DBCC CHECKIDENT: check the current identity value and give us the choice to modify the value

SCOPE\_IDENTITY, IDENT\_CURRENT, and @@IDENTITY are similar functions because they return values that are inserted into identity columns.

IDENT\_CURRENT is not limited by scope and session; it is limited to a specified table. IDENT\_CURRENT returns the value generated for a specific table in any session and any scope. For more information, see [IDENT\_CURRENT (Transact-SQL)](https://msdn.microsoft.com/en-us/library/ms175098.aspx).

SCOPE\_IDENTITY and @@IDENTITY return the last identity values that are generated in any table in the current session. However, SCOPE\_IDENTITY returns values inserted only within the current scope; @@IDENTITY is not limited to a specific scope.

# Stored procedure

--definition

A stored procedure is nothing but a routine that encapsulate a serious of SQL statement. It can have multiple input parameters and multiple output. When a stored procedure is created can first executed, an execution plan of the procedure will be cached, which well be later followed directory by subsequent executions.

--advantages

1. Performance:

2. User friendly: encapsulate multiple SQL statement into a logic unit of work, allow users to bypass the complex logic to get the desired result

3. Security: rather than give a user the permission to access a database, you can grant permission for a stored procedure.

4. Network traffic: rather than send the entire SQL statement to the server, you can just send the name of a stored procedure along with its parameters.

--disadvantage

1. Execution plan latency

2. Cannot be called from select statement, must exec, only way to exec procedure

3. [No table variable output] can’t return table variable as output variable, since table variables are read-only in procedure.

--syntax

1. Create a procedure

create proc p1(@var int, @var1 varchar(100)....)

as

begin

....

end

exec p1 10,'Peter'

2. Catch a return variable

--@rtvalue to catch return value

declare @rtvalue int

execute @rtvalue=p1 150

print @rtvalue

3. Begin...end is not necessary

4. with output variable, and catch output variable

alter procedure p2(@bid int, **@titlevalue varchar(max) out**)

as

select @titlevalue=JobTitle

from AdventureWorks2012.HumanResources.Employee

where BusinessEntityID=@bid

declare @title varchar(500)

exec p2 100, **@title out**

print @title

--Recompile SP

Alter 🡪 can be used in development stage

Sp\_recompile ‘p4’ 🡪 mark procedure for recompile

With recompile 🡪 recompile at execution

--GTK

1. Can’t use the USE <database name> command

2. Can’t use a CREAT statement to create some objects, such as view, function, trigger, procedure

3. Some system stored procedure sp\_bindrule, sp\_unbindrule, sp\_executesql, sp\_helptext

# User defined functions

Functions are subroutines used to encapsulate frequently performed logic.

**Syntax**

**Scalar function**

create function f1(@no1 int, @no2 int)

returns int --every function must have returns

as

begin

return @no1+@no2

end

select dbo.f1(1,4)

tabular function

inline function

create function f3(@beid int)

returns table as

return (select \* from AdventureWorks2012.HumanResources.Employee

where BusinessEntityID=@beid)

select \* from dbo.f3(100)

multiline

create function f4(@beid int)

returns @tabvar table (beid int, firstname varchar(100), lastname varchar(100))

as

begin

insert into @tabvar

select hre.BusinessEntityID, pp.FirstName,pp.LastName

from AdventureWorks2012.HumanResources.Employee hre

inner join AdventureWorks2012.Person.Person pp

on hre.BusinessEntityID=pp.BusinessEntityID

where hre.BusinessEntityID=@beid

return

end

select \* from f4(100)

# Function vs stored procedure

1. Function and stored procedure server separate purpose.

Function is used for computations while store procedure is used to business logic.

1. Stored procedure can have multiple output which function must have one output. It mandatory for a function to have output but not for store procedure
2. Stored procedure can call function but UDF can’t call stored procedure
3. Stored procedure can only be called from execution and function can be called from select statement

Only stored procedure:

Dynamic SQL, error handling, transaction, temporary table

Only function:

Table variable, select statement

# Table variable vs Temp table

1. Definition🡪 scope c. usage

In general, table variable works well with function, it is not a good choice in stored procedure since procedure can’t output table variable. table has more functionality than table variable, such as indexes(other than primary key and unique key), rollback, select into

1. Scope
2. Select … into
3. Performance: in my experience, temp tables usually have better performance, query optimizer will sometimes generate poor plans for @table. For this reason, a general rule is use table variable with there are less 1000 rows, otherwise use temp table. But this will dependent on situation. Table variable don’t support statistics
4. Work in Stored Procedure and UDF

Table variable can be used as an output of UDF, but it can not be a output of Stored procedure.

Temp table can’t be used in UDF

1. Transaction, DDL, index

Only temp table: transaction, DDL, index (other than primary key, unique key)

Only table variable: UDF output

# Trigger

--definition: a database object which is a precompiled set of sql statement, which will be automatically executed on a particular dml, ddl, or logon event. You may think of a trigger as automatically executed stored procedure

--Three types of triggers

DML, DDL, Logon

--syntax

create trigger <trigger\_name> on <talbe\_name>

<after/instead of> <dml event>

as

begin

...

End

# Cursor

--definition

In SQL Server the cursor is a tool that is used to iterate over a result set, or to loop through each row of a result set one row at a time.

--syntax

DECLARE @BusinessEntityID as INT;

DECLARE @BusinessName as NVARCHAR(50);

**DECLARE @BusinessCursor as CURSOR;**

SET @BusinessCursor = CURSOR FOR

SELECT BusinessEntityID, Name

 FROM Sales.Store;

OPEN @BusinessCursor;

FETCH NEXT FROM @BusinessCursor INTO @BusinessEntityID, @BusinessName;

WHILE @@FETCH\_STATUS = 0

BEGIN

 PRINT cast(@BusinessEntityID as VARCHAR (50)) + ' ' + @BusinessName;

 FETCH NEXT FROM @BusinessCursor INTO @BusinessEntityID, @BusinessName;

END

CLOSE @BusinessCursor;

DEALLOCATE @BusinessCursor;

SQL profiler: keep track of details of query execution, all the details generated will be stored in a trace file

DTA (Database Engine Tuning Advisor)

# Query optimizer

-->

Create indexes, or update statistics if the index is already created, check whether there is significant index fragmentation, if yes, reorganize or rebuild,

Rebuild: if average fragmentation in percent is higher than 30% or average page space usage in percent is lower than 70%

Reorganize: average fragmentation in percent is 5% to 30%, or average page space usage is 70% to 95%

-->

Adjust queries (replace set, sub queries with joins if possible, replace cursor with while loop if possible, use three part naming convention whenever possible, avoid using order by)

-->

If it is a DML operation, check if there are any trigger that drag down the performance, try to remove trigger if possible

Trigger on table, partition table

-->

If this query is used frequent. Consider use Stored procedure, user defined function which enable cached execution plan.

If it is already in the form of store procedure, then make sure the cached exec plan is update.

-->

If we are query a huge data, then consider partition this big table into small groups.

# Index

--definition: a database object that helps to improve the performance of data retrieval.

--Two types: clustered and non-clustered

--Good for select statement but will slow down dml operation, index is commonly used DWH

--Balanced tree

For every index created on a table, SQL server will allocates a separate B-Tree which contains root, intermediate and leaf nodes. Each of this nodes a 8 kb page. The root and intermediate hold the starting value of the clustering key for each child node plus the pointer to its child nodes

--Clustered index

When the CI is created, all the data is pulled from the heap and is sorted according to the ascending order of the cluster key, and dump at leaf nodes of the b-tree structure. In this case, the physical storage order is in consistence with the logical sorted order of clustering key. And due to this fact, then we can only have one clustered index on the same table

--Non-Clustered index

When a NCI is created, SQL-server created a b-tree structure and logically sort all the data according to ascending order of the clustering key. Then assign the sorted clustering key along with the pointer into the leaf level of b-tree structure. Those pointers will point to the actual data on heap. NCI will not affect the physical storage order the data, thus we can have multiple Non-Clustered Index

--Two type of non-clustered index

a. NCI-H: if you create NCI on heap, then it is call NCIH. The leaf level of a balanced tree for NCIH has non clustering key value for that NCI, and RID pointers. RID pointers are the combination of 3 numbers, extend number, page number, offset number. RID pointer is used to locate a particular record from heat memory. The process of jump from the leaf of a NCIH with the help of RID is called RID lookup

b. NCI-CI: the leaf level of the b-tree has clustering key value and key value pointer. Key pointer is actually nothing but the clustering key value for clustered index. So it will utilize a cluster index B-tree to locate the records.

--syntax

create unique clustered index <index\_name> on <table\_name> (col1,col2...)

--parameter sniffing 🡪 execution plan depend on the passing parameter

--statistics

--other types of index

a. covering index(index with included column)

b. filter index(index with where clause)

# --JRD

BRD-->Business requirement document

FRD-->Function requirement document

Alternative Document--> stakeholder analysis(SA), business analysis plan(BAP), current state analysis(CSA), scope statement, FRS, SRS

Why document?

Keep track of changes, share info among all persons, finalize and find agreement on requirement, prevent change or legal issues.

* Alternative Methods  
  <http://www.bridging-the-gap.com/what-requirements-specifications-do-business-analysts-create/>
* Business Requirements  
  <http://en.wikipedia.org/wiki/Business_requirements>
* How to Document Business Requirements  
  <http://www.techiesbytes.com/2013/04/how-to-write-good-business-requirement.html>
* Example of BRD  
  <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&ved=0CDgQFjAF&url=http%3A%2F%2Foe.ucdavis.edu%2Flocal_resources%2Fdocs%2Fbusiness_requirements.docx&ei=hnJSVNDSHfK_sQTU44CoCQ&usg=AFQjCNFM88xN0vxckTkFx0b7pp9e272t-g&sig2=oG0IEiBW98q0FCVQRJ44iA&bvm=bv.78597519,d.cWc&cad=rja>

# SDLC (software development life cycle)

Requirement Analysis 🡪 Design 🡪 implementation 🡪 testing 🡪evolution

* Scrum Method  
  <http://scrummethodology.com/>

The Scrum model suggests that projects progress via a series of sprints. A sprint will be no more than a month long, most commonly two weeks. There will be a planning meeting at the start of the sprint, where team members figure out how many items they can commit to during the sprint. At the end, these features are done, meaning coded, tested and integrated into the evolving product or system. There will be daily short meetings, including the ScrumMaster and the product owner. During that time, team members share what they worked on the prior day, will work on that day, and identify any impediments to progress.

At the end of a sprint, the team conducts a sprint review during which the team demonstrates the new functionality to the PO or any other stakeholder who wishes to provide feedback that could influence the next sprint.

ScrumMaster, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level.

The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

Spiral models: This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects.

A prototype is evolved by a fourfold procedure: (1) evaluating the first prototype in terms of its strengths, weaknesses, and risks; (2) defining the requirements of the second prototype; (3) planning and designing the second prototype; (4) constructing and testing the second prototype.

RUP models:

Extreme Programming (XP): very similar to scrum. But (a) tasks in an iteration is priories by client, not the team. (b) Iteration cycle is usually shorter than scrum, usually one or two weeks.(c) XP team accept changes during a iteration.

* Waterfall, Agile, & Scrum  
  <http://www.business2community.com/strategy/project-management-methodologies-waterfall-agile-scrum-0597550>
* Deciding Methodology   
  <http://www.doingbusiness.org/methodology/starting-a-business>
* <http://www.socialmediatoday.com/content/business-process-management-how-choose-between-different-methodologies>
* <http://businessmodelalchemist.com/blog/2011/01/methods-for-the-business-model-generation-how-bmgen-and-custdev-fit-perfectly.html>

# Save point used in explicit transaction or implicit transaction

Implicit transaction: the server will automatically start a new transaction if the currently transaction is committed or roll back.

The transaction remains in effect until you issue a COMMIT or ROLLBACK statement.

# When to reorganize or rebuild?

Rebuild:

Average fragmentation in percent higher than 30%

Average page space use in percent, lower than 70%

Reorganize:

Average fragmentation in percent 5% to 30%

Average page space use in percent, 70%-95%

Options along with rebuild or reorganize

# Steps to do table partition?

Can you create trigger on a view? Only instead of trigger can be used on a view.

Trigger on view: when you have a view define by joining multiple table, in order to modify the modify the underlying data, you can use an instead of trigger on the table

ADO, OLEDB, ODBC

# Why do we want surrogate key?

**1. To separate the DWH from the operatonal environment**  
e.g. reuse historical ID

**2. Performance**  
save space + faster joins (with integer)

**3. History preservation**

**4. Late arriving dimension**

# What types of dimensions are there?

SCD

Rapid Changing Dimension, Rapid Growing Dimension

Conformed Dimension vs. Roll Playing Dimension

Degenerated Dimension

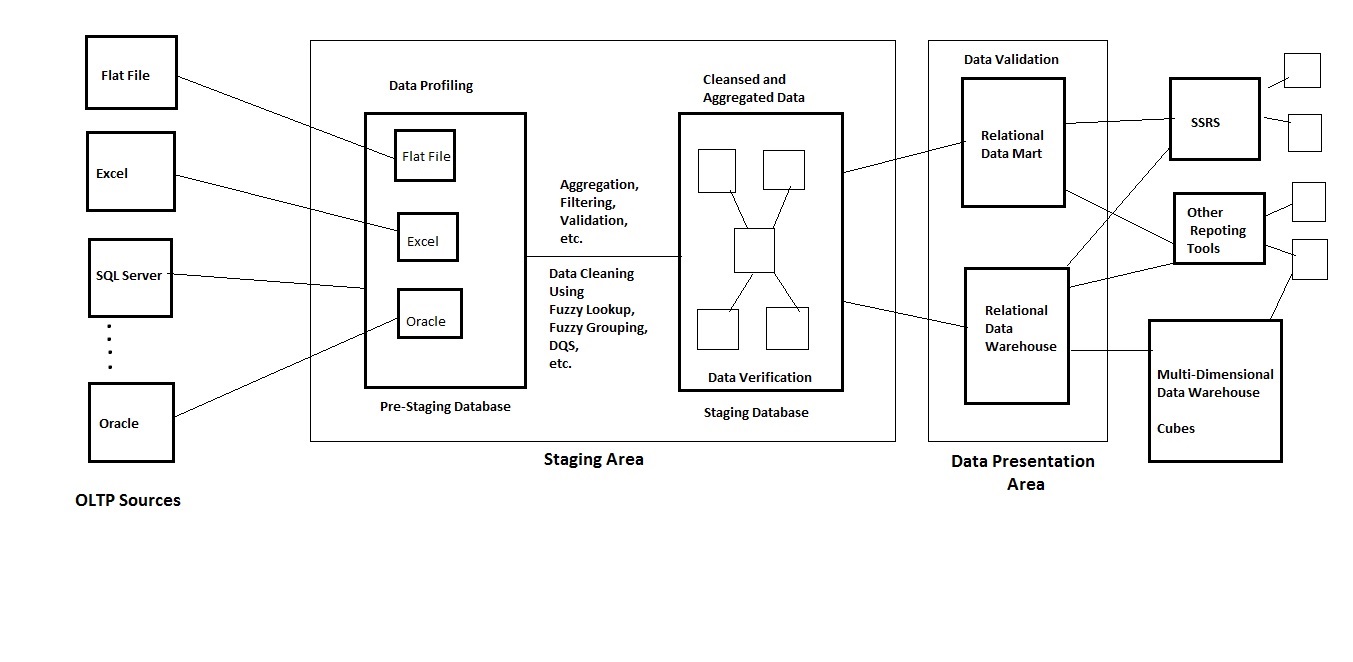
Junk Dimension

# What types of facts table are there?

**Cumulative:** This type of fact table **describes what has happened over a period of time**. For example, this fact table may describe the total sales by product by store by day. The facts for this type of fact tables are mostly additive facts.

**Snapshot**: This type of fact table **describes the state of things in a particular instance of time**, and usually includes more semi-additive and non-additive facts.

# ETL strategy



**Extract**

We started by loading data from various sources, including SQL server, Access Database, Excel. In SSIS, we create different connection managers to load this sources into a pre-staging database.

**Excel**🡪 EXCEL source adapt, Microsoft OLE DB Provider for Jet 4.0

**Access 2007** 🡪 OLE DB provider for the Microsoft Office 12.0 Access Database Engine.

**SQL server** 🡪 OLE DB, Native provider

Usually, the transformation start from the pre-staging, from the pre-staging database, we perform data profiling on all extracted table, in order to get more insight on the underlying data. For example, to find the potential candidate key. To see how one column is depend on another.

Data profiling (column value distribution, column length distribution, null value ratio, column statistics, candidate key, dependency, overlapping)

Beside data profiling, at pre-staging era, we also spend significant time performing data cleansing. This includes correct the spelling typos, misspelling, different abbreviation. Convert one data type to another. Remove the duplicated rows by using the fuzzy aggregation transformation.

Other transformation will also be applied to the data to accommodate the business requirement, such as aggregation and derived column.

So final transformed data will be transfer to another staging data base, from there the data will loaded to the data mart.

In term of load, we will first conduct an initial load which will load all the data available from OLTP. Then periodically, incremental loading will be carried out to insert new data and update modified data. We can use method like checksum, time stamp, and change data capture.

Transformation

# Problem with Excel file extraction

**Missing values**. The Excel driver reads a certain number of rows (by default, 8 rows) in the specified source to guess at the data type of each column. When a column appears to contain mixed data types, especially numeric data mixed with text data, the driver decides in favor of the majority data type, and returns null values for cells that contain data of the other type. (In a tie, the numeric type wins.). You can modify this behavior of the Excel driver by specifying Import Mode. To specify Import Mode, add **IMEX=1** to the value of Extended Properties in the connection string of the Excel connection manager in the **Properties** window.

**Truncated text**. When the driver determines that an Excel column contains text data, the driver selects the data type (string or memo) based on the longest value that it samples. If the driver does not discover any values longer than 255 characters in the rows that it samples, it treats the column as a 255-character string column instead of a memo column. Therefore, values longer than 255 characters may be truncated. To import data from a memo column without truncation, you must make sure that the memo column in at least one of the sampled rows contains a value longer than 255 characters, or you must increase the number of rows sampled by the driver to include such a row. You can increase the number of rows sampled by increasing the value of **TypeGuessRows** under the **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Excel** registry key.

**Data type and length conversions**. Integration Services does not implicitly convert data types. As a result, you may need to use Derived Column or Data Conversion transformations to convert Excel data explicitly before loading it into a non-Excel destination, or to convert non-Excel data before loading it into an Excel destination. In this case, it may be useful to create the initial package by using the Import and Export Wizard, which configures the necessary conversions for you. Some examples of the conversions that may be required include the following:

* + Conversion between Unicode Excel string columns and non-Unicode string columns with specific codepages
  + Conversion between 255-character Excel string columns and string columns of different lengths
  + Conversion between double-precision Excel numeric columns and numeric columns of other types

# What is CDC, time stamp, checksum?

Time stamp

(Data Type)Time stamp is a data type which hold a **binary number of 8 byte size**.

(Incremented)Every time a **row with a timestamp column** is modified or inserted, an **incremented** time stamp value will be inserted to the timestamp column.

(How it work)In order to perform loading using time stamp, first add a column with timestamp data type in Source table. For initial loading, loading all the current data to destination table. Create a tracking table, record the largest timestamp value from the inserted records. For incremental loading, filter the source records with timestamp value greater than the timestamp in tracking table. Update tracking table with the large timestamp of this part of data. This part of data is either updated or newly inserted. Then compare the business key, if the business key same, then update, if not, then insert.

Pros: Good Performance, Guarantees unique binary number for each dml modification, there will be no collision, good for large database

The only cons is that the procedure may be complex if number of table are increased.

CDC

(Definition) CDC provides an easy way to capture changes to data in a set of database tables so that these changes can be transferred to a second system, such as a data warehouse

(Implementation) To implement CDC, you first make sure SQL server job agent is running. Enable the DB to support CDC [sys.sp\_cdc\_enable\_db].

Enable the table to support CDC [sys.sp\_cdc\_enable\_table @source\_schema=?, @source\_name=?,@role\_name=?,@supports\_net\_changes=1]

(CDC.tables)When enabling CDC, the server will generate a new CDC schema, several system tables. One of the table is cdc.stg\_CDCSalesOrderHeader\_CT, which the change tracking table. All changes are stored with additional columns describing, for example the type of change. 1 stand for delete, 2 stand for insert, 3 stand for value before update, 4 stand for value after update.

(Two jobs)Two jobs inside the SQL server agent are created, one for capture, one for clean-up.

(SSIS CDC Component)

CDC control task: The task maintains the state of the CDC package in an SSIS package variable. It also persist an SSIS package variable in a database table so the status is maintained across package, for example, you may have a package to do initial load. **This variable will be used by the CDC source data flow component to determine the current processing range for change records.**

CDC source adapter: offers multiple ways to **retrieve change data**. Among which the NET mode is very suitable for typical DW ETL sensorial

For example, insert one record and update it twice, it will be seen as one insert with the updated value

CDC splitter: splitter a single flow of change rows from CDC source component into different data flows for insertion, update and deletion operation. You can direct rows to corresponding destinations.

Checksum

(Definition)Checksum computes a hash value, called the checksum value, over list of arguments. For every delta operation on included columns, checksum value gets updated.

(Computed column and tracking table) In the source table, add a computed column as checksum based on columns that are expected to change. 🡪 Create a tracking table with two columns, Business key and checksum column.

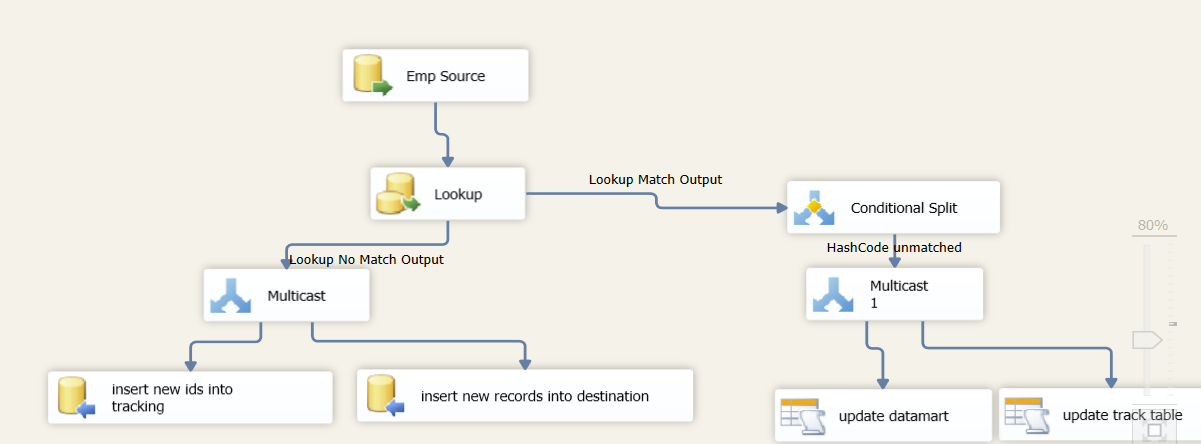
(Initial load)Populate destination table, popular tracing table for all business key and corresponding checksum   
(Incremental) compare Business Key of tracking table with source table. In SSIS, this will be a lookup transformation with tracking table as reference table. Not match mean new record, insert into destination, insert into tracking table. In SSIS, you will need to configure not match to not match output. Then matched, either no change or update. Compare when hash value match, with conditional spilt. Hash value match, then original record. Hash value not match that means update, therefore update destination and tracking tale.

(Pros) Fast and Easy to implement, good for small and mid size DB.

(Cons) Probability of Collision is high. Checksum will be different for tow checksum that are being compared

Inserted, Updated, Deleted, Not changed

How to identify delete, compare the OLTP table to OLAP table,



Slow changing dimension transformation

(Definition)The SCD transformation coordinates the updating and inserting of records in data warehouse dimension tables.

(Types of change supported, and corresponding output)

**Changing attribute/type 1** 🡪 Changing Attributes Updates Output

**Historical attribute/type 2** 🡪 Historical Attribute Inserts Output & New Output

**Fixed attribute/type 0 🡪** Fixed Attribute Output

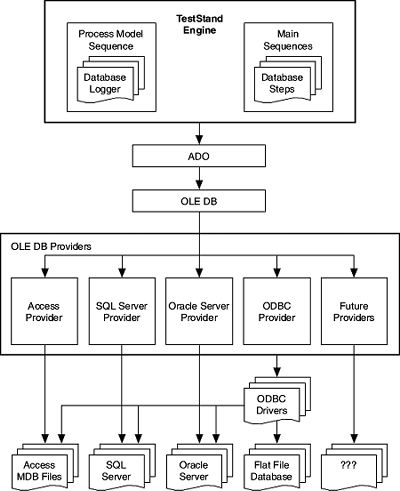
**Inferred member/late arriving dimension** 🡪 Inferred Member Updates

(Configuration) create data source🡪create slowly changing dimension transformation🡪slowly changing dimension wizard 🡪 connection manager for the dimension table 🡪 select a business key 🡪 define dimension column change type, type 0,1,2 🡪 for the type 2, specify how you want to indicate historical record, start/end date or current/expire 🡪 inferred member indicator

# Describe entire ETL process

# Process of design a data mart

1. JRD, business requirement
2. Data Sources
3. Granularity
4. Design the fact and dimensions
5. Determine whether to use a top down approach and down to top approach
6. Star schema or Snowflake
7. ETL strategy

Data sources, EXA<

# OLEDB🡪relational + non-relational

ADO🡪API to OLEDB

ODBC🡪relational (seems obsoleting)

# How to optimize SSIS packages

(Control flow level)

1. **Precedence constraints**, MaxConcurrentExecutables
2. **Event handler** for error handling
3. **Master-child package**, ExecuteOutOfProcess
4. Use **Execute SQL task** (instead of doing everything on the SSIS level)
5. **Lo**g only things that are necessary instead of everything

(Data flow level)

1. Use **SQL command as OLE DB source** and use Joins and order by here insdead of doing them using SSIS transformations
2. Try to avoid **blocking transformations** Avoid merge join and merge transformation because the input must be sorted
3. IsSorted property on the output of an upstream data flow component to True
4. Lookup transformation 🡪 full cache reference table
5. Fuzzy lookup transformation 🡪 use the ETI(error tolerance index), Exhaustive Match to False, Warm Caches to True
6. Increase the **Buffer size** of data pipelines, by default 10MB, Maximum 100MB. You can also increase the DefaultBufferMaxRows
7. When it comes to a local server, use SQL server destination
8. Increase **EngineThread** which is the same as MaxConcurrentExecutables in the control level
9. Use tem tables to perform bulk update in place of OLEDB command since it updates one record at a time

How to optimize ETL process

# How to configure a package/how to make a package dynamic?

**Configuration in Project Deployment Model**

Parameterize

Master-Child package

SSIS catalog🡪SSIS environment

Actually, after you deploy a package to the server, each time before you executed it. You will have the choice to modify configuration

**Configuration in Package Deployment Model**

XML

SQL server

Registry Entry

Parent package variable

Window environment variable

# Lookup vs. Fuzzy lookup vs. Fuzzy group

Lookup

1. (definition)The lookup transformation performs lookups by joining data in input column with columns in a reference dataset
2. (reference connection)The lookup transformation use an OLE DB connection manager or a cache connection manager to connect the reference dataset
3. (reference dataset type)The reference dataset can be a cache file, an existing table or view, a result from a query
4. (no match)The lookup transformation tries to perform an equi-join between vales in the transformation input and the reference dataset. By default, the transformation treats rows without matching entries as errors. However, you can configure the transformation to redirect rows to no match output
5. (multiple match)If there are multiple matches in the reference table, the lookup will only return the first match returned by the lookup query.
6. (data type)The transformation support joining columns with any data type but DT\_R8, DT\_R4, DT\_TEXT, DT\_NTEXT, DT\_IMAGE
7. (Configuration) set up base table 🡪 create lookup transformation 🡪 configure connection manager for reference table, select cache mode, and configure how to deal with unmatched rows. 🡪 define joining column, column you want to loop up in the result set 🡪 destination to catch matched and unmatched output
8. (Usage) determine which new record are which are old, by comparing the current table with previous table.

Fuzzy Lookup

1. (Definition) the fuzzy lookup transformation performs data cleansing tasks such as standardizing data and correcting data, by comparing the base table to a reference dataset. It will use fuzzy match to return one or more close matches for each input records
2. (reference)The reference dataset **must be a table in SQL server database**
3. (fuzzy match/exact match) You will have choice to perform exact match although this is a fuzzy lookup
4. (Data type) Only support the DT\_WSTR and DT\_STR for fuzzy matching
5. (Maximum number of Matches/token delimiter/similarity thresholds)

The similarity thresholds can be match can be set as a number between 0-1, **The closer to 1 the threshold is, the more similar the records must be qualify as match**

1. (similarity score/confidence score) The output will contain two statistics. Similarity score is a mathematics measure that indicate the level of similarity. A similarity score of 1 means exact match. Confidence score is a measure of how likely a particular value is the best match among all match returned.
2. (Exhaustive/Warm Caches)

Exhaustive🡪 more intensive search

Warm Caches🡪 reference will loaded into memory before transformation.

Fuzzy aggregation

1. (definition)The fuzzy group transformation performs data cleansing tasks by identifying rows that are likely to be duplicates and selecting a canonical row
2. (token delimiter/similarity threshold) Threshold can be set to something between 0 and 1, the higher the threshold, the more similar the record have to be in order to be qualify as duplicates.
3. (Data Type) String
4. (key in/key out/score)The transformation produces one output row for each input row, with the following addition columns.

Key in, a column that identifies each row

Key out, a column that identifies a group of duplicate rows

Score, indicate the level of similarity of the input row to the canonical row

1. (FuzzyComparisonFlags) A property used to configure how the transformation compare string data.
2. (Exhaustive) if set to be true, the transformation compares every row in the input to every other row in the input

DQS

1. (definition) A data quality service is a feature introduced in SQL 2012. It uses data quality service (DQS) knowledge base to perform data correction, enrichment, standardization.
2. (domain specific knowledge) Unlike fuzzy lookup and grouping, DQS depends on domain specific knowledge base(KB)

…to be continue

# What is OLE DB command transformation?

The OLE DB command transformation runs an SQL statement for each row in a data flow. For example, you can run an SQL statements that insert, update or deletes rows in a database table

Typically, the SQL statement will contain parameters. The parameters are mapped to the columns of the transformation input.

Let take an example, suppose in the transformation input, you have multiple rows, which contain two columns, sales id and sales amount column. Suppose these are rows you want to update in the sales table. Then when configure the OLE DB transformation, UPDATE SALES SET SALE\_AMOUNT = ? WHERE SALES ID = ?, Then in parameter mapping, map these two question marks to reverent columns in the input. Then the task will update the data one row after another.

# Time stamp stored procedure

I created a stored procedure to perform data loading using time stamp. The procedure extract data from a source table, perform both initial loading as will incremental loading. It will automatics detect a particular loading is incremental loading or initial and load data and update tracking table automatically. If it is incremental load. It will detect the updated records and inserted records. The procedure is pretty complex in that it involves dynamic SQL, table variable, joins, if…else statement, system function and CTEs.

How to do table partitioning?

Creating a partitioned table or index typically happens in four parts:

1. **Create a filegroup or filegroups** and corresponding files that will hold the partitions specified by the partition scheme.
2. **Create a partition function** that maps the rows of a table or index into partitions based on the values of a specified column.
3. **Create a partition scheme** that maps the partitions of a partitioned table or index to the new filegroups.
4. Create or modify a table or index and specify the partition scheme as the storage location.

How to drop identity constraint?

SET IDENTITY\_INSERT [ *database***.**[ *owner***.**] ] { *table*} { ON | OFF }

SQL injection Attack

Let’s table dynamic query as an example, since dynamic query is depended on the value of variable passed by front end users, the users will have some control over what query is executed at the backend. In this way, a hacker may image the query structure and pass some value or statement that have special meaning for SQL to break the query to retrieve more information.

Error handling

This is usually done with TRY CATCH block. You can encapsulate SQL statements in a Begin try and End try block. If there is any error happens in the try block, the control will be pass to the catch block.

In the catch block, for example you can print out the error message. We have a function called ERROR\_MESSAGE (), which will return the system error message. We can also use Raise Error () function to print out our customized error message, along with the severity number to indicate how serious the problem is. There is also a system variable @@error, which returns the error number for the last transaction executed.

Security Package

**Encrypt [All/Sensitive] With [User Key/Password]**

**Cursor**

NTK

1. Cursor
2. Error handling
3. OLEDB command
4. Common ETL strategy(checksum, CDC, timestamp)
5. Transformation type, synchronous and asynchronous
6. Different types of transformations
7. cross feet change notation
8. insert, update, delete anomaly

Hierarchical column

Cross apply outer apply

Cannot TOP and OFFSET in the same query

Error caught by a catch statement are all errors with severity >10 that don’t close the connection

Non clustered intex used for have only a few, evenly-distributed unique value

Concat()