Week 2 – Data Warehouse Design Overview

There are many different strategies for designing a data warehouse. Inmon presents a classical design process that has been accepted in the industry as a best practice. He notes that data warehouses cannot be built in the same way as traditional applications. As was seen in the first chapters, it is often a process that takes years to complete. This week will focus on Inmon’s process at a high level. It will explore the process for creating a data model in detail, and provide the framework for a deeper dive in future weeks.

Remember our discussion of Inmon’s design. Data flows from operational systems, into a staging area, and ends up in the data warehouse. Let’s focus on the data that comes into the warehouse.



He describes three types of data:

* Archival data is historical data. Since it is not changing, it is the easiest to load.
* Data in the operational environment presents some challenges; however, given the time-variant aspect of data warehousing it would simply be a snapshot of a particular state of the data at a given time.
* Ongoing updates present the biggest challenge since operational systems are constantly changing.
* The data warehouse has to utilize a strategy for managing the incoming data.

The loading of staging tables requires that data is limited to an appropriate size for a given organizational need. He suggests five techniques to limit data – Refer to Figure 3-4 on pg. 75

* Time Stamped – Operational system time stamps records. The warehouse load routines simply look for a given range. This is extremely popular in many warehouse designs because it allows the load routine to only scan records of interest.
* Delta or Changed – The load routine checks for changes in the operational system and loads based on these changes. This is also an extremely popular method because only those records that have changed are touched. The more real-time the update on a warehouse, the more this method is used.
* Using Log or Audit Files from Transactional System – Operational systems (especially transactional systems) frequently produce audit files or logs. These can be used to identify changes in the operational data and further limit the need to scan through large data sets. While some designs use this approach, it is not as popular as the first two.
* Modify Application Code for Loading – This is never a good option. Production stability is dependent on a consistent set of load routines. Modifying the code every time new data is loaded introduces a significant opportunity for error. Again, while some designs have employed this option, it is rarely used due to this stability concern.
* Resource-Utilization or Operational System Snapshots – This is the last resort when nothing else works. It is cumbersome and complex. Snapshots of data from the operational system are compared in a before-and-after approach. The changes between the two produce a delta file that is in turn loaded to the warehouse. It would not be considered best practices.

To implement this design, we create a process model and a data model.

* The process model applies only to the operational system. Figure 3-7 on pg. 78 has a good relationship diagram of process and data models. As the name suggests, it provides an overall map of a business process.
* Process models typically consist of the elements functional decomposition, context-level zero diagram, data flow diagram, structural chart, state transition diagram, HIPO chart, and pseudocode. The reading will go into greater detail on each of these elements, but as you are reviewing the process model please post any questions to the forum.
* The data model applies to both the operational system and the data warehouse. It indirectly applies to department / data mart, and individual systems as well.
* The three levels of the data model are:
  + High-level or Entity Relationship Diagram (ERD). This is a high level view of the data that includes entities and relationships. For example, with a manufacturing company there may be entities of Part, Order, and Supplier. Orders have relationships to the Part and Supplier entities (a Part is part of an Order for a Supplier)
  + Mid-level or data item set (DIS) – This is where data begins to be grouped together. Using the same example, there may be different types of Orders that all belong together in a group. The relationship of these Orders would be outlined on the DIS.
  + Low-level or physical model – This is where the data outlined at a high level in the ERD, grouped together in the DIS, comes together to produce a data diagram showing fields, tables, and key fields that are used in joining data elements together.
* The process and data models serve very different purposes. You cannot simply swap one for the other when talking about warehouse design

Normalization is the final topic for this week.

* The most common uses of normal form are first, second and third.
  + 1NF – Data is unique in the table and there is a primary constraint
  + 2NF – Includes 1NF plus similar data is grouped together
  + 3NF – Includes 2NF plus data is not redundant. Foreign Keys are used to reference data in other tables where necessary. 3NF is a main concept in data warehouse design theory.
* Normalization is important because it reduces the amount of redundant data that is stored in the warehouse. As a result, the amount of effort to retrieve data from the warehouse is significantly reduced. It produces faster query times as well as less traffic across the hardware environment. For example, a description field may be normalized because it appears in multiple places but would be updated infrequently.
* Denormalization is also important because there are some situations where the architect would want to improve performance. Joining data between tables is often resource intensive. If a query required a large number of tables to be joined together to produce a description, then it may make sense to denormalize it to individual tables. That would allow the same information with fewer (less costly) joins.

This week’s reading is Inmon Chapter 3 and 4 and Hadoop Application Architectures

pages 38 to 77. Also review the deeper discussion on third normal form found on the techopedia web site link that is posted.

Please remember the general forum rules on posting. If you have any questions about the assignment, please post in the weekly thread.