

IDE Main Store - Metadata and data storage





IDE Process Store – User-based
processing environment

# Introduction

The data warehousing approach we apply is taken from [Ralph Kimball](http://en.wikipedia.org/wiki/Ralph_Kimball), with the implementation following a slightly modified [Operation Data Store](http://en.wikipedia.org/wiki/Operational_Data_Store) architecture.

From the database schema fragments shown above our implementation consists of two database:

1. Main Store Database: The permanent holding area for all data and metadata. Contains all “versions” (changes) that have been made to the data.
2. Process Store Database: A user based storage area for data that is being modified. A copy of the data is moved from the main store to the process store as required. Multiple copies may exist at any given time.

# The Main store

Essentially, this database consists of metadata tables which are used to describe the datum that are either imported from source or created / modified as part of most statistical process.

In order to do this data is broken down to a “cell” (datum) level and stored as a string value in a “fact table”. The metadata which is captured around the cell at a minimum describes the basic information known from the incoming dataset, variable (column) name, data type, dataset name. We add additional metadata to provide grouping and ultimately assist in later statistical processing.



The overall solution is designed to support different “fact” structures, however in reality we only have two implemented, a basic single set storage structure termed “fact\_x” and a second which can store related datasets “fact\_transaction\_x”.

Multiple instances of these table structures are used to partition the data, which is done solely to increase performance as different physical storage can be assigned to each partition. Partitioning is applied based on our cii\_key (collection\_instance\_instrument) which roughly equated to the period the dataset is for (monthly, quarterly, bi-annual, annual).

The main store approach isn’t kind to deleting data, so we don’t. Instead each fact has a status assigned to through the change\_history table, with new cells being inserted as the facts or their status change. This again helps with performance, but does mean that lots of versions can build up if data is changing frequently – something the standardisation of our statistical processes is designed to reduce.

# The Processing Store

This database is designed to be a user isolated environment where processes can be run without increasing the version count of facts (as described above). It enables both the “production” processing and trailing of experimental / incremental improvements to be done within the same system, using the same data, tools, etc, without adversely affecting data held in the main store.

Essentially, a copy of the data that is required for the processing is moved from the main store fact table(s) to the process fact table(s) and associated with a “process”, which in turn is associated with a specific user. Only processes which are designated as “production processes” can then in turn move data back from the process store to the main store.

In combination, the main store / process store approach eliminates the need for using timestamps to track what has changed, as only one production process can act on a dataset (collection\_instance) at any given time.

# Data Processing

The “processing” that is done on the collections consist mainly of:

1. Derivations
2. Automated edits
3. Imputation

Where possible automated processing is used, although some manual editing is provided for within the overall system. However to facilitate automated process the system enables the capture of user defined inputs to the set of standard tools which have been integrated into the system. These input are termed configuration items and are held as xml.



The configuration set(s) are only applied to data held in the processing store. A configuration set will also hold in the instructions to move data to / from the process store.

# Reporting

In order to determine what has occurred when a given configuration set has been applied, OLAP cubes are used to summarise and display the changes between the raw and final data, providing drill through to the actual process that modified a given cell.

The same type of summary information is also available on the main store OLAP cube although this cubes primary purpose to link collections which share common dimensions.