Unit 8 Notes

Introduction to ETL

As we mentioned in the previous lecture, the ETL performs three major operations: (E) extracts the data from the data sources, (T) cleanses and conforms them, and (L) delivers them to the presentation server, to be loaded into the DB. Managing these operations can be considered a fourth major operation. Most ETL services are available today as off-the-shelf tools, which are a must for any serious DW/BI project. However, the ETL development still takes 70% of the time and effort to build the DW/BI system. It is particularly important that the ETL tools work with our key data sources – otherwise, we need to implement the extract code ourselves. Kimball splits the ETL functionality into 34 subsystems. Each particular DW may use a slightly different set.

ETL requirements and constraints

The main goal of the ETL component, as with all DW components, is to meet the DW business users' needs. Some were collected during the business requirement definition phase (Chapter 3), while others were collected during the data source investigation for data modeling (Chapter 7). The ETL team matches all these high-level requirements against the actual data. Some mismatches are to be expected, and both requirements and data may need adjustment.

The ETL system is subject to additional types of requirements and constraints. Legal **compliance** requirements, **security**, **archiving**, and **lineage** overlap each other. They are handled by Subsystem 32 (Security System, pp 415) and Subsystem 33 (Compliance Manager, pp 415-416). Meeting these requirements involves making data backup copies, following protocols, and documenting the algorithms and processes. Since serious security breaches are much more likely to come from within the organization than from the outside, role-based security is recommended for all ETL data and metadata. Compliance means "maintaining the chain of custody" of the data. Lineage, also known as data provenance, is specific metadata attached to each dataset to enable tracing its origins and subsequent transformations. Provenance helps users assess data and process authenticity, trust, and reproducibility. In the age of Big Data distributed and used collaboratively across the internet, data provenance has become a critical issue for private businesses and national governments alike.

ETL requirements related to data delivery to downstream components include **data latency** and **formatting**. Data latency is the throughput of the ETL system as part of the deployed DW/BI system. Meeting it involves choosing the right hardware architectures and software environments. **Real-time processing** is a common requirement in this area, coming from users who expect the DW to be updated continuously throughout the day. Such updates belong to three categories: **instantaneous**, **frequent**, and **daily**. "Instantaneous" means that the user screen responds instantly to all changes in the source system. This part is usually implemented as an Enterprise Information Integration (EII) solution. The source system itself is responsible for updating the users' screens and answering to their queries. "Frequent" means that the user screen is updated many times a day (e.g., every 15 minutes). This part is

usually implemented as micro-batches in typical ETL architecture. “Daily” updates of the users’ screens are implemented as batches in the typical ETL. In addition, real-time reporting usually relies on a **real-time partition**, which is physically separated from the other DW tables. Ideally, this is a true DB partition whose fact table is partitioned by date. The ETL must also deliver the data in the best **format** for the dimensional model and BI apps, which face the end users and have their own requirements and constraints. The ETL team, data modeling team, and BI team must work together to establish the data handoff protocol.

ETL Tools

In the past, ETL developers used to spend a significant amount of time developing hand written scripts and programs to extract, transform and load data into data warehouses. Recent versions of the ETL tools available in the market have considerably improved this process. They provide Graphical User Interfaces (GUI) with drag-and-drop capabilities that develop automated programs on the fly with several lines of code behind them. These automated programs significantly minimize the development time. The ETL tools also provide template options where ETL programs can be created once and reused in the future for multiple purposes. The newer tools are also providing an increased number and types of connectors to connect to various types of database systems (Oracle, SQL Server, Teradata, DB2, etc), ERP applications (SAP, Peoplesoft, etc), financial planning and reconciliation tools such as Hyperion, file formats (csv, xml, etc) and more.

The recent tools also provide version control capabilities and simplify the development process in a multi-user environment. The tools also provide options to capture log information associated with the data load process. The log information facilitates easier troubleshooting and also enables trend analysis to compare data processing times across time periods. The log information can also be used for reporting purposes. Common ETL tools include Informatica, IBM's DataStage, Oracle Data Integrator, SAP Data Services, and Ab Initio.

The ETL tools also enable structuring the data load processes in parallel or in a sequential fashion. For instance, multiple dimension tables can be set up to load in parallel and thus minimizing the data processing time. The tools provide options to set up dependencies. For example, the data load workflow process can be set up in such a way that loading of all dimension tables have to be completed prior to loading the fact table. The ETL tools come with scheduling options/features. The data load processes can thus be scheduled to start at specific times on specific days. However, some organizations prefer using dedicated scheduling tools to schedule ETL jobs.

Error handling and notification is another key important component of ETL design. There are various options to handle errors. The ETL process can be set up to immediately abort the data load process upon the first error or made to continue upon error with error records or records that failed to load written to a separate table for review and troubleshooting or set up to abort after certain number of errors. The ETL tools have capabilities to send email notifications to groups or personnel upon successful completion of processes or email alerts upon failures.

Prologue to ETL Design

For the ETL design process, the ETL architects and developers should review and understand the data model and the data architecture. The dimensional model helps them review the dimensions and facts. The data dictionary, if available, can be used as a starting point to document information about the sources for the various dimension tables, fact tables and their attributes/columns. If a data dictionary is not available, the source list can be created based on the dimensional model. The data model also helps create the **Source to Target Mapping** document (**STM**, aka Data Transformation Mapping (DTM)), which is important in ETL development. The STM helps document the necessary transformation rules or mapping rules while moving data from one layer in the DW architecture to another (source to staging, staging to ODS, ODS to DW, DW to DM). We will be creating an STM as part of Assignment 3.

The ETL design effort also involves identifying the best times for extracting the data from the various data sources, loading them into the various DW layers, the order in which the different tables should be loaded, among others. Source data extracted from various source systems become available for data extraction at different times. For example, sales information for Latin America region might be available from the operational systems overnight while the sales information for the Asia region might be available during the day time. Depending upon the reporting requirements and the availability of source data, ETL data extraction jobs have to be scheduled accordingly. The data extraction scheduling time should also consider a time when the extraction process will impose minimal strain on the source system performance. As long as there is minimal or no transformation logic involved in moving data from the source system to the staging area, source system performance is not impacted.

Typically, data loading processes have dependencies associated with them. For instance, the dimension tables typically have to be loaded prior to loading the fact tables. This helps avoid referential integrity constraint issues. Recent versions of the ETL tools have options to set up **workflows** or sequences to automatically load the various dimensions and fact tables in the preferred sequence or order.