarchies enabled |
| Collection permissions | Applies to all collections |
| Navigational Access | Implicit. Automatically granted to higher levels of model if permission granted to lower levels |

If you want a user to have limited access to specific members, you can use the **hierarchy member** permissions.

These are the levels of permissions for entities

* Read-Only
* Deny
* Update

16.3 – Using Master Data Services Add-in for Excel

MDS Add-in allows you to update batches of MDS data without using the staging process. And also to create and edit entities

MDS Add-in for Excel can de-duplicate data by using Data Quality Services if it, and MDS are on the same server.

You can used MDS security to define row-level security and create nonoverlapping roups of members for different users so they don’t update the same row.

17.1 – Creating and Maintaining a Knowledge Base

A knowledge base consists of domains that are mapped to data source fields. Composite domains span multiple fields.

You can get values for domains by the following:

* By knowledge discovery
* With manual editing
* By importing the values from an Excel file

**Building a DQS KB involves**:

|  |  |
| --- | --- |
| Knowledge discovery | A computer-guided process that uses a data sample |
| Domain management | An interactive process in which you manually verify and extend domains in a KB |
| Reference data services | A process in which you validate domain data against external data maintained and guaranteed by an external provider |
| Matching policy | A process in which you define rules to identify duplicates |

Domains can be exported or imported as follows:

* Import and export all knowledge bases from an instance with the **DQSInstaller.exe** command prompt utility. This creates a DQS backup file (.dqsb)
* Import or export (to a .dqs file) an entire knowledge base or a single domain from Data Quality Client.
* Import values from a Microsoft Excel file to a domain with Data Quality Client.
* Import domains from an Excel file with the knowledge discovery activity in Data Quality Client.
* Import new values from a finished DQS cleansing project with Data Quality Client.

**Leading value** – value with which you want the value to be replaced.

**Domain Management**

A domain contains a semantic representation of a specific field (column of a table) in your data source. The following properties can be defined.

|  |  |
| --- | --- |
| Data type | Eg string, date etc |
| Use leading values | This property defines whether to replace all synonyms of a value with the leading value you define. |
| Normalize | Normalize strings to remove special characters and thus improve the likelihood of matches |
| Format output | Format strings to uppercase, lowercase, or capitalized for each word. |
| Speller | Spellcheck the domain |
| Syntax algorithms | Disable checking strings for syntax errors. |

**Synonyms** – set of values that can be corrected to a single leading value

**Term-based relation** – used correct a term that is part of a domain value and not the complete domain value eg Inc. should always be expanded to Incorporated

You can define correct, invalid, erroneous, and corrected values for a domain

**Domain rule** - condition that DQS uses to validate, standardize, and/or correct domain values

**Cross-domain rule** - a rule in a composite that tests the relationships between two or more single domains. For example, you can check that a specific city is always in a specific country

17.2 – Creating a Data Quality Project

**DQS Projects**

Projects can be Cleansing or Matching.

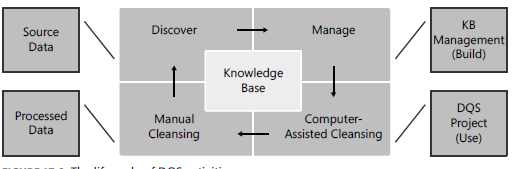
Use the DQS Cleansing transformation for batch cleaning during the ETL process.

Projects can be opened, unlocked, renamed and deleted.

A DQS project uses a single knowledge base.

**Data Cleansing**

Life cycle of DQS activities



A cleansing project has the following stages:

|  |  |
| --- | --- |
| Mapping | Map source columns to KB domains. |
| Computer-assisted cleansing | DQS uses the KB with built-in algorithms and heuristics to find the best match of an instance of data you are cleansing to known data domain values. |
| Interactive cleansing | You review the results of the computer-assisted cleansing and additionally correct data. Accept or reject |
| Export | Export the cleansed data. You can export the data to SQL Server tables or Excel files |

Status of each value can be one of the following

|  |  |
| --- | --- |
| Invalid | Value does not comply with domain rules |
| Corrected | Corrected during the computer assisted cleansing because the confidence for the correction was above the minimal score for the auto-corrections threshold |
| Suggested | Values have a confidence level higher than the auto-suggestions threshold and lower level than the threshold for auto-corrections. |
| Correct |  |
| New | The value cannot be mapped to another status. If you approve a new value, DQS moves it to the corrected values; if you reject it, DQS moves it to the invalid values. |

17.3 – Profiling Data and Improving Data Quality

**SSIS Data Profiling Task**

Simple to use but limited results are XML and require custom code in a Script task to use. Data Profile Viewer can be used to view XML.

Can use it to provide the following

|  |  |
| --- | --- |
| Column Length Distribution | This helps you find strings of unexpected length. |
| Column Null Ratio | Use this to find the percentage of NULLs in a column |
| Column Pattern | Expresses patterns in strings as regular expressions and then calculates the distribution of these regular expressions |
| Column Statistics | This gives you the minimum, maximum, average, and standard deviation for numeric columns, and the minimum and maximum for datetime columns. |
| Column Value Distribution | This gives you the distribution of values for discrete columns |
| Candidate Key | Percentage of unique values in columns, thus helping you identify columns that are candidates for keys. |
| Functional Dependency | Reports how much the values in a dependent column depend on the values in a set of determinant columns. |
| Value Inclusion | Finds the extent to which column values from one table have corresponding values in a set of column values of another table, thus helping you find potential foreign keys. |

18.1 – Data Mining Task and Transformation

Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis.

DM techniques are divided in two main classes

|  |  |
| --- | --- |
| directed algorithms | Target variable that is used to supervise the training; this target variable is the focus of the mining activities, and the trained model is used to explain the values of it with selected input variables |
| undirected algorithms | Trying to discover new patterns inside the data set as a whole, without any specific target variable. |

There are four distinct steps in a typical data mining project:

1. Identifying the business problem
2. Using DM techniques to transform the data into actionable information
3. Acting on the information
4. Measuring the result

Remember, only SSAS installed in Multidimensional mode supports data mining.

Tables in an SSAS database allows nested tables.

How do you perform a prediction by using a data mining model?

You need to create a DMX prediction query that joins patterns stored in an SSAS mining model with your data.

Which SSAS mining algorithm is useful for forecasting? Time Series

Which language do you use for data mining prediction queries? DMX

Which SSAS mining algorithm is useful for market basket analysis? Association Rules

18.2 – Text Mining

**Text mining** is about using an application to automatically gain an understanding of specific text blocks.

The **Term Extraction** SSIS transformation can retrieve the key terms from a Unicode string or text column. It does the following for English text:

* It can indentify words by using different word separators
* It tags words in different forms: noun, verb etc
* It stems words to their internal dictionary form
* It normalizes words
* Finds sentence boundaries

You can also define exclusion words.

Returns two columns:

|  |  |
| --- | --- |
| Term | noun or a noun phrase |
| Score | Frequency or Term Frequency/Inverse Document Frequency |

Has the following properties

|  |  |
| --- | --- |
| Frequency threshold | Specify the number of times a word needs to appear in order to extract |
| Maximum length of term | This is the maximum number of words in a phrase |
| Use case-sensitive term extraction |  |

The **Term Lookup** transformation uses a dictionary of terms that is stored in a column in a SQL

Server table and applies this dictionary to an input data set. Apart from this, works similarly to Term Extraction.

Returns two columns:

|  |  |
| --- | --- |
| Term |  |
| Frequency |  |

18.3 – Preparing Data for Data Mining

SSIS Sampling is one of the most important tasks of a data mining project. Sample should be 30% and should be selected randomly.

Percentage Sampling task - can select a specific percentage of rows for a sample

Row Sampling transformation - can define exactly how many rows you want to have in your sample

19.1 – Script Task

**EntryPoint** property specify the method with which the script should start executing. By default, a method called Main in the ScriptMain class is the starting point

Can you log the execution of your script in the script task?

Yes, you can use the Log property of the Dts object to log the execution of your script.

Do change connection string in a script task do this

Dts.Connections["ContactViper"].ConnectionString = "blah";

19.2 – Script Component

Like script task but use in data flow

You can use the script component to use web services in your SSIS package data flow.

Like configuring Script Task but have to define input and output columns. When SSIS generates the default code for the component, it uses this metadata to create a class for each input and output. The classes have a property for each input or output column.

Output can be blocking or non-blocking based on **SynchronousInputID** property.

**ExclusionGroup** to specify how to re-direct rows to different outputs.

19.3 – Implementing Custom Components

A custom component needs references to the below SSIS libraries

|  |  |
| --- | --- |
| Microsoft.SqlServer.DTSPipelineWrap | to create data flow objects and automate data flow operations |
| Microsoft.SQLServer.DTSRuntimeWrap | to create control flow objects and automate control flow operations |
| Microsoft.SqlServer.PipelineHost | To access SSIS data flow objects and methods |

Two principle classes based on.

Microsoft.SqlServer.Dts.Pipeline.PipelineComponent

Microsoft.SqlServer.Dts.Pipeline.PipelineBuffer

Asynchronous outputs are required when transformation can produce 0, 1 or more rows per input row.

A custom component can be used for the following roles:

* Data Source
* Data Destination
* Data Transformation

Design-time methods can be used:

* To allow the components to be configured
* To allow the components to be validated

Run-time methods can be used:

* To allow the components to be configured
* To allow internal variables to be set
* To allow external sources to be accessed
* To allow data to be processed

20.1 – Identity Mapping Problem

Identity mapping problem occurs when data is merged from multiple sources that can update independently and have no necessarily common key.

Need an authoritative source which could be MDS

Algorithms that find similar strings. DQS matching uses **nGrams**.

Identity matching is a quadratic problem. Can be reduced by:

Partitioning

Sorting neighbourhood

Pruning - intelligently pre-selects a subset of rows from one table for a match with all rows or a single pre-selected batch from the other table

SSIS Fuzzy Lookup transformation is useful for identity mapping

SSIS Fuzzy Grouping transformation is useful for de-duplication

20.2 – Using DQS and the DQS Cleansing Transformation

**DQS Cleansing Transformation**

Use to correct data. Need a knowledge base to do this. Once clean can…

**DQS Matching**

Requires separate matching policy KB. This consists of matching rule. Can specify the following to enhance matching:

* Importance of each domain value
* Specify if a domain value has to be an exact match
* Define degrees of similarity ie a similarity tolerance for numbers and dates

You also define the minimum score for matching. By default it is 80%

DQS survivorship rule = Most complete and longest record

How do you use the matching results when building a matching DQS knowledge base?

You use the matching results interactively to find the best matching rule.

20.3 – Implementing SSIS Fuzzy Transformations

Fuzzy components can deal with much large numbers of records than DQS but is limited to string.

Algorithm starts by using the **Jaccard index**.

Fuzzy transformations convert strings to sets before they calculate the weighted Jaccard similarity (ie break into individual words.) This is called **tokenizer**.

Weight is assigned to each token. Hight weight if occur infrequently.

Tokens are also transferred into more normalised versions.

Also use **Levenshtein distance**. This is the maximum number of edits required to transform one text into another.

Versions of fuzzy transformations

|  |  |
| --- | --- |
| SSIS Fuzzy Lookup transformation | Used for identity mapping. Output includes two additional columns   * \_Similarity * \_Confidence |
| SSIS Fuzzy Grouping transformation | Used for de-duplicating. Output includes:   * \_key\_in * \_key\_out * \_score |
| The Excel Fuzzy Lookup Add-in |  |

This is how you can influence Fuzzy Lookup matching rules

* Define a maximum number of matches per input row.
* Define custom token delimiters.
* Define similarity thresholds.

Additional stuff

**Step to allow a SSIS package in a job access to file share**

Create a domain user and grant access to the share

Create credentials that reference the domain user

Add a proxy the references the credentials

Assign the proxy to the SSIS package execution subsystem

**32-Bit and 64-Bit Data Providers**

The SSIS development environment is a 32-bit environment. At design time, you only have access to 32-bit data providers, and as a consequence you can only enlist those 64-bit providers in your SSIS projects that also have a 32-bit version available on the development machine.

The SSIS execution environment, on the other hand, is dictated by the underlying operating system, which means that, regardless of the version of the provider that you used at design time, at run time the correct version will be used.

**Excel connection manager**

To connect to a spreadsheet with xlsx extension, requires the Microsoft Access Database Engine 2010 (or ACE driver.) Need to install 32bit version to run in Design. In the execution environment it will need the 64 bit version installed. Problem is, you can’t install both on the same server.

**Merge template**

MERGE Production.UnitMeasure AS **target**

USING (SELECT @UnitMeasureCode, @Name) AS **source**

WHEN **MATCHED** THEN

WHEN **NOT MATCHED** THEN

**Query to use to view rows sent between buffers**

select \* from SSISDB.catalog.executions

select \* from SSISDB.catalog.execution\_data\_statistics where execution\_id = ?