Data importing and control

Introduction

Many batch systems are based on the availability of source data imported into a data warehouse then moved into data marts for reporting and querying. This process gives a framework for building the data marts and importing data. It is based on source data received as a periodic import from another system which may be files. Note that a file import gives isolation of the systems and gives a system that is easy to develop, test and control.

Database structures

Based on Kimball processes the data warehouse is often a collection of star schemas and the data marts a subset of this with very little processing. Only the terms star schema and snowflake schema are commonly referenced.

Star Schema

A star schema is built around a fact table, which may be considered store of transactions, and dimensions which may be considered tables that hold the attributes for the facts/transactions. Dimensions are linked to the fact table via a surrogate key, often an identity. Note: for a dimension which would consist of a single small value may be represented as a column on the fact table, this is called a degenerate dimension and is logically no different.

A fact table will hold references to dimensions (keys) and numeric values which are summable (measures). The measure may not be summable over every range in which case they are called semi-additive measures.

Usually the speed of access of data is governed by the amount of disk i/o. In this respect in number of rows usually fact tables are very large and dimensions vary between very small and the same size as the fact table. As the fact table only has short value columns its size is reduced. The dimensions are usually composed of fewer wider columns. Therefore the disk i/o will be greatly reduced compared to a flat table but greater than a normalised database. Therefore we would expect a star schema to give resource usage and performance between that of a normalised and denomalised database. Note that SQL Server has been optimised to process star schemas so the advantage of normalisation 9in performance terms is less than it once was. The great advantage of star schemas is that they are easy to understand and query and so need less knowledge of database theory to implement

Snowflake schema

A snowflake schema is similar to a star schema except that there is a hierarchy of dimensions. This allows for some normalisation to be performed on the dimensions improving resource usage and data integrity.

Galaxy schema

A galaxy schema is composed of a collection of fact tables and associated dimensions – some of which will be common to multiple fact tables (conformed). This is usually called a collection of data marts and would be the norm for a data warehouse based on dimensional modelling.

Fact constellation schema

The dimension hierarchies are split into separate dimensions.

File processing

Data can be received in many ways but the simplest is to transfer via files. This means that the systems do not need to be connected and the data extract and data load can be asynchronous processes.

Things to watch out for
The daily file will be delivered at 10:00 every day
 What is the likelihood of this
 What do we do if it is late
 What happens if the file is not processed for a few days
The daily file will be named myfile.txt
 What happens if the file is not processed for a few days
We will transfer the file to your system with this name
 What happens if we try to process the file while it is being transferred

There is no need to be concerned about these if the system is set up flexibly

Weaknesses in systems occur at changes in data ownership and changes in process control. It is better for the owner of the data to be in control of the process. Therefore the source system should extract the data and create a file. At this point the destination system takes over ownership and controls the processing.
I would advise a filename including a timestamp e.g. xxxxx_yyymmdd_hhmmss.ext. The formats yyyymmdd hh:mm:ss and yyyy-mm-ddThh:mm:ss are the only two which are unambiguously interpreted by sql server.
Important
The file will appear in the system as soon as the creation starts. This is just inconvenient for a file system transfer as there will be an error if access is attempted. Often ftp does not honour the file locks so access can be successful on an incomplete file. I would therefore recommend that either a file is always transferred with a temporary file name then renamed on completion or a control file transferred after the file to indicate it is available.
Then the destination system can poll for files and process any that arrive. There is no need to schedule a processing window dependent on the expectations from the source system.

If the file is created with a fixed name then part of the file detection should be to rename the file with a timestamp. This timestamp should be obtained from the file creation date rather than the detection date. The rename task should be executed before the file detection, this is to allow historic files to be placed in the folder and processed and also so that the file detection component can be reused independently of the other processing needed.

The destination system should implement the file detection and file processing as separate tasks. This means that a record is kept of when the file is first detected and the processing can be scheduled separately, also ifthe file processing takes a long time or fails it does not affect the file detection.

If we go back to the “Things to watch out for“

The daily file will be delivered at 10:00 every day
 We don’t really care – we will process whenever the data is available

The daily file will be named myfile.txt

Better if the source system can give distinct names but we will rename via a dedicated task

We will transfer the file to your system with this name

Source system should change to rename after transfer – otherwise this gets complicated and we may have to put in a delay after detection before processing or keep checking the file size

File Polling

The simplest way of implementing this is via a stored procedure but this requires features that may be blocked by your implementation. In this case SSIS is probably the expected technology.
In any case the component should be parameterised. The following parameters are usually all that is required.
The file name is inserted into the FileProcess table

SourceFilePath
SourceFileMask

And for SSIS packages for the location of the FileProcess table
ServerName
DatabaseName

For the rename task – again a stored procedure or SSIS package
SourceFilePath
SourceFileMask
DestinationFileMask

We now have two tasks which can be scheduled to run frequently (usually between one and five minutes) to detect a file, rename it and insert the filename into a table for processing.

Processing infrastructure

The processing method is based around an infrastructure which has no knowledge of the data or structures.
Therefore we have the following tables table

FileProcessing	The names of the files to process
BatchType	The batches to run on the files
ProcessControl	The steps to be run on each batch
BatchControl	The current status of a batch – last process run
BatchProcessHistory	The history of batches run including command executed

ProcessControl holds the tasks to be run on a batch but has references to BatchControl to obtain any information specific to the current batch e.g. file name, date.

Processes

File Naming

I would recommend a data stamp be included with a file name preferably by the source system or, if that is not possible, added as part of the detection process.
A good default file name format is xxxxxxx_yyyymmdd_hhmmss.xxx.

File detection

This is one of a number of generic components usually limited to polling a folder or ftp site or a time based trigger.

For a folder receiving files via ftp caution needs to be taken that the file is not processed before the delivery is complete as often windows file locking is not implemented for ftp.

In all cases it is better that the source system either the renames/moves the file after delivery or delivers a control file to indicate delivery is complete.

This task just deals with the file detection and renaming. Any other processing e.g. unzipping is dealt with as part of the batch process.

Batch Creation

Create self contained batches to carry out processing on the file.

A batch contains a series of tasks that are carried out on a file including archiving.

Zipping multiple days worth of files will be carried out as another batch.

Batch Processing

The tasks within a batch are executed on the file.

This is controlled by the stored procedure s_ProcessControl. Note: this does not need to be a stored procedure, other technology could be used. For each batch it executed the next task in ProcessControl using the data in BatchControl.

File Processing

File detection

Optionally poll input folder, detect and optionally rename file
Poll input folder and report file detected (Insert file name into FileProcess table)

Batch creation

May be a single batch to process the file or may be many batches triggered by file detection
Create a batch or batches for the file

Batch processing

Run the tasks associated with the batch on the file data
Archive the file when batches complete

General processing for a file

Detect file – add to FileProcess
Create Batch – add to BatchControl
Batch Processes

- Import File to staging table
- Populate staging tables for dimensions
- Populate staging tables for facts
- Update data warehouse dimensions
- Update data warehouse fact table
- Archive file

Advantages

Allows the reuse of components and auto generated code. It provides logging and error handling for all processes executed giving a record of performance and processing carried out.
Provides a flexible priority system between batches and time windows for each.
Visibility of the tasks that will be carried out for a batch.
Allows dependencies between batches.
Provides an error alerting system including batches which are late or overrunning.