Overview

- Understand, explain and discuss ETL
- Create the first SSIS project.

ETL Process

- Extract, Transform, The ETL Process Load (ETL)

Original slides were written by Torben Bach Pedersen
Aarborg University (Denmark)

Improving Data Quality

- Appoint “data quality administrator”
- Responsibility for data quality
- Includes manual inspections and corrections!
- Source-controlled improvements

  The optimal?
- Construct programs that check data quality

  Are totals as expected?

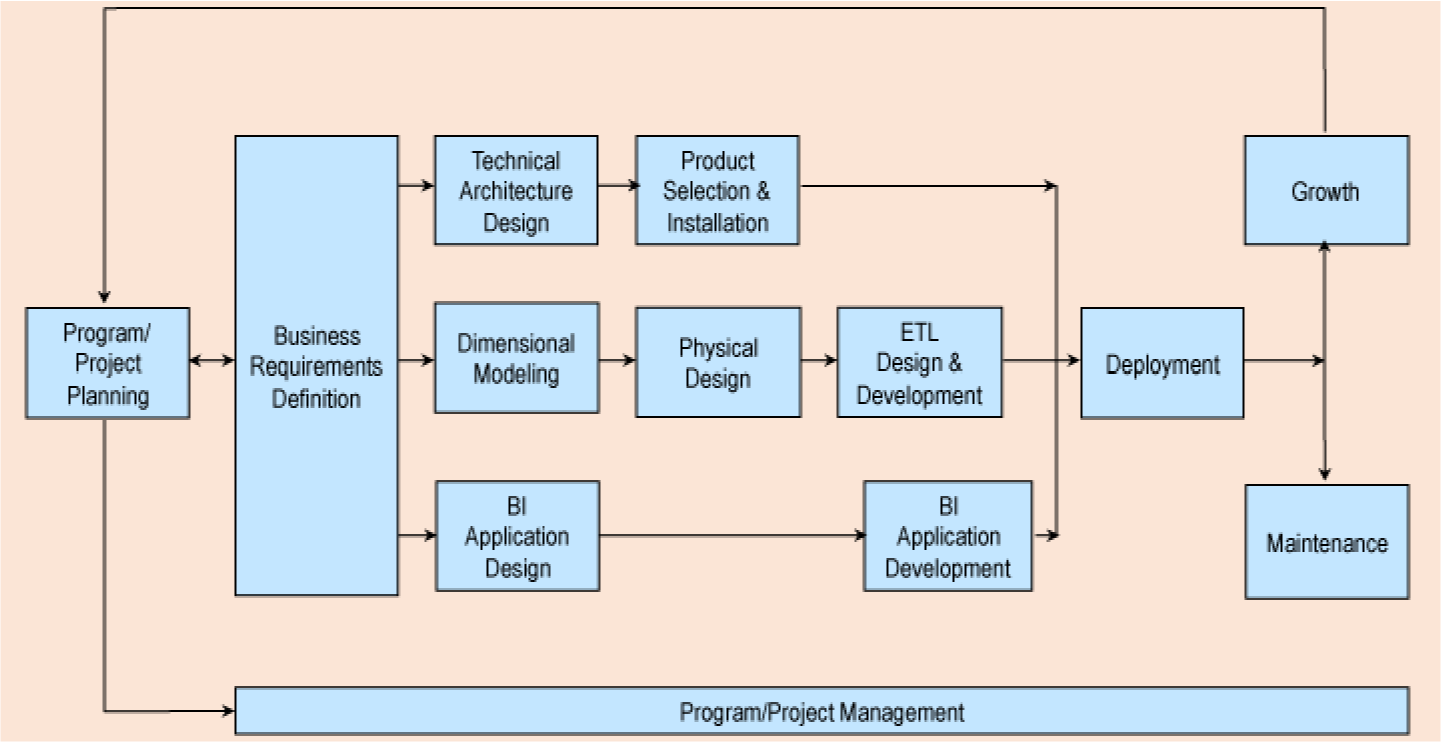
  Do results agree with alternative source?

  Number of NULL values?
- Do not fix all problems with data quality

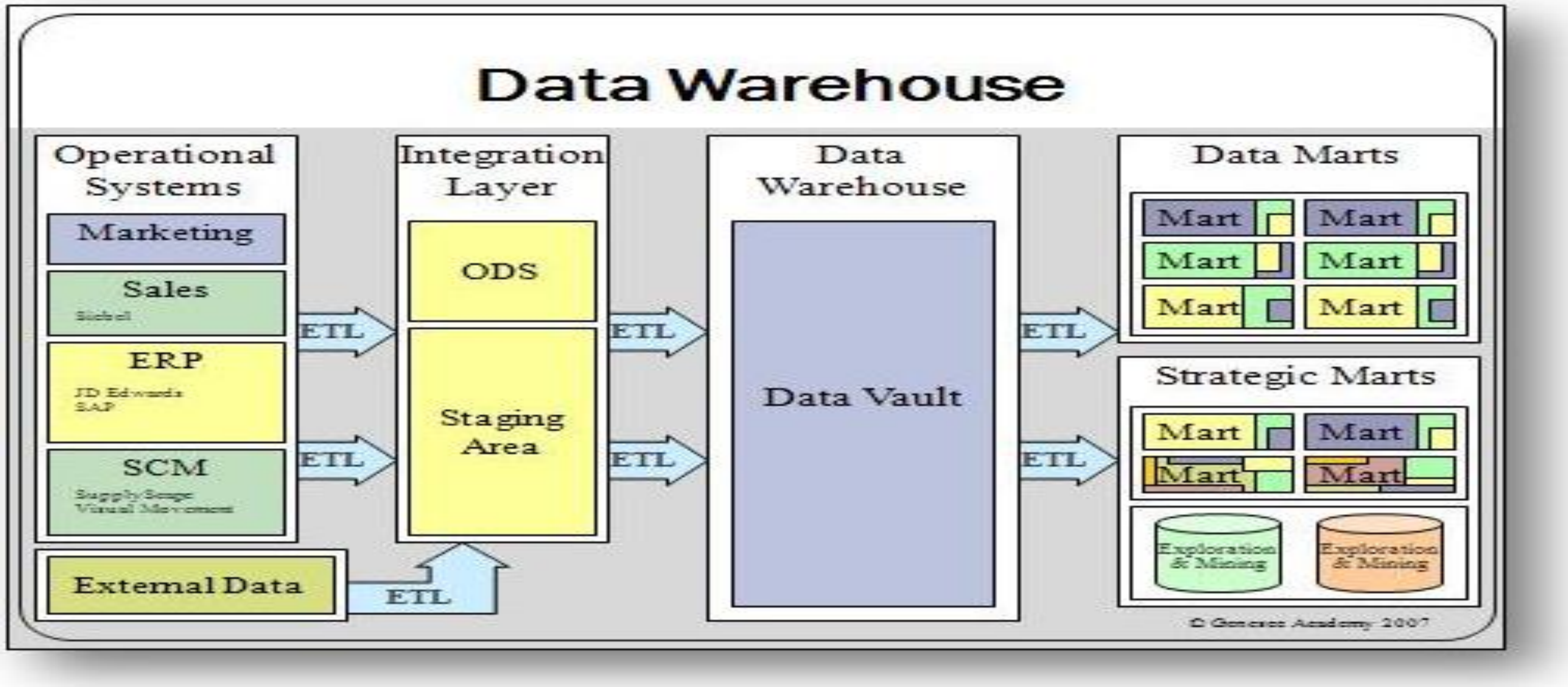
  Allow management to see “weird” data in their reports?

  Such data may be meaningful for them? (e.g., fraud detection)

Life cycle of Data warehouse Design


Program/ Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Design
BI Application Development
Deployment
Growth
Maintenance
Program/Project Management

Executing Environment



The ETL Process

- The most underestimated process in DW development
- The most time-consuming process in DW development

  80% of development time is spent on ETL!

  - Extract

    Extract relevant data
  - Transform

    Transform data to DW format

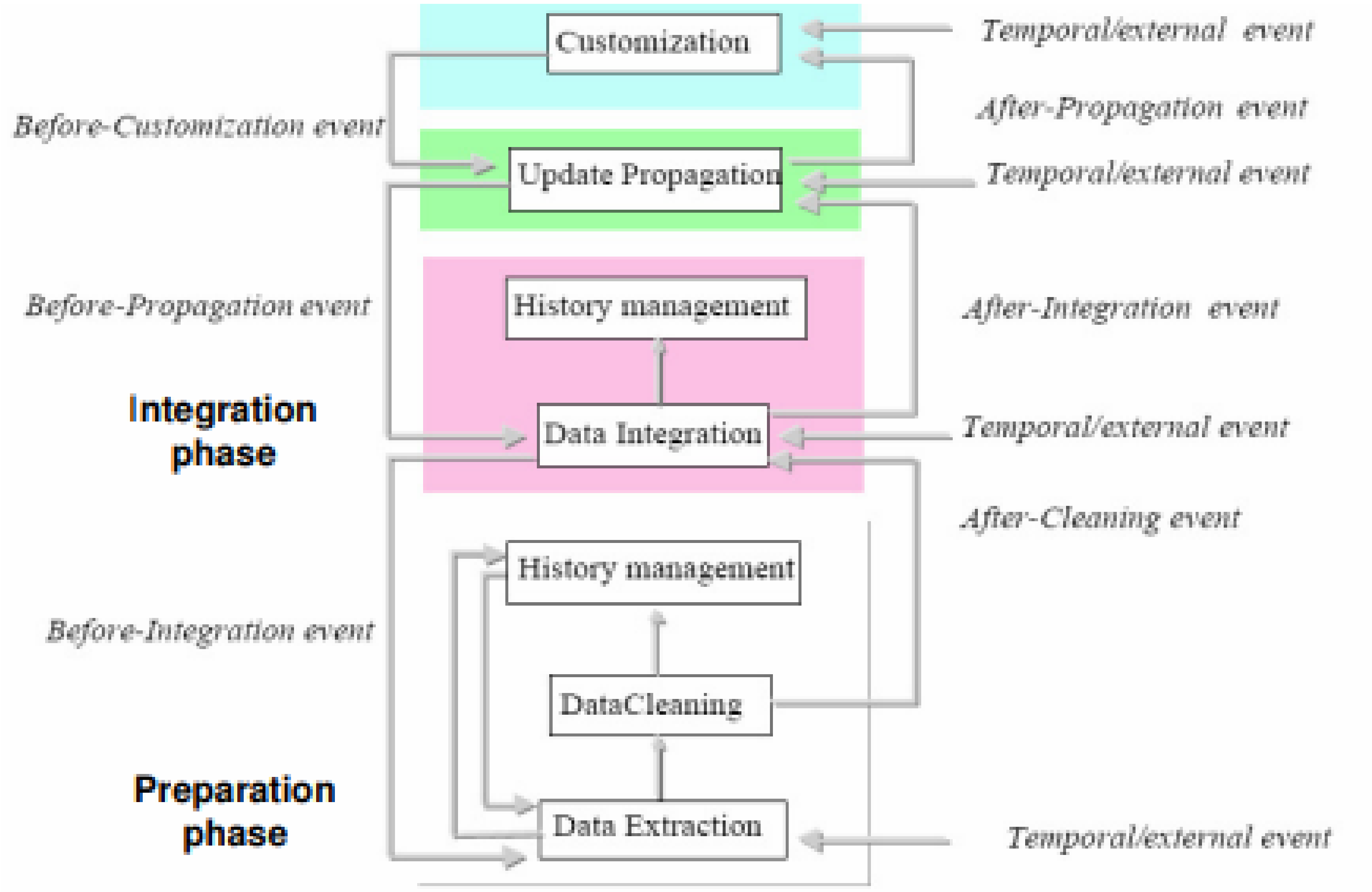
    Build keys, etc.

    Cleansing of data
  - Load

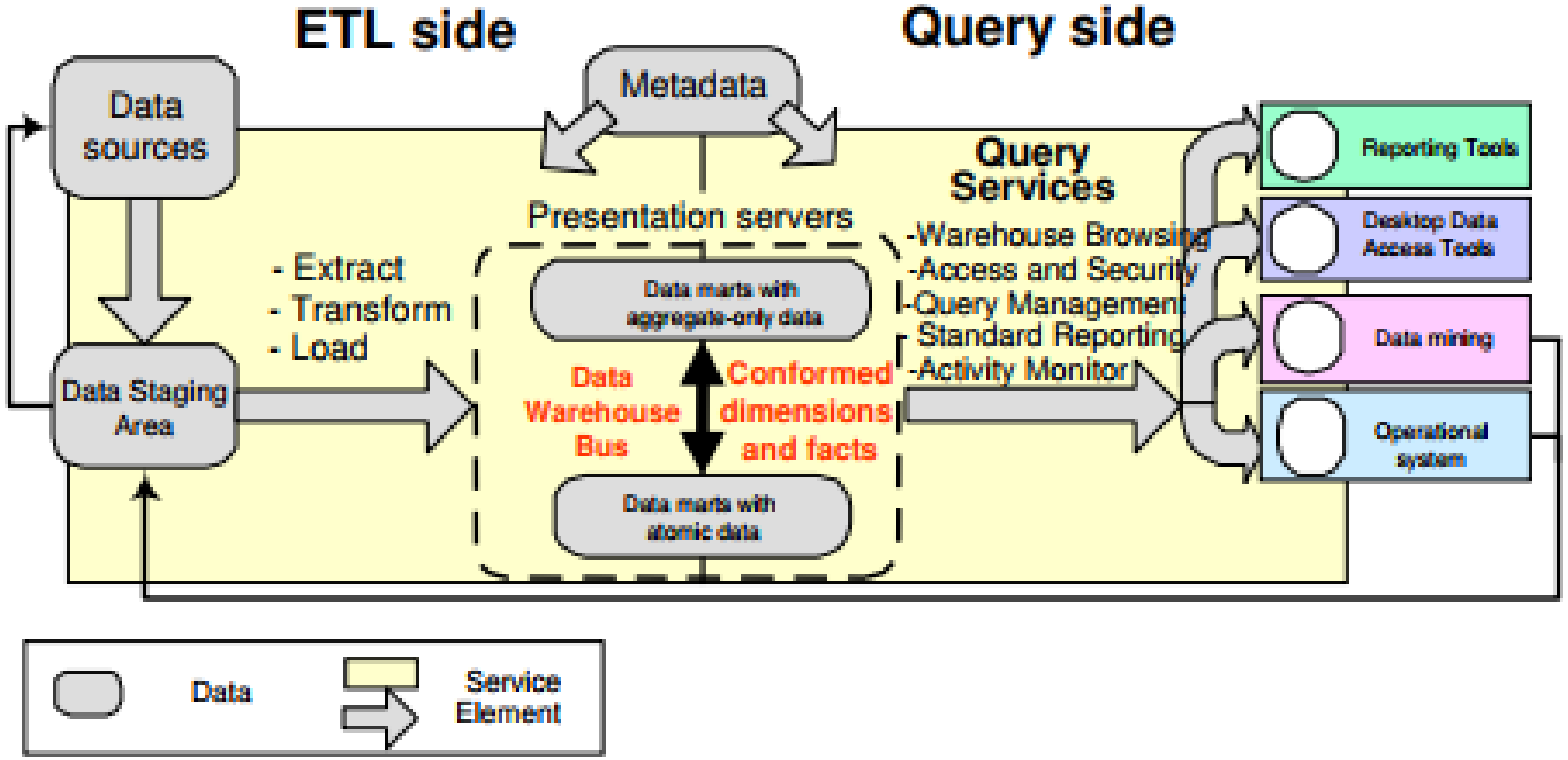
    Load data into DW

    Build aggregates, etc.

Refreshment Workflow – (flow upwards)



ETL in the Architecture



Data Staging Area (DSA)

- Transit storage for data in the ETL process
  - Transformations/cleansing done here
  - No user queries
  - Sequential operations on large data volumes
  - Performed by central ETL logic
  - No need for locking, logging, etc.
  - RDBMS or flat files? (DBMS have become better at this)
  - Finished dimensions copied from DSA to relevant marts
  - Allows centralized backup/recovery
- Often too time consuming to initial load all data marts by failure
  - Backup/recovery facilities needed
  - Better to do this centrally in DSA than in all data marts

- Often too time consuming to initial load all data marts by failure
Initial load means blank out data mart and load from scratch
Vs
Incremental load/update, much faster but can have many problems; i.e. missing an update

ETL Construction Process

Plan

1) Make high-level diagram of source-destination flow

2) Test, choose and implement ETL tool

3) Outline complex transformations, key generation and job sequence for every destination table
Construction of dimensions

4) Construct and test building static dimension

5) Construct and test change mechanisms for one dimension

6) Construct and test remaining dimension builds Construction of fact tables and automation

7) Construct and test initial fact table build

8) Construct and test incremental update

9) Construct and test aggregate build (you do this later)

10) Design, construct, and test ETL automation

production keys -> tell us where the data was produced; can trace back to origin

Building Dimensions

Exam Q: types of changing dimensions
0, 1, 2, 3

- Static dimension table
- DW key assignment: production keys to DW keys using table
- Combination of data sources: find common key?
- Check one-one and one-many relationships using sorting
- Handling dimension changes
- Find the newest DW key for a given production key
- Table for mapping production keys to DW keys must be updated
- Load of dimensions
- Small dimensions: replace
- Large dimensions: load only changes

Building Fact Tables

- Two types of load
- Initial load

  ETL for all data up till now

  Done when DW is started the first time

  Very heavy - large data volumes
- Incremental update

  Move only changes since last load

  Done periodically (e.g., month or week) after DW start

  Less heavy - smaller data volumes
- Dimensions must be updated before facts

  The relevant dimension rows for new facts must be in place

  Special key considerations if initial load must be performed again

Types of Data Sources

- Non-cooperative sources u do all the work; i dump data, u do conversion, etc.

  Snapshot sources – provides only full copy of source, e.g., files

  Specific sources – each is different, e.g., legacy systems

  Logged sources – writes change log, e.g., DB log

  Queryable sources – provides query interface, e.g., RDBMS

- Cooperative sources u tell me what u want, i produce in that format for u

  Replicated sources – publish/subscribe mechanism

  Call back sources – calls external code (ETL) when changes occur

  Internal action sources – only internal actions when changes occur

  DB triggers is an example

- Extract strategy depends on the source types

Extract

- Goal: fast extract of relevant data
- Extract from source systems can take long time
- • Types of extracts:
- Extract applications (SQL): co-existence with other applications DB unload tools: faster than SQL-based extracts
- Extract applications the only solution in some scenarios
- Too time consuming to ETL all data at each load

  Extraction can take days/weeks

  Drain on the operational systems

  Drain on DW systems

=> Extract/ETL only changes since last load (delta)

Data Capture Mechanisms

make data avail to DWH

MQ - back ups, will not lose sth if u put it into the queue

- Messages
- Applications insert messages in a “queue” at updates

+ Works for all types of updates and systems \

- Operational applications must be changed+operational overhead

- DB triggers

Triggers execute actions on INSERT/UPDATE/DELETE

+ Operational applications need not be changed

+ Enables real-time update of DW

- Operational overhead

- Replication based on DB log

Find changes directly in DB log which is written anyway

+ Operational applications need not be changed

+ No operational overhead

-Not possible in some DBMS

Conversions

- Data type conversions

ep-cid-dic

EBCDIC ASCII/UniCode

- String manipulations
- Date/time format conversions
- Normalization/denormalization
- To the desired DW format to canonical form, std form for presenting the data

Depending on source format • Building keys

Table matches production keys to surrogate DW keys

Correct handling of history - especially for total reload

Data Quality

- Data almost never has decent quality
- Data in DW must be:

e.g. if have 10,10,10 sum should show up as 30 in dwh, if see 40 000, there is a problem for example

- Precise DW data must match known numbers - or explanation needed
- Complete
- DW has all relevant data and the users know
- Consistent
- No contradictory data: aggregates fit with detail data
- Unique
- The same things is called the same and has the same key (customers)
- Timely
- Data is updated ”frequently enough” and the users know when

Types Of Cleansing

Conversion and normalization

- Text coding, date formats, etc.
- Most common type of cleansing

Special-purpose cleansing

- Normalize spellings of names, addresses, etc.
- Remove duplicates, e.g., duplicate customers

Domain-independent cleansing Domain – independent does not require knowledge of the domain

- Approximate, “fuzzy” joins on records from different sources

Rule-based cleansing

User-specifed rules, if-then style

Automatic rules: use data mining to find patterns in data

Guess missing sales person based on customer and item

Cleansing

- Mark facts with Data Status dimension
- Normal, abnormal, outside bounds, impossible,…
- Facts can be taken in/out of analyses
- Uniform treatment of NULL

Use explicit NULL value rather than “special” value (0,-1,…)

Use NULLs only for measure values (estimates instead?)

Use special dimension keys for NULL dimension values

Avoid problems in joins, since NULL is not equal to NULL

Mark facts with changed status

New customer, Customer about to cancel contract, ……

Load – How to speed up

- Goal: fast loading into DW
- Loading deltas is much faster than total load
- SQL-based update is slow

  Large overhead (optimization, locking, etc.) for every SQL call
- DB load tools are much faster oracle, sql server, etc. all have their own db load tools
  - Index on tables slows load a lot
    - Drop index and rebuild after load
    - Can be done per index partition
- Parallellization

  Dimensions can be loaded concurrently

  Fact tables can be loaded concurrently

  Partitions can be loaded concurrently

Load – Preserving relationships

- Relationships in the data
- Referential integrity and data consistency must be ensured (Why?)
- Can be done by loader
- Aggregates

  Can be built and loaded at the same time as the detail data
- Load tuning

  Load without log

  Sort load file first

  Make only simple transformations in loader

  Use loader facilities for building aggregates
- Should DW be on-line 24*7?

  Use partitions or several sets of tables (like MS Analysis)

Microsoft Integration Services

- Microsoft’s ETL tool

Part of SQL Server

- Tools

  Import/export wizard - simple transformations

  BI Development Studio – advanced development

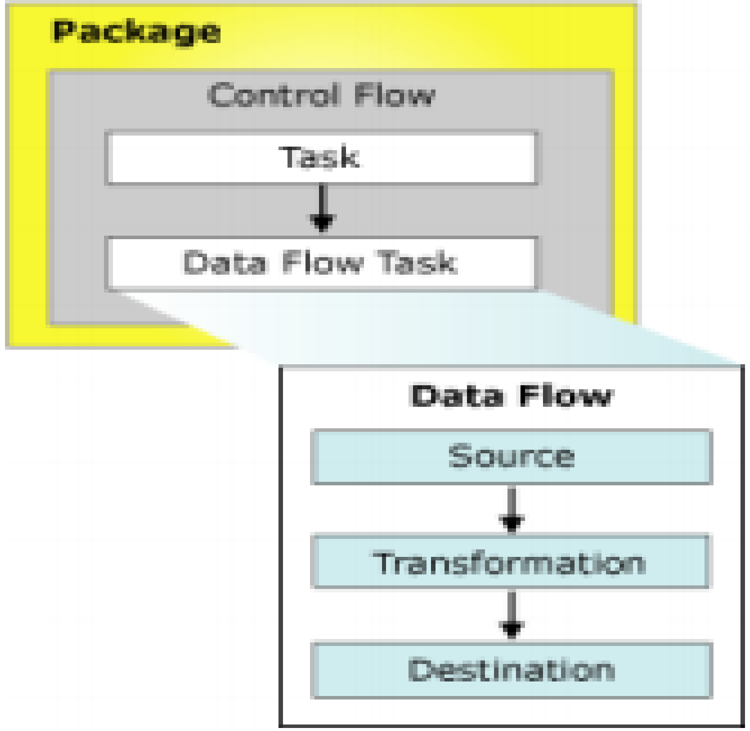
- Functionality available in several ways

  Through GUI - basic functionality

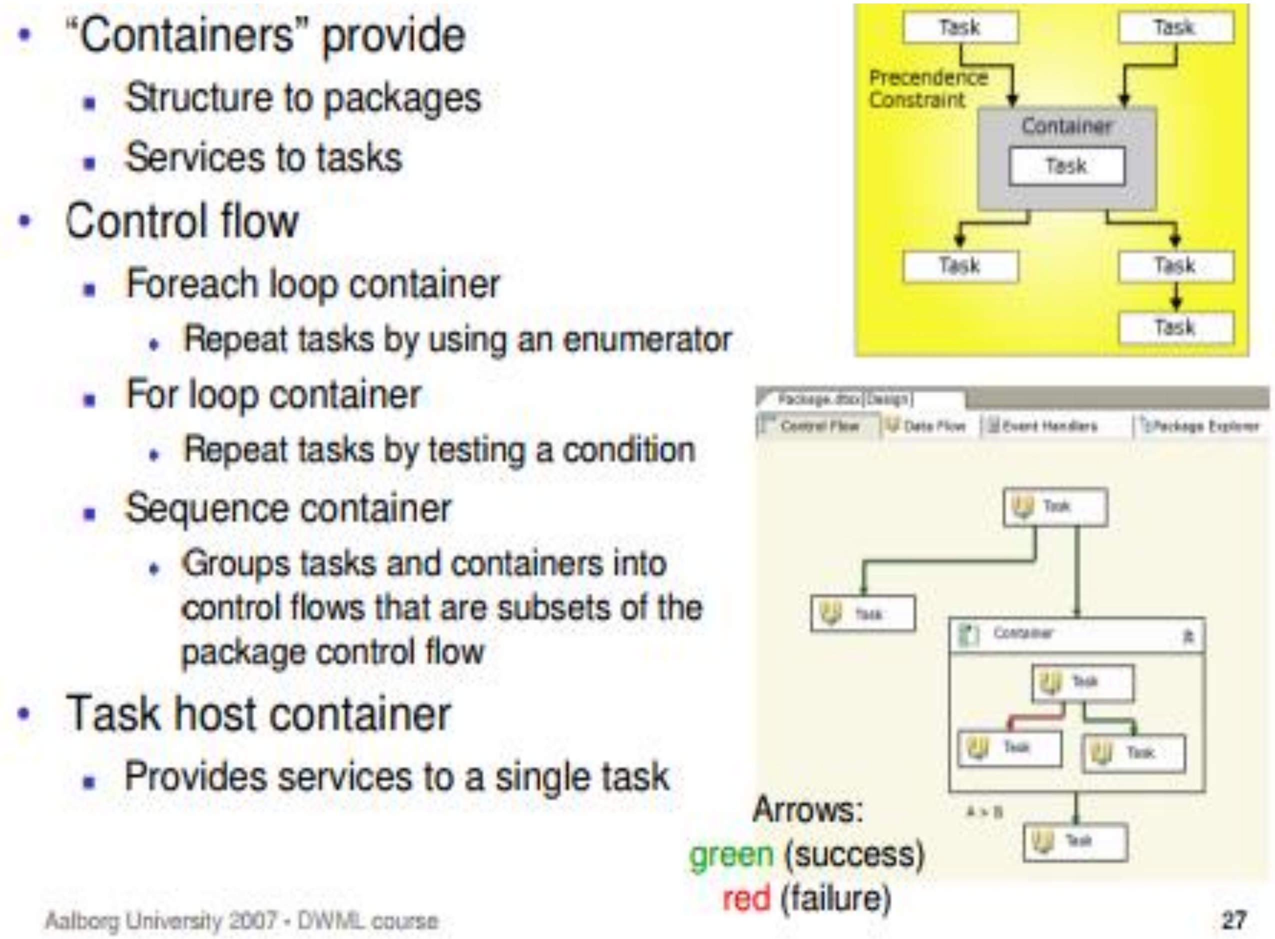
  Programming - advanced functionality

Packages

- The central concept in IS
- Package for:
  - Sources, Connections
  - Control flow
  - Tasks, Workflows
  - Transformations
  - Destinations
  - ……



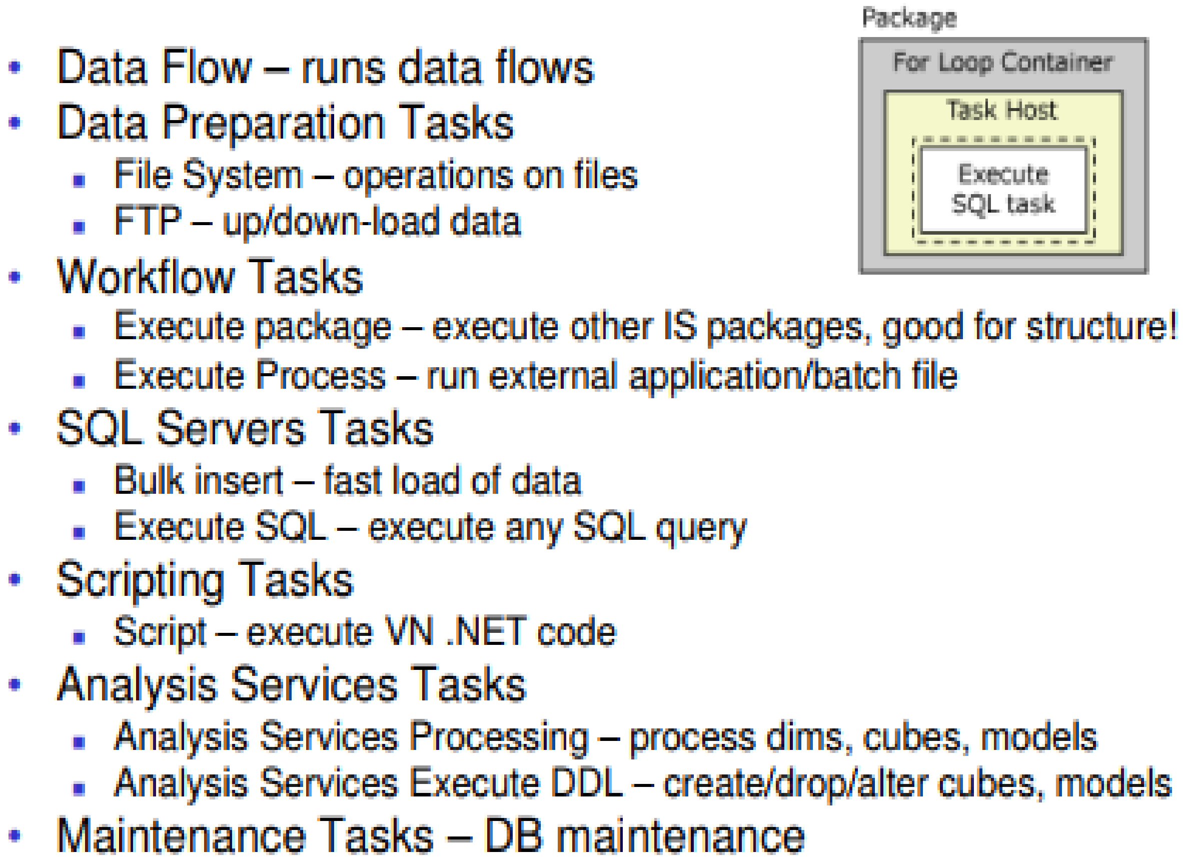
Package Control Flow



- “Containers” provide
  - Structure to packages
  - Services to tasks
- Control flow
  - Foreach loop container
    - Repeat tasks by using an enumerator
  - For loop container
    - Repeat tasks by testing a condition
  - Sequence container
    - Groups tasks and containers into control flows that are subsets of the package control flow
- Task host container
  - Provides services to a single task

Arrows:
green (success)
red (failure)

Tasks



Package
For Loop Container
Task Host
Execute SQL task

- Data Flow – runs data flows
- Data Preparation Tasks
  - File System – operations on files
  - FTP – up/down-load data
- Workflow Tasks
  - Execute package – execute other IS packages, good for structure!
  - Execute Process – run external application/batch file
- SQL Servers Tasks
  - Bulk insert – fast load of data
  - Execute SQL – execute any SQL query
- Scripting Tasks
  - Script – execute VN .NET code
- Analysis Services Tasks
  - Analysis Services Processing – process dims, cubes, models
  - Analysis Services Execute DDL – create/drop/alter cubes, models
- Maintenance Tasks – DB maintenance

Hints on ETL design

- Don’t implement all transformations in one step! Build first step and check that result is as expected

Add second step and execute both, check result

Add third step…

- Test SQL before putting into IS
- Do one thing at the time

  Copy source data one-one to DSA

  Compute deltas - Only if doing incremental load

  Handle versions and DW keys

  Versions only if handling slowly changing dimensions

  Implement complex transformations

  Load dimensions

  Load facts

ensure the control totals match
u have ctrl total before
u will have ctrl total after
compare these two and make sure they match