Introduction to ETL

Tathagata Bhattacharjee

iSHARE2 Support Team
Data Warehouse

• A data warehouse is a system used for reporting and data analysis.

• Integrating data from one or more different sources to create a central repository of data, a data warehouse

• A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data
Data Warehouse

• **Subject-Oriented**: A data warehouse can be used to analyse a particular subject area. For example, “HDSS Data” is a particular subject area.

• **Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying an event date, but in a data warehouse, there will be only a single way of identifying an event date.

• **Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 10 years, 5 years, ... This contrasts with a transactions system, where only the most recent data is kept.

• **Non-volatile**: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.
ETL

Source A
Files, Databases, Message Queues, Web Services

ETL
Read
Apply Logic
Write

Source B
Files, Databases, Message Queues, Web Services

Target

iSHARE 2
an INDEPTH project
Pre ETL Tasks

• **Profiling** the data:
  
  – Know your data
  
  – Will give direct insight in the data quality of the source systems.
  
  – Find out how many rows have missing or invalid values, or what is the distribution of values in a specific column.

• This knowledge will help to specify rules (data standards / quality checks) in order to cleanse the data and to keep really bad data out of the repository.

• Doing data profiling before designing the ETL process helps you to better design a system that is correct, robust and has a clear structure.
ETL

• Data moved from source to target databases

• Focus: Preparing the data for reporting / analysis

• ETL = Extract -> Transform -> Load
  
  – **Extract:** Get the data from source(s) as efficiently as possible
  
  – **Transform:** Perform calculation / map data / clean data
  
  – **Load:** Load data to the target storage
Value Addition

• Removes mistakes and corrects data
• Documented measures of confidence in data
• Capture the flow of transactional data
• Adjust data from multiple sources to be used together
• Structures data to be useable by any analysis tool
• Enables subsequent analytical data processing
Dirty Data

- Absence of Data / Missing Data
- Multipurpose Fields
- Cryptic Data
- Contradicting Data
- Violation of Data Rules
- Reused Primary Keys
- Non-Unique Identifiers
- Data Integration Problems
Data Cleaning
Data Cleaning: Parsing / Combining

- **Parsing** locates and identifies individual data elements in the source files and then *isolates* these data elements in the target files.
  - Examples include full name stored in one column

- **Combining** locates and identifies individual data elements in the source files and then *combines* these data elements in the target files
  - Examples include event date stored in different columns as date, month and year
Data Cleaning: Correcting

- Correct parsed individual data components using sophisticated data algorithms and secondary data sources.

- Correct data according to data rules

- Example includes converting the combined date into a standard date format
Data Cleaning: Standardizing

- **Standardizing** applies conversion routines to transform data into its preferred (and consistent) format using both standard and custom data rules.
Data Cleaning: Matching

- Searching and matching records within and across the parsed, combined, corrected and standardized data based on predefined data rules to eliminate duplications, sequences.
Data Cleaning: Consolidating

- Analysing and identifying relationships between matched records and **consolidating**/merging them into correct representation

<table>
<thead>
<tr>
<th>Migration Dates</th>
<th>Sequence</th>
<th>Event</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-05-09</td>
<td>1</td>
<td>Inmigration</td>
<td>1995-06-06</td>
</tr>
<tr>
<td>1995-06-06</td>
<td>2</td>
<td>Outmigration</td>
<td>2006-05-09</td>
</tr>
</tbody>
</table>
Data Staging

- Used as an interim step between data extraction and later steps
- Accumulates data from asynchronous sources using native interfaces, flat files, FTP sessions, or other processes
- Data in the staging file is transformed and loaded to the warehouse
- There is no end user access to the staging file
- An operational data store may be used for data staging
Data Transformation

• Transforms data in accordance with the data rules and standards that have been established

• Example include: format changes, splitting up fields, replacement of codes, derived values, and aggregates
ETL Tools

• Few popular commercial and freeware(open-sources) ETL Tools

• Commercial ETL Tools:
  – IBM Infosphere DataStage
  – Informatica PowerCenter
  – Oracle Warehouse Builder (OWB)
  – Oracle Data Integrator (ODI)
  – SAS ETL Studio
  – Business Objects Data Integrator (BODI)
  – Microsoft SQL Server Integration Services (SSIS)
  – Ab Initio

• Freeware, open source ETL tools:
  – Pentaho Data Integration (PDI) - Kettle
  – Talend Integrator Suite
  – CloverETL
  – Jasper ETL
PDI Key Features

- drag-and-drop data integration
- graphical extract-transform-load (ETL) designer
- rich library of pre-built components to access, prepare, and blend data from various sources
- integrated enterprise scheduler for coordinating workflows
- agile views for modeling and visualizing data on the fly
Thank you