Chapter 5

DW Development and ETL

Learning Objectives

- Explain data integration and the extraction, transformation, and load (ETL) processes
- Basic DW development methodologies
- Describe real-time (active) data warehousing
- Understand data warehouse administration and security issues

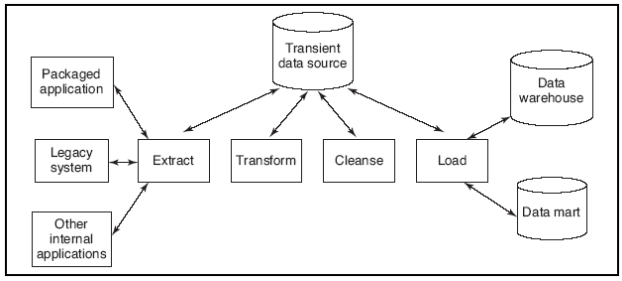
Data Warehouse Development

Eleven major tasks that could be performed in parallel for successful implementation of a data warehouse (Solomon, 2005) :

1. Establishment of service-level agreements and data-refresh requirements
2. Identification of data sources and their governance policies
3. Data quality planning
4. Data model design
5. ETL tool selection
6. Relational database software and platform selection
7. Data transport
8. Data conversion
9. Reconciliation process
10. Purge and archive planning
11. End-user support

Extraction, Transformation, and Load (ETL) Process

- Extracting data from outside sources
- Transforming it to fit operational needs (which can include quality levels)
- Loading it into the data warehouse

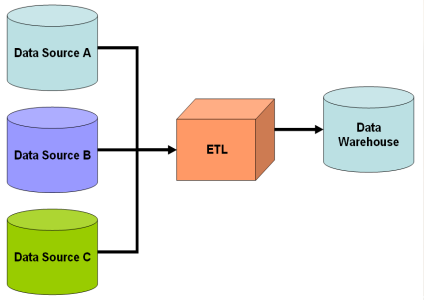


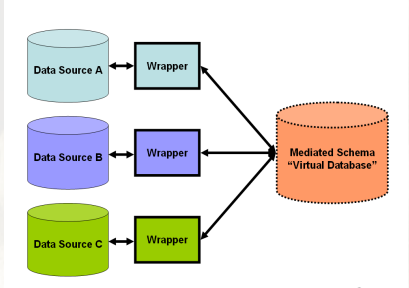
Extract (the E in ETL)

- Most data warehousing projects consolidate data from different source systems
  - Relational databases and flat files, Information Management System (IMS), Virtual Storage Access Method (VSAM), or even fetching from outside sources such as web spidering
- Each separate system may also use a different data organization / format.
- Data integration is required

Data Integration

- The process of combining data residing at different sources and providing the user with a unified view of these data
- Two approaches:





Enterprise application integration (EAI)

- Several applications in a business
  - Supply chain management (inventory/shipping), CRM (customers), etc
- Cannot communicate with one another in order to share data or business rules
  - islands of automation
- Inefficiencies:
  - identical data are stored in multiple locations and in different ways

EAI: definition

- EAI is the use of software and computer systems architectural principles to integrate a set of enterprise computer applications
- EAI links applications within a single organization together, while avoiding having to make sweeping changes to the existing applications

EAI: challenges

- Large challenges of EAI:
  - systems to be linked reside on different operating systems,
  - use different database solutions,
  - and different computer languages,
  - and in some cases are legacy systems that are no longer supported by the vendor who originally created them
- It is very hard to modify the systems in any way

EAI: how it works

- Mediation
  - EAI acts as broker between multiple applications. Whenever new information created, EAI propagates the changes to other relevant applications.
- Federation
  - EAI acts as covering facade across multiple applications. EAI system performs all interactions with the underlying applications on behalf of the requester.
- Both patterns are often used concurrently.
  - The same EAI system synchronizes multiple applications (mediation), while servicing requests from external users against these applications (federation).

EAI: connectivity technologies

- EAI connects to applications through a set of adapters:
  - Programs that know how to interact with an underlying application
- Adapters can be specific to an application or can interact with any application through a standard communication protocol, such as SOAP or SMTP.
- The adapter could reside in the same process space as the EAI or execute in a remote location through industry standard protocols such as message queues or web services.

EAI: pros vs. cons

- Pros
  - Single unified consistent access interface to various applications
  - Real time access to information access among various systems
- Cons
  - High Cost involved (especially for small and mid-sized businesses)
  - EAI implementations are very time consuming and resource intensive at times
  - In 2003 it was reported that 70% of all EAI projects fail

Enterprise information integration (EII)

- Data within an enterprise can be stored in various formats:
  - relational databases (in a large varieties), text files, XML files, spreadsheets, each with their own indexing and data access methods
- Standardized APIs to retrieve and modify data from a generic data source:
  - ODBC, JDBC, OLE DB, and ADO.NET
- Standard formats for representing data within a file, the best-known is XML, which has emerged as a standard universal representation format

EII: definition

- EII is the process of information integration, using data abstraction to provide a single interface for viewing all the data within an organization
- The goal of EII is to get a large set of heterogeneous data sources to appear to a user or system as a single, homogeneous data source

EII: how it works

- EII enables **loose coupling** between applications and heterogeneous-data stores
  - EII evaluates requests for information, queries the individual data sources, and delivers customized output to the requesting application
  - Applications access a view as if its data were physically located in a single database, even though individual data may reside in different source systems. EII transparently handles connectivity with back-end databases and applications

EAI vs. EII

- EAI focuses on linking applications together (typically in real time), whereas EII is focused on presenting a unified view of data.
- EII is viewed as a **by-product** of EAI, because, since the application processes are interlinked, the flow of information needs to be channeled for a unified view

EAI and EII software

- http://en.wikipedia.org/wiki/Comparison_of_business_integration_software

Transform (the T in ETL)

- Applies a series of rules or functions to the extracted data to derive the data for loading into the data warehouse
- After transformations, data meet the business and technical needs of the data warehouse

Transform: examples

- Selecting only certain columns to load
- Translating coded values (e.g., if the source system stores 1 for male and 2 for female, but the warehouse stores M for male and F for female), this calls for automated data cleansing; **no manual cleansing** occurs during ETL
- Encoding free-form values (e.g., mapping "Male" to "1" and "Mr" to M)
- Deriving a new calculated value (e.g., sale_amount = qty * unit_price)
- Filtering
- Sorting
- Joining data from multiple sources (e.g., lookup, merge)

Transform: examples

- Generating surrogate-key values
- Applying any form of simple or complex data validation.
  - If validation fails, it may result in a full, partial or no rejection of the data, and thus none, some or all the data is handed over to the next step, depending on the rule design and exception handling

Load (the L in ETL)

- Loads the data into the data warehouse
- Depending on the requirements of the organization, this process varies widely:
  - May overwrite existing information with cumulative, updated data every week,
  - May add new data in a historized form, for example, hourly.
- The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs

ETL cycle (overview)

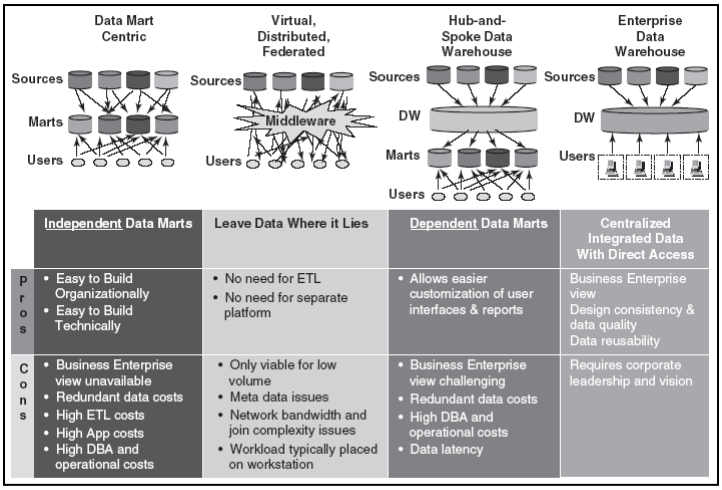
1. Cycle initiation
2. Build reference data
3. Extract (from sources)
4. Validate
5. Transform (clean, apply business rules, check for data integrity)
6. Stage (load into staging tables, if used)
7. Audit reports (for example, on compliance with business rules. Also, in case of failure, helps to diagnose/repair)
8. Publish (to target tables)
9. Archive
10. Clean up

ETL: real-life issues

- The slowest part of an ETL process usually occurs in the load phase
- Divide a big ETL process into smaller pieces and allow roll-back in case of failure
  - tag each data row with "row_id", and tag each piece of the process with "run_id", which to roll back and rerun the failed piece
- Tools:

  http://en.wikipedia.org/wiki/Extract,_transform,_load#Tools

Data Warehousing Development methodologies



	Independent Data Marts	Leave Data Where it Lies	Dependent Data Marts	Centralized Integrated Data With Direct Access
Pros	• Easy to Build Organizationally • Easy to Build Technically	• No need for ETL • No need for separate platform	• Allows easier customization of user interfaces & reports	Business Enterprise view Design consistency & data quality Data reusability
Cons	• Business Enterprise view unavailable • Redundant data costs • High ETL costs • High App costs • High DBA and operational costs	• Only viable for low volume • Meta data issues • Network bandwidth and join complexity issues • Workload typically placed on workstation	• Business Enterprise view challenging • Redundant data costs • High DBA and operational costs • Data latency	Requires corporate leadership and vision

FIGURE 5.6 Alternative Architectures for Data Warehousing Efforts

Data Warehouse Development

- Data warehouse development approaches
  - Inmon Model: EDW approach
  - Kimball Model: Data mart approach

Parameter	Data Mart (Kimball)	EDW (Inmon)
Approach	Bottom-up	Top-down
Architecture structure	Data marts with data bus and conformed dimensions for consistency	EDW feeds data marts
Complexity	Low	High
Data orientation	Process oriented	Subject oriented
Tools	Dimensional modeling	ER
Audience	End users	IT professionals
Objective	Ease end-users' access and reasonable response times	Sound technical solution based on database technology

Parameter	Data Mart (Kimball)	EDW (Inmon)
Scope	1 subject	Several subjects
Develop time	Months	Years
Develop cost	< $100K	> $1M
Develop difficulty	low-med	high
Size	< 100 GB	> 1 PB
Data transformations	low-med	high
Update freq.	Hour,day,week	Week,month
Num users	10s	100s
User types	Business analysts, managers	Enterprise analysts, senior executives
Business spotlight	Optimize activities within a business area	Cross-functional optimization and decision making

Data Warehouse Development

- Which model is best?
  - There is no one-size-fits-all strategy to data warehousing
  - One alternative is the **hosted** warehouse (BI service providers)
    - but privacy issues

Data Warehouse Development

- Massive data warehouses and scalability
  - The main issues pertaining to scalability:
    - The amount of data in the warehouse
    - How quickly the warehouse is expected to grow
    - The number of concurrent users
    - The complexity of user queries
  - Good scalability means that queries and other data-access functions will grow linearly with the size of the warehouse

Real-Time Data Warehousing

- **Real-time (active) data warehousing**

  The process of loading and providing data via a data warehouse as they become available

Real-Time Data Warehousing

- The need for real-time data
  - A business often cannot afford to wait a whole day for its operational data to load into the data warehouse for analysis
  - Provides incremental real-time data showing every state change and almost analogous patterns over time
  - Maintaining metadata in sync is possible
  - Less costly to develop, maintain, and secure one huge data warehouse so that data are centralized for BI/BA tools
  - An EAI with real-time data collection can reduce or eliminate the nightly batch processes

Real-Time Data Warehousing

- Levels of data warehouses:
  1. Reports what happened
  2. Some analysis occurs
  3. Provides prediction capabilities,
  4. Operationalization
  5. Becomes capable of making events happen

Real-Time Data Warehousing

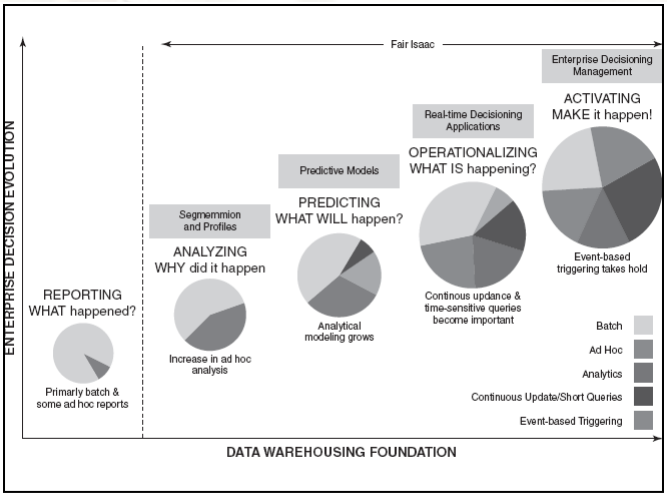


FIGURE 5.10 Enterprise Decision Evolution

Data Warehouse Administration and Security Issues

- **Data warehouse administrator (DWA)**

  A person responsible for the administration and management of a data warehouse

  Both technical and business skills are required (contrast to DBA)

Data Warehouse Administration and Security Issues

- Effective security in a data warehouse should focus on four main areas:
  - Establishing effective corporate and security policies and procedures
  - Implementing logical security procedures and techniques to restrict access
  - Limiting physical access to the data center environment
  - Establishing an effective internal control review process with an emphasis on security and privacy

WHAT'S NEXT?

Data Warehouse Design

- Data warehouse structure: The Star Schema
  - **Dimensional modeling**
    A retrieval-based system that supports high-volume query access
  - **Dimension tables**
    A table that address *how* data will be analyzed

Data Warehouse Design

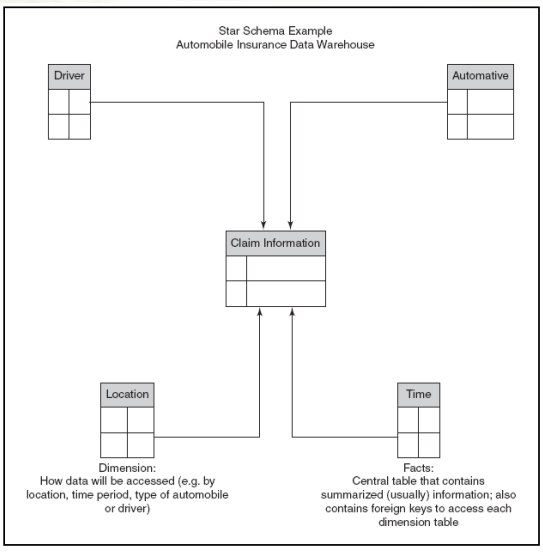


FIGURE 5.9 Star Schema