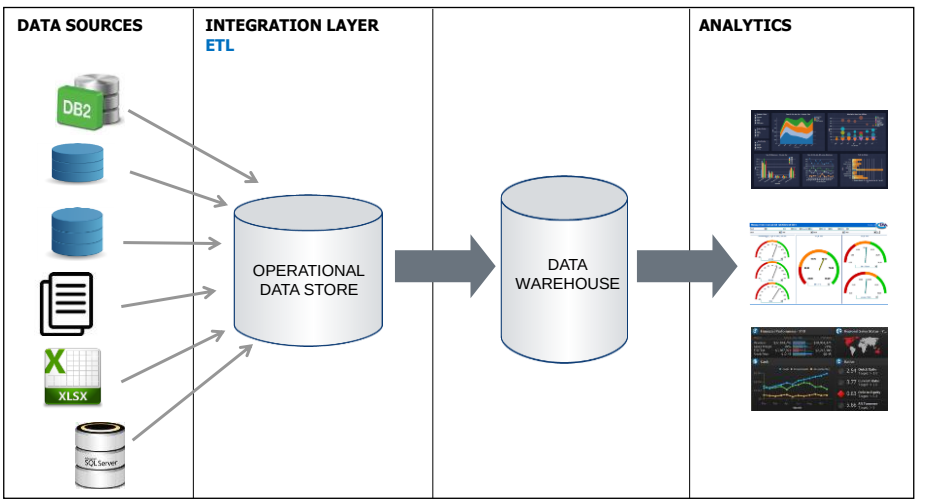
POZNAN UNIVERSITY OF TECHNOLOGY

DW Loading and Refreshing Techniques: ETL

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Poznań, Poland
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

ETL in DWS architecture



Developing ETL

- Designing and developing ETL processes
  - critical for DW functioning
  - challenges
    - data quality
    - data freshness
    - performance of ETL execution (time window for a DW refreshing)
    - source evolution
    - ETL optimization
  - costly (time & money)
    - up to 70% project resources
      - staff
      - hardware
      - software

Developing ETL

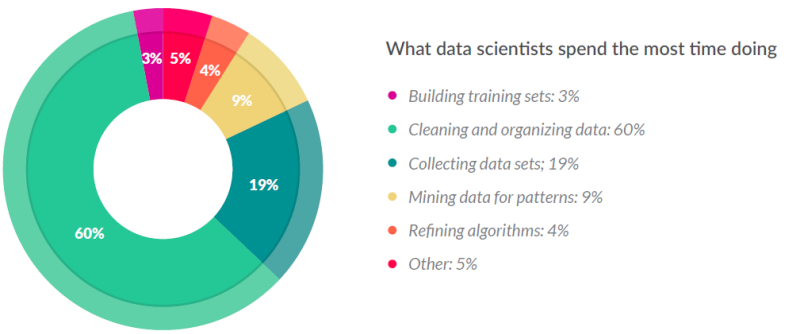
- Gartner Report on DW projects in financial institutions from the Fortune 500 list
  - 100 of staff in a DW project
  - 55 ETL
  - 17 system admins (DB, hardware)
  - 4 system architects
  - 9 BI consultants
  - 5 programmers
  - 9 managers
  - hardware (multiproc. servers, TB disks, 5mln USD)
  - ETL software (1mln USD)
  - # data sources: 10 to 50

Developing ETL

- # data sources to integrate
  - large banks: hundreds
- Types of data sources to integrate
  - databases (all possible)
  - text files
  - spreadsheets
  - streaming data (more and more frequently)

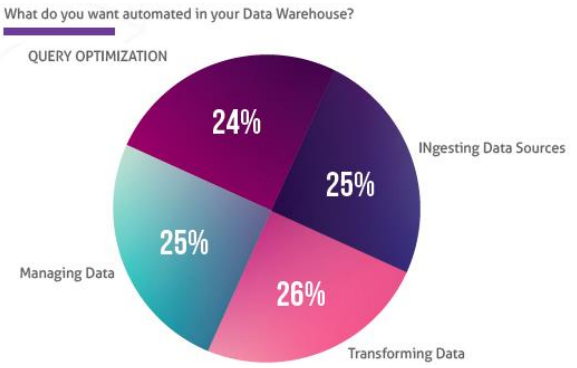
Developing ETL

- Data Science Report. 2016, CrowdFlower

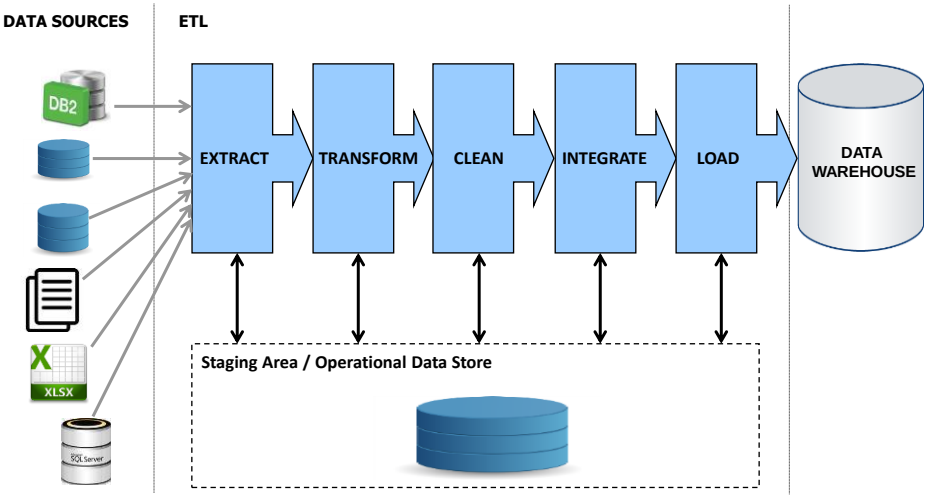


Developing ETL

➲ **Panoply Data Warehouse Trends Report 2018**

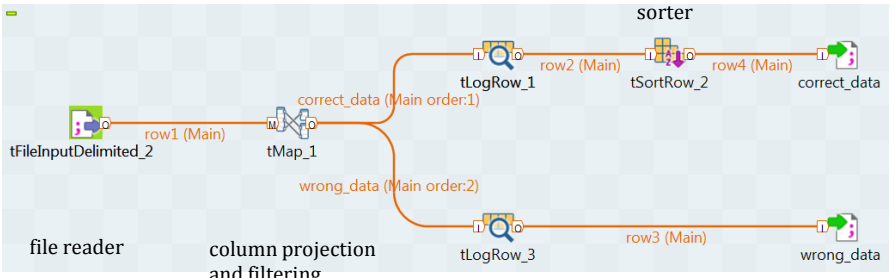


ETL architecture



Example ETL process

- Talend Open Studio



DW designing

- Analysis of available data sources
- Deciding on DS access technologies (see Topic 1)
- Data profiling
- Data ingest
  - full
  - incremental
- Transforming
- Cleaning and homogenizing
- Merging
- Duplicate elimination
- Uploading into a DW

Data sources

- Identify relevant DSs
- DS description
  - business area (e.g., HR, payroll, sales, loans, marketing, ...)
  - importance
  - business user
  - business owner
  - technical/infractructure owner
  - hardware + OS
  - DBMS
  - schema
  - # transacions/day
  - data volume increase/day
  - DB size

Typical predefined connectors

- IBM DB2
- SQL Server
- Oracle
- Sybase ASE, IQ
- Netezza
- Vertica
- Teradata
- SAS
- SAP Hana
- Greenplum

- PostgreSQL
- MySQL
- SQLite
- FireBird
- ODBC data source
- JDBC data source
- Excell
- Access
- Text, XML, JSON files
- Hive
- Impala
- MongoDB
- Cassandra
- ...

Data profiling

- Analyzing data sources
- Main categories of tasks
  - structure discovery (schema, relationships)
  - content analysis (data values, data quality, daily size increase)
  - relationship discovery
- Application areas
  - ETL for DW
  - data conversion and migration
  - data quality analysis in production DSs
- Tools
  - statistics
  - data mining

Data profiling

- Data types and allowed lengths
- Discovering schema
  - UNIQUE attributes
  - PK candidates
  - FK candidates
  - functional dependencies
  - embedded value dependencies (if a denormalized schema)
- Statistics on data
  - min, max, count, avg, distinct, variance, stdev
- Computing data distributions (histograms)
- Assessing costs of potential joins

Data profiling

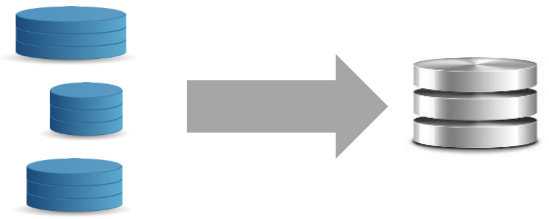
- Discovering data quality
  - identify NULL/NOT NULL columns
  - count #rows with NULL or default value for each attribute
  - identify valid allowed values for attributes
  - identify domains of attributes
  - count #records with values other than expected
  - discover value formats (date, phone No, address, ...)
  - discover outliers
  - discover wrong values
  - % of: missing values, typos, non-standardized values

Data profiling tools

- Open source
  - Quadient DataCleaner
  - Aggregate Profiler
  - Talend Open Studio for Data Quality
  - Melissa Data Profiler
- Commercial
  - IBM InfoSphere Information Analyzer
  - SAP Business Objects Data Services for Data Profiling
  - Informatica Data Profiling Solution – Data Explorer
  - Oracle Enterprise Data Quality
  - SAS DataFlux

Loading data into DW

- Reading the whole data source
  - text/binary dump files ⇨ DB export
  - XML files
  - SQL select + gateway / ODBC
  - snapshots
- Reading changes
  - need to detect data changes



Detecting data changes

- Requirements
  - minimum or none source system changes
  - minimum interference with a data source
- Solutions
  - audit columns
  - snapshot comparison
  - system maintained log of changes on a table (e.g., snapshot log)
  - snapshots
  - triggers ⇨ synchronous transfer
  - analysis of a redo log (transaction log)
    - periodically (log scraping)
    - on-line - continuously (log sniffing)

Snapshot/replica

- Copy of a table or a subset of its columns and rows
- Refreshing
  - automatic with a defined interval
  - on demand
- SQL Server
- IBM DB2
- Oracle

Data transformation

- Transform to a common data model
  - relational
  - object-relational
  - semistructured
  - NoSQL
  - graph
  - ...
- Transform semantically identical data to a common (the same) representation
  - extract text values (e.g., parts of an address)
- Remove unnecessary columns

Data transformation

➲ **Merge semantically identical records**

TaxpayerID	Fname	Lname	Country	Title	Domain
132018	Robert	Wrembel	Poland	Prof.	data wrehouses

+

TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968

TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born	TaxpayerID	Country	Title	Domain
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968	132018	Poland	Prof.	data wrehouses

Fname	Lname	Country	Title	Domain
Robert	Wrembel	Poland	Prof.	data wrehouses

1sName	2ndName	Last_Name	Education	City	Born
Robert	Andrew	Wrembel	univ.	Poznań	1968

do the records represent the same entity?

Data transformation

➲ **Requirements**

- **iterative and interactive process**
  - **define transformation**
  - **run process**
  - **verify resutls**
  - **modify transformation (if needed)**
  - **run process**
  - **...**
- **extendible and easy to modify**
- **as much data as possible should be transformed automatically**
- **as much steps as possible should be automatic**

Data cleaning

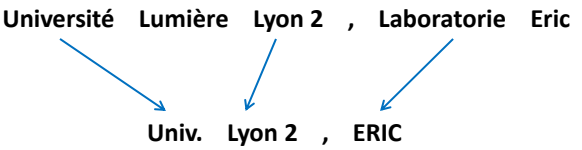
- Remove/replace null values
- Correct typos
  - dictionaries (spelling, names, cities, countries)
- Correct semantical errors
  - gross = net + vat
  - address consistent with ZIP code
- Standardize values
  - date format
  - currency
  - capital/small letters
  - abbreviations
  - synonyms (Word Net)

Data deduplication

- Records must be cleaned
  - no special signs, no punctuations
  - no abbreviations
- Problem: how to decide if 2 records represent the same entity?
  - {Wrembel, Robert, ul. Wyspiańskiego, Poznań}
  - {Wrębel, Robert, ul. Wyspiańskiego, Poznań}
- Case 1: natural identifiers (e.g., ID, email, mobile number) available
- Case 2: no natural identifiers available
  - approximate/probabilistic decision based on a **similarity measure**

Data deduplication

- Simple similarity measure
  - # matching atomic tokens (text strings) / # total atomic tokens in compared records
  - # total atomic tokens: 8
  - # matching atomic tokens: 3
  - similarity=3/8



Data deduplication

- Soundex
  - returns a code of pronunciation of an input
  - soundex('Smith')=soundex('Smit')=S530
- Levenhstein/edit distance
  - minimum number of inserts and deletes (updates) of characters in order to convert L1 to L2
  - L1 and L2 identical ⇨ didtance=0
  - ABC ⇨ ABCDEF: distance=3
  - DEFCAB ⇨ ABC: distance=5

DW refreshing

- When?
  - synchronous (after a source transaction was committed) ⇨ (near) real-time DW
  - asynchronous ⇨ traditional DW
    - with a defined frequency
    - on demand
- How?
  - full (1st DW load)
  - incremental (all next loads)
- How data arrive?
  - batch ⇨ traditional DW
  - stream ⇨ (near) real-time DW

DW refreshing

- In a constraint time window (typically 8h)
  - SSD throughput: 500MB/s → 1TB in approx. 35min
  - Magnetic disc throughput: 100MB/s → 1TB in approx. 3h
- Efficiency is crucial
  - read from DSs only necessary data
- Do not execute in a DS
  - sorting
    - DISTINCT
    - set operators
    - GROUP BY
  - NOT and non-equijoins (typically require full scan)
  - functions in the WHERE clause

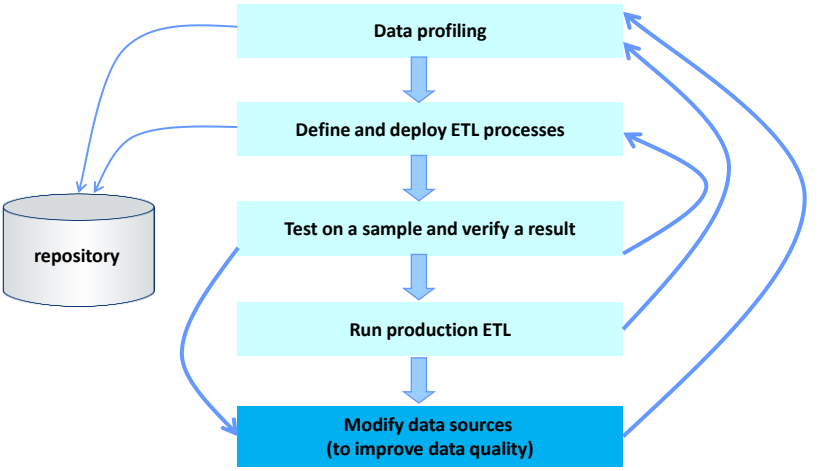
DW refreshing

- Where to filter data?
  - at a data source, if
    - not overloaded with its proper processing
    - powerful query optimizer
    - good use of indexes
  - in an ETL layer, otherwise
    - sorting in a database
    - sorting in an OS (awk)
- Separate inserts from updates
  - updates → standard path
  - inserts → direct load path
- Decide how to maintain additional data structures
  - indexes
  - materialized views
- Integrity constraints in a DW?

DW refreshing

- Parallel loading
- Collecting DB statistics after refreshing
- DB defragmentation

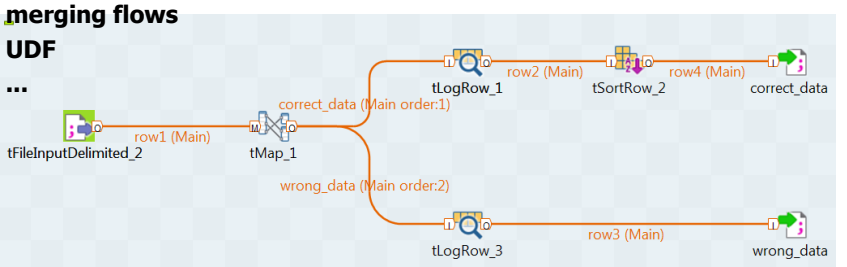
Summary: ETL design process



- Jarke M., et. al.: Improving OLTP Data Quality Using Data Warehouse Mechanisms. SIGMOD Record, (28):2, 1999

Implementing ETL

- **ETL → workflow → graph of tasks connected by flows**
- **Typical tasks**
  - **aggregation (max, sum, ...)**
  - **filtering**
  - **join**
  - **look-up**
  - **sequence generation**
  - **sorting**
  - **splitting flows**
  - **merging flows**
  - **UDF**
  - **...**



Metadata

- On data sources
- On ETL processes
- On data warehouse

- On data sources
  - location (IP address)
  - hardware + operating system
  - type (RBD, OBD, XML, spreadsheet, ...)
  - schema
  - access methods (SQL, XQuery, dump file, ...)
  - connection credentials
  - results of data profiling
  - volume
  - performance characteristics

Metadata

- On ETL
  - data storage architecture of ODS and DW (e.g., disk capacities, row-store / column-store)
  - metadata on a dataset to upload to DW (e.g., size, avg. record lengths)
  - definitions of ETL tasks/steps
  - available dictionaries (e.g., cities, zip codes, names)
  - workflow execution schedules
  - execution statistics (e.g., elapsed time, CPU time, #I/O, RAM usage, throughput, disc access conflicts, #records uploaded, #records rejected)
  - dependencies between tasks for impact analysis
  - mappings between DS and DW structures
  - data lineage
  - execution logs

Requirements for ETL

- Efficiency
  - finishing in a predefined time window
  - estimating execution termination
- Optimizable
- Fault-tolerance
  - restart after removing errors from a break point
  - restart from the beginning
  - recovery after crash
- Manageability
  - scheduling executions
    - time-based
    - token-based
  - stopping and restarting tasks
  - impact analysis
  - easy modifiable workflows

Requirements for ETL

- Producing high quality data
- Security: access control
- A palette of predefined steps
- Automatic code generation
- Support of UDFs
- Automatic reporting on termination, errors, exceptions, and progress
- Parallel processing
- Direct path loading
- Semi-automatic adjustment to DS changes

Off-the-shelf vs. in-house

- Off-the-shelf
  - faster design and deployment
  - integrated data repository
  - metadata management
  - workflow execution scheduling
  - built-in drivers to multiple DSs
  - impact analysis
  - incremental data loading
  - parallel processing
  - price
  - often require more advanced architectures → cost
- In-house-developed
  - longer design and development
  - thorrough testing
  - dedicated to a given scenario
  - not customizable/flexible
  - may be tuned to a given scenario
  - may be less expensive

Off-the-shelf technology?

- Financial Times (18 Sep, 2013): Why big IT projects crash
- www.ft.com
- "... the Texas state auditor's office examined 13 IT projects, nine of which had overrun. It concluded, admittedly on a small sample, that agencies using commercial off-the-shelf technology exceeded their budgets by a smaller amount and took less time to complete their projects than those that did not"

Gartner Report

- **Commercial**
  - IBM Data Stage
  - Informatica
  - Microsoft Integration Services
  - ABInitio
- **Open-source**
  - Talend Open Studio
  - Pentaho Data Integration
  - CloverETL
  - Apache NiFi

Source: Gartner (July 2018)