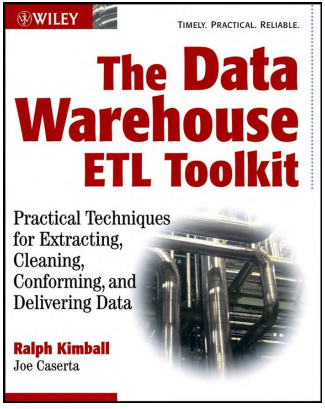
Database Administration

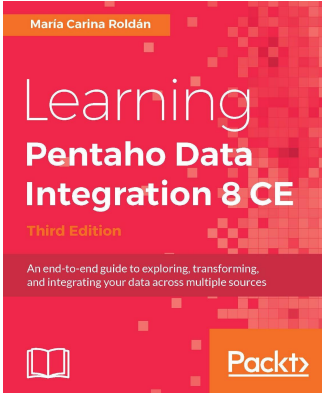
Lecture 05: ETL – Extraction, Tranformation & Load

Kimball, Caserta & Roldán

25 de agosto de 2025

Database Administration: ETL – Extraction, Tranformation & Load.





Content has been extracted from *"The Data Warehouse ETL Toolkit"* by Kimball & Caserta, 2004. Visit kimballgroup.com and *"Learning Pentaho Data Integration 8 CE"* by Roldán, 2018. Visit oreilly.com.

What is a Data Warehouse?

- **Purpose:** Publish organizational data assets to support decision-making.
- **Core traits:** Subject/process-oriented, integrated, time-variant, non-volatile.
- **Users:** Analysts, managers, apps (dashboards, reports, OLAP, data science).
- **Outcomes:** Trusted metrics, faster insight, single version of the truth.

Main Components (Kitchen & Dining Metaphor)

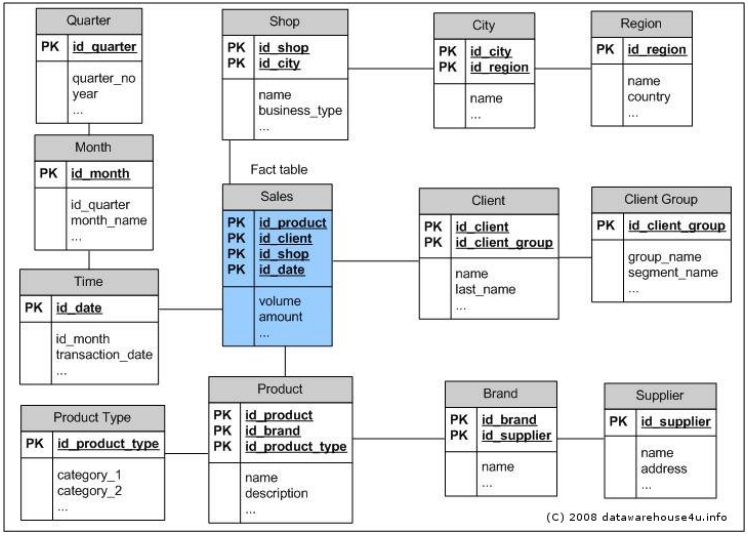
Back Room (Kitchen)

- Staging & processing area (no end-user access)
- Raw sources → standardized, high-quality data
- Governance: lineage, quality checks, security, restart/recovery

Front Room (Dining room)

- Dimensional models / data marts (atomic + aggregates)
- Access by BI tools: SQL, reports, dashboards, OLAP, ML
- Performance tuning, semantic consistency (*conformed* dims/facts)

DW Schema


Quarter
PK id_quarter
quarter_no
year
...
Shop
PK id_shop
PK id_city
name
business_type
...
City
PK id_city
PK id_region
name
...
Region
PK id_region
name
country
...
Month
PK id_month
id_quarter
month_name
...
Fact table
Sales
PK id_product
PK id_client
PK id_shop
PK id_date
volume
amount
...
Client
PK id_client
PK id_client_group
name
last_name
...
Client Group
PK id_client_group
group_name
segment_name
...
Time
PK id_date
id_month
transaction_date
...
Product
PK id_product
PK id_brand
PK id_product_type
name
description
...
Brand
PK id_brand
PK id_supplier
name
...
Supplier
PK id_supplier
name
address
...
Product Type
PK id_product_type
category_1
category_2
...
(C) 2008 datawarehouse4u.info

ETL Responsibilities

- Add value: quality, standard units, surrogate keys, SCDs
- Preserve lineage & auditability; archive staging
- Automate: scheduling, exceptions, recovery/restart
- Meet **latency** needs: batch or streaming (real-time)

Plan

Why Requirements Matter

- ETL design begins with surrounding the requirements.
- Requirements are non-negotiable constraints to adapt to.
- Early architectural decisions drive:
  - Hardware & software
  - Coding practices
  - Personnel & operations
- Clear mission: define back room, staging, operational data stores, presentation area.

Plan

Business Needs

- End users' information requirements.
- Business needs drive the choice of data sources.
- Interviews and investigations often uncover:
  - Hidden complexities or limitations
  - Additional capabilities of data sources
- Continuous dialogue between ETL team, architects, and end users.

Compliance & Security

- Sarbanes–Oxley and other regulations demand:
  - Proof of accuracy, completeness, and lineage.
  - Archived copies and documented algorithms.
- Security:
  - Role-based access control (via directory server).
  - Separate ETL subnets, controlled backups, logs.

Other Requirements

- **Data Profiling:** assess quality, completeness, and usability.
- **Integration:** conforming dimensions & facts.
- **Latency:** batch vs streaming delivery.
- **Archiving & Lineage:** keep staged data + metadata.
- **End User Interfaces:** responsibility of ETL to simplify delivery.
- **Skills & Legacy:** staff expertise and existing licenses impact design.

Plan

ETL Tool vs. Hand Coding

ETL Tool Advantages:

- Faster development, metadata management.
- Built-in scheduling, connectors, lineage tracking.
- Good performance at scale.

Hand-Coded Advantages:

- Unlimited flexibility, OOP, unit testing.
- Full control over metadata.
- Avoid vendor lock-in.

Other Architectural Issues

- Proven technology vs. untested tools.
- Batch vs. Streaming data flows.
- Task dependency: Horizontal vs. Vertical (latency vs consistency).
- Scheduler automation and monitoring.
- Exception handling, quality management, recovery.
- Metadata repositories and process-flow tracking.

Plan

The Back & Front Rooms

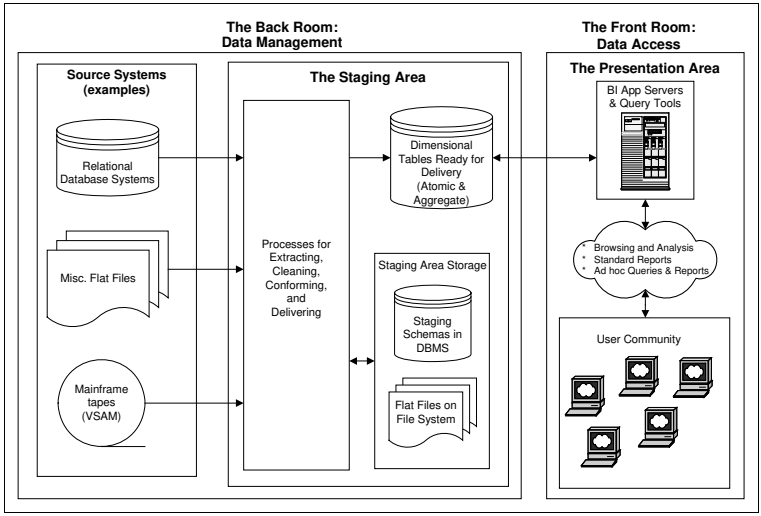


Figure 1.1 The back room and front room of a data warehouse.

The Back Room – Data Management

- Kitchen metaphor: preparation behind the scenes.
- Four staging steps:
  1. Extract
  2. Clean
  3. Conform
  4. Deliver
- Strictly off-limits to end users.

Four staging steps

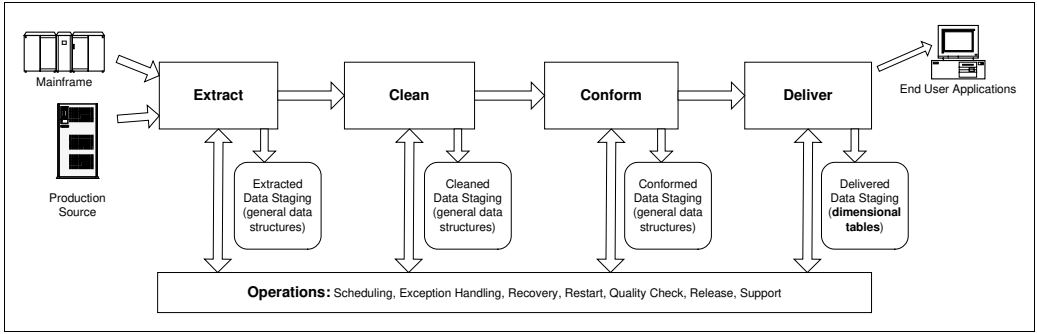


Figure 1.2 The Four Staging Steps of a Data Warehouse.

The Front Room – Data Access

- Presentation layer for queries, dashboards, OLAP cubes.
- Data marts = measurement-intensive subject areas.
- Must be atomic (detail) + aggregated (pyramidal).
- Can be centralized or decentralized, but always conformed.

Plan

Mission of the Data Warehouse & ETL Team

Data Warehouse:

- Publish data assets to support decision-making.
- Deliver reliable, usable, and timely information.

ETL Team:

- Build the back room.
- Add value by cleaning and conforming data.
- Protect and document lineage.
- Deliver data for querying, reporting, dashboards.

Closing Thoughts

- Requirements are the foundation of ETL design.
- Early architecture decisions shape the entire system.
- Back room = preparation; Front room = access.
- The ETL team's mission is strategic to DW success.

Plan

Pentaho Data Integration in a nutshell

- **PDI (a.k.a. Kettle)** = engine + tools for **Extract, Transform, Load (ETL)**.
- Part of the **Pentaho BI Suite**: analysis (Mondrian), reporting, data mining (Weka), dashboards (CTools), *etc.*
- Tight platform services: *authz/authn, scheduling, web services, scalability, failover.*
- Community roots → adopted by Pentaho; rapid evolution with frequent releases.

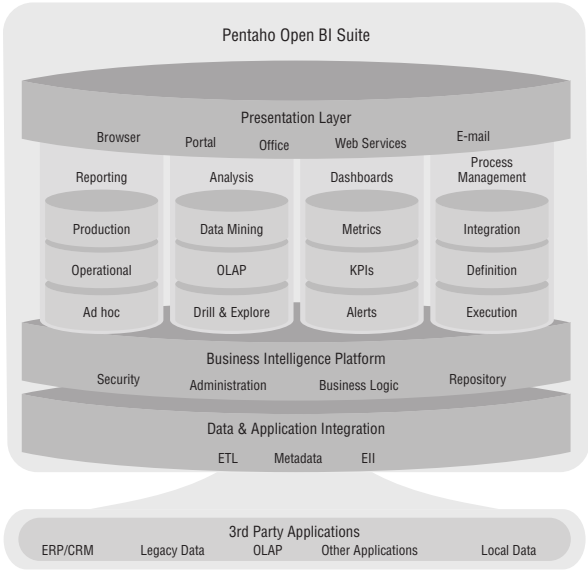
Where PDI fits in the BI stack

- **Data Integration (PDI)** feeds:
  - OLAP (Mondrian)
  - Reports (*Pentaho Reporting*)
  - Data mining (Weka, R/CPython steps)
  - Dashboards (CDE/CCC/CDA)
- Can run standalone or embedded in the platform.

Key benefits

- Open source ecosystem
- Broad connectivity
- Visual design (Spoon)
- Scales from laptop to cluster

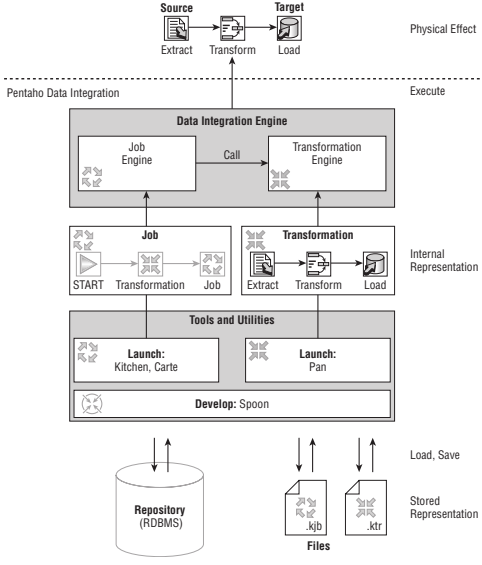
Pentaho Architecture



Characteristics

- Data integration: Combining data from different sources to provide a unified view.
- Pentaho Data Integration (PDI) offers tools for ETL (Extract, Transform, Load).
- Core PDI components: Transformations, Jobs, and the Data Integration Engine.

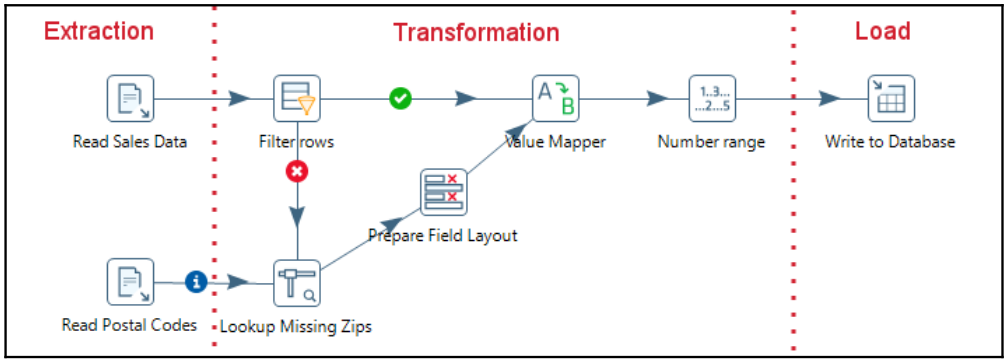
Data Integration Arquitecture



Plan

Typical uses (beyond "just ETL")

- **Load data warehouses/marts** ($\mathrm{E} \to \mathrm{T} \to \mathrm{L}$ pipelines, SCD handling).



Data Integration Activities

- Extraction: Retrieving data from various sources.
- Change Data Capture (CDC): Identifying changes in source data.
- Data Staging: Intermediate storage for transformation.
- Data Validation and Cleansing: Ensuring data quality.
- Key Management and Aggregation.
- Dimension and Fact Table Loading.

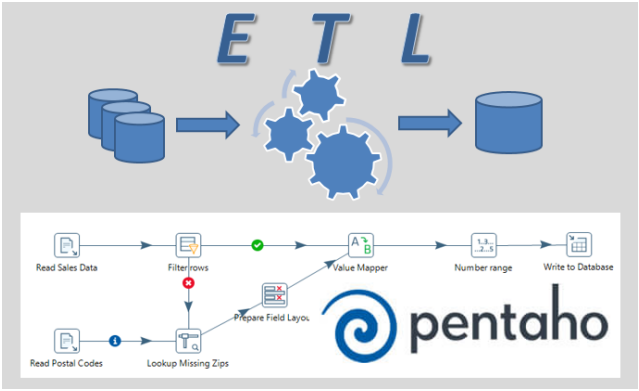
Typical uses (beyond "just ETL")

- **Integrate systems** (ERP + CRM, mergers, multi-source unification).
- **Data cleansing** (validation, standardization, deduplication, defaults).
- **Migrations** (schemas/files $\leftrightarrow$ RDBMS/spreadsheets).
- **Exports & interoperability** (regulatory, inter-dept sharing).
- **Orchestration/automation** (emails, scheduled jobs, preprocess feeds for reports/dashboards).

Plan

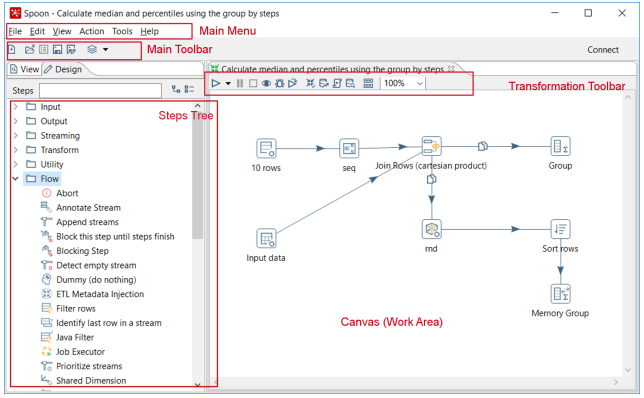
Pentaho Data Integration Components

- **Kettle**: Data integration engine.
- **Spoon**: Graphical IDE for designing transformations and jobs.
- **Kitchen**: Command-line tool for executing jobs.
- **Pan**: Command-line tool for running transformations.
- **Carte**: Remote execution engine.



Getting Started with Spoon

- Launch Spoon and create a new transformation.
- Add steps to extract, transform, and load data.
- Connect steps using “hops”.
- Preview and execute transformations.



Spoon interface at a glance

- **Main Menu/Toolbar**
- **Design** (Steps tree)
- **View** (structure/logs/metrics)
- **Canvas** (your pipeline graph)
- **Transformation Toolbar** (preview, run, debug)

Mental model

Transformation = *steps* + *hops* (dataflow oriented).
Artifacts are metadata (XML) interpreted by the Kettle engine.

Extending PDI with the Marketplace

- **Tools** → **Marketplace**: browse/install plugins by *Type* and *Maturity*.
- Two lanes: *Community* vs *Customer*; stages 1–4 (*lab* → *production-ready*).
- Some plugins are EE-only; descriptions indicate availability.

Plan

Hello, World! (hands-on in 90 seconds)

1. **File** → **New** → **Transformation**.
2. Drag **Input** → **Data Grid** to canvas; define a `name` column and sample rows.
3. Drag **Scripting** → **User Defined Java Expression**; hop from Data Grid to UDJE.
4. In UDJE, create field `hello_message = ``Hello, '' + name + ``!''`.
5. **Preview** (magnifier icon) on UDJE to sample output; then **Run**.

Save it

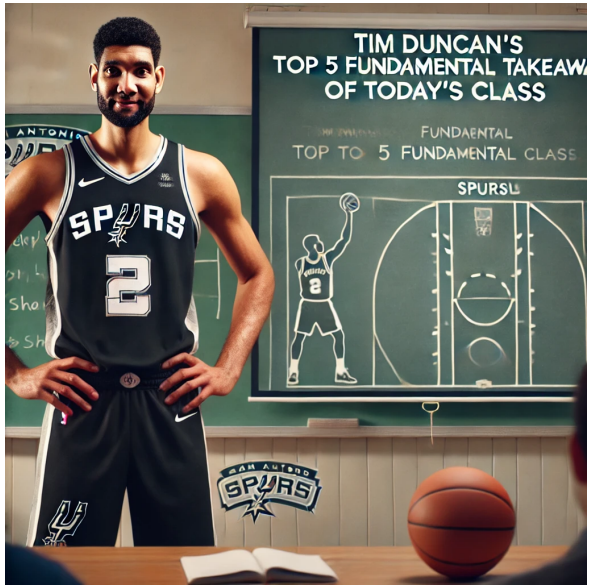
Edit → **Settings** → name, description, extended description → **File** → **Save**.

Summary

- PDI is a versatile, pluggable **data integration** engine with a visual designer.
- PDI provides tools for efficient ETL processes.
- Transformations process data at the record level.
- Jobs orchestrate multiple tasks.
- Spoon offers a user-friendly interface for development.
- You can **prototype quickly** (preview/run) and scale up as needed.
- Marketplace accelerates adoption via **plugins**—mind the maturity stage.

End of Lecture 5.

TDT5FTOTC


TIM DUNCAN'S
TOP 5 FUNDAMENTAL TAKEAW
OF TODAY'S CLASS
FUNDAENTAL
TOP TO 5 FUNDAMENTAL CLASS
SPURS
SPURS
SPURS

TDT5FTOTC

5 PDI, formerly Kettle, is a **powerful engine and suite of tools** for **(ETL) processes**, vital for integrating scattered information and a core part of the Pentaho BI Suite.

TDT5FTOTC

5 PDI, formerly Kettle, is a **powerful engine and suite of tools** for **(ETL) processes**, vital for integrating scattered information and a core part of the Pentaho BI Suite.

4 A **fundamental architectural decision** in ETL system design is choosing between a **vendor ETL tool** or **hand-coding**, significantly impacting development, metadata management, automation, flexibility, and performance.

TDT5FTOTC

5 PDI, formerly Kettle, is a **powerful engine and suite of tools** for **(ETL) processes**, vital for integrating scattered information and a core part of the Pentaho BI Suite.

4 A **fundamental architectural decision** in ETL system design is choosing between a **vendor ETL tool** or **hand-coding**, significantly impacting development, metadata management, automation, flexibility, and performance.

3 The **ETL process**—comprising **extracting, cleaning, conforming, and delivering** data into a dimensional format—is the core of data warehousing, consuming at least **70 % of project time, effort, and cost**.

TDT5FTOTC

5 PDI, formerly Kettle, is a **powerful engine and suite of tools** for **(ETL) processes**, vital for integrating scattered information and a core part of the Pentaho BI Suite.

4 A **fundamental architectural decision** in ETL system design is choosing between a **vendor ETL tool** or **hand-coding**, significantly impacting development, metadata management, automation, flexibility, and performance.

3 The **ETL process**—comprising **extracting, cleaning, conforming, and delivering** data into a dimensional format—is the core of data warehousing, consuming at least **70 % of project time, effort, and cost**.

2 A data warehouse employs a **two-component architecture**: a **'back room'** dedicated to data management and preparation, and a **'front room'** for user data access and analysis.

TDT5FTOTC

5 PDI, formerly Kettle, is a **powerful engine and suite of tools** for **(ETL) processes**, vital for integrating scattered information and a core part of the Pentaho BI Suite.

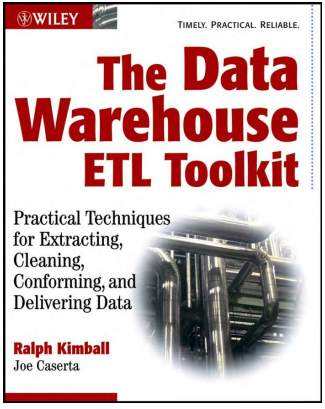
4 A **fundamental architectural decision** in ETL system design is choosing between a **vendor ETL tool** or **hand-coding**, significantly impacting development, metadata management, automation, flexibility, and performance.

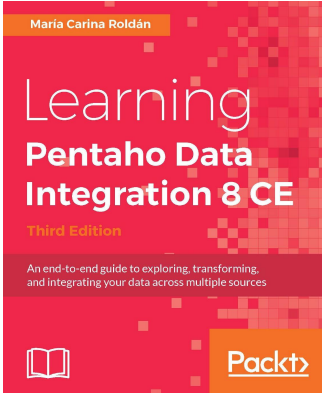
3 The **ETL process**—comprising **extracting, cleaning, conforming, and delivering** data into a dimensional format—is the core of data warehousing, consuming at least **70 % of project time, effort, and cost**.

2 A data warehouse employs a **two-component architecture**: a **'back room'** dedicated to data management and preparation, and a **'front room'** for user data access and analysis.

1 The **data warehouse's central mission** is to **publish organizational data assets for effective decision-making**, with the **ETL team's core task** being to build the 'back room' by cleaning, conforming, documenting lineage, and delivering data dimensionally.

Database Administration: ETL – Extraction, Tranformation & Load.





Content has been extracted from *"The Data Warehouse ETL Toolkit"* by Kimball & Caserta, 2004. Visit kimballgroup.com and *"Learning Pentaho Data Integration 8 CE"* by Roldán, 2018. Visit oreilly.com.