Start-Tech Academy

PDI Installation

Installation instructions

Download Link
https://sourceforge.net/projects/pentaho/

- **Download** the zip file (Current latest version is 9.0.0)
- Unzip it in a folder of you choice

This is the **community edition**. All the things taught in this course can be implemented on enterprise edition also. Enterprise edition has some additional features, you learn more about them in this video: https://www.hitachivantara.com/en-us/video/pentaho-community-edition-vs-enterprise-edition.html

If your office or your client is using older version of PDI, you can find the older versions in the 'files' tab

Opening PDI

Spoon

1. PDI has a desktop graphical designer tool called **Spoon**

2. To launch Spoon in Windows, Run Spoon.bat
   - For Mac and Unix, open Terminal window and type spoon.sh

3. After some time, the spoon window will open and you will see a welcome screen

Demonstration

What and Why

1. What is a transformation and what is a job?
    - Transformations for ETL
    - Jobs for supporting activities like file management and emailing
2. Aim is to show the capability of PDI so that you know exactly what to expect
3. You may or may not do it on your system, this is just a demonstration. You will get plenty of practice exercise throughout the course.

The case

John

Analytics Manager at a furniture sales company
John wants to collect sales data to make a sales dashboard. The file John receives is incomplete. Also, the job is repetitive, i.e. the file is received every week. Therefore, John wants to automate the process of sales data collection.

Actions

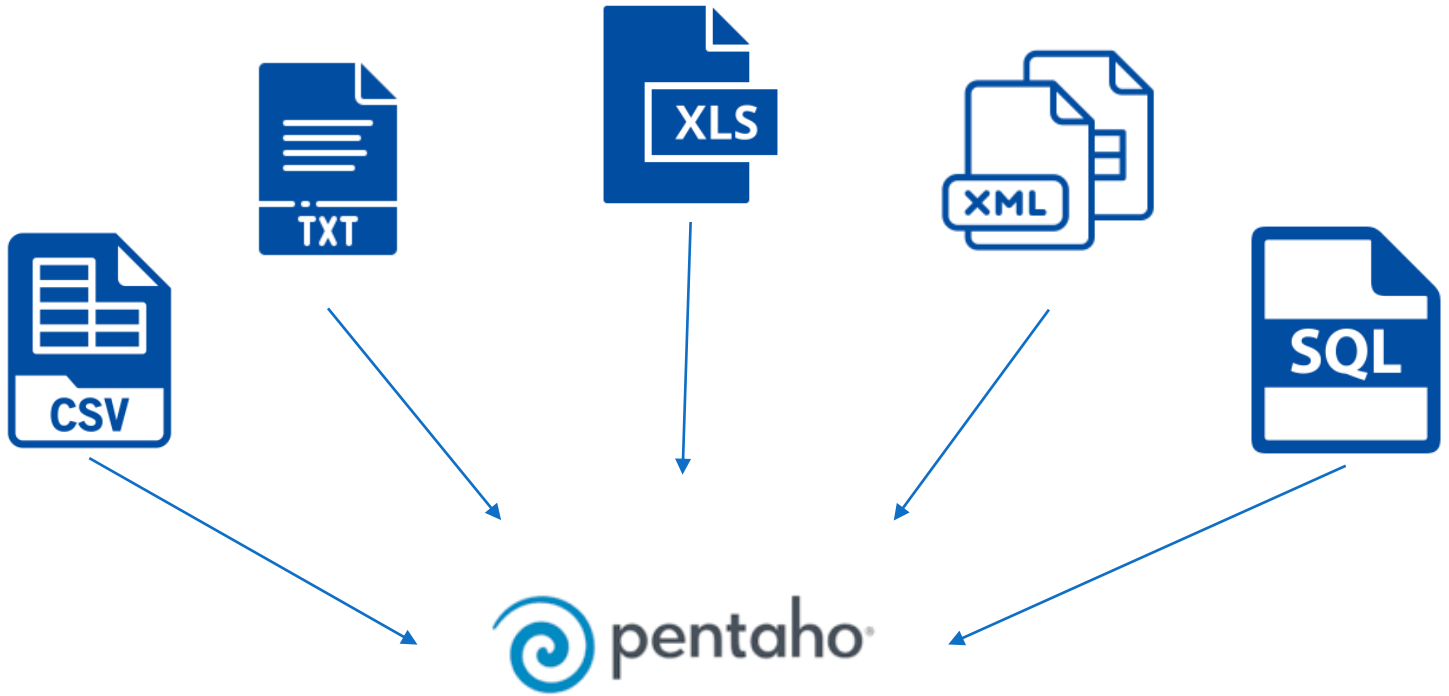
Automation Steps

1. Check whether there is a sales file in sales folder
2. If file is available, import data from file — Extract
3. Identify the rows with missing data and create a separate file for them — Transform
4. Upload the complete file as an Excel file in dashboard folder — Loading
5. Send the incomplete file to the sales manager for rectification

Steps 2,3 and 4 will be done in a PDI Transformation. Steps 1, Transformation and step 5 will be run as part of PDI Job

Extraction

Data Sources

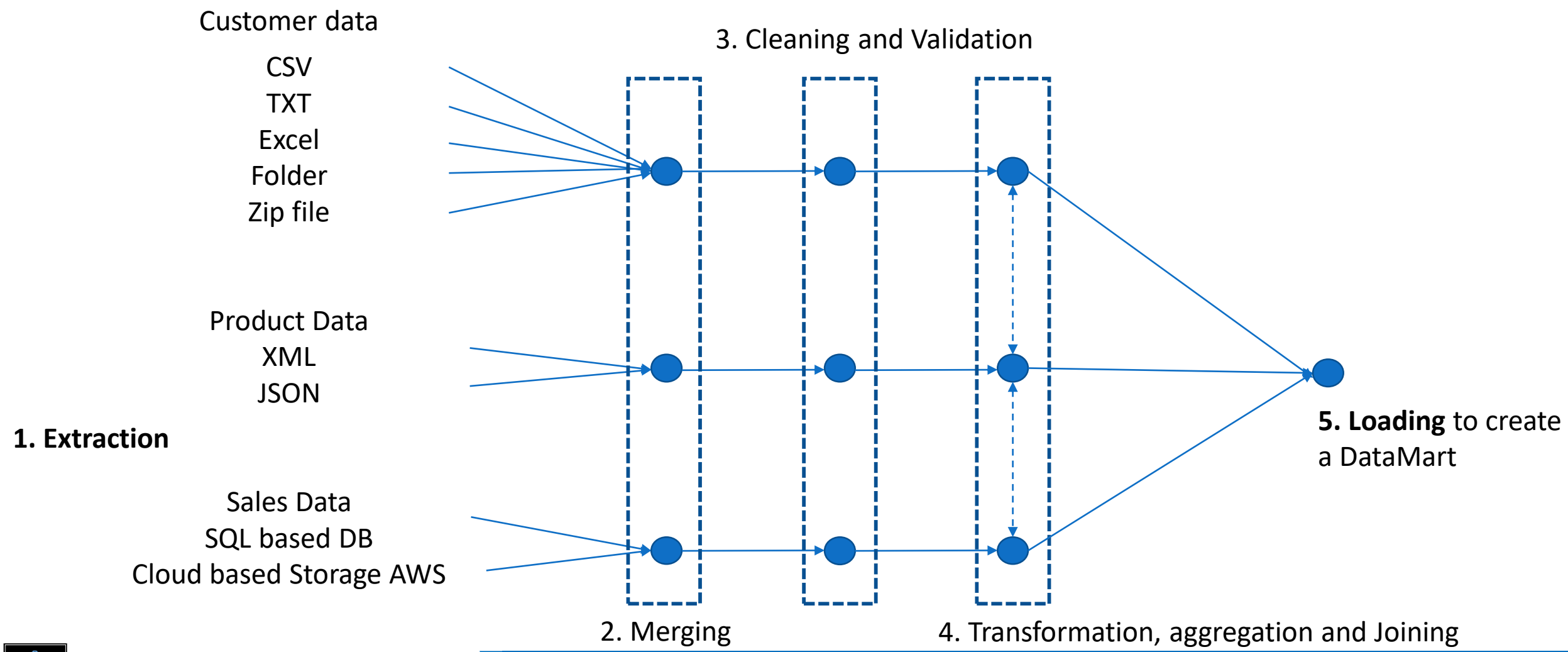


About the Datasets

Three Datasets

1. Sales Data
2. Customer Data
3. Product Data

The ETL Problem



Sales Data

Steps

1. Two CSV files in the resources: **SalesforSQL.csv** and

   SalesforAWSandHadoop.csv

2. Set up PostgreSQL Database and load data into it.

3. Set up AWS S3 Cloud Storage and put SalesforAWSandHadoop.csv there

4. Use PDI to connect with both and fetch data

If you want to learn complete SQL, check out our SQL course:
https://www.udemy.com/course/the-complete-sql-masterclass-for-data-analytics/

Setting up PostgreSQL

CREATE TABLE

```
Create table Sales (
Order_Line int primary key,
Order_ID varchar,
Order_Date date,
Ship_Date date,
Ship_Mode varchar,
Customer_ID varchar,
Product_ID varchar,
Sales numeric,
Quantity int,
Discount numeric,
Profit numeric
);
```

Setting up PostgreSQL

Importing Data from CSV file

Change the location as per your installation directory
COPY sales from 'C:\Program Files\PostgreSQL\12\data\dataset\SalesforSQL.csv' delimiter ',' csv header;

To Check if the data has been correctly imported, run the select command
SELECT * FROM sales;

Setting up PostgreSQL

Summary

1. Install PostgreSQL and PGAdmin on your System
2. Create a new Database called SalesDB
3. Open Query Tool and run the CREATE TABLE Command
4. Copy the ‘SalesforSQL.csv’ file and paste it in the data folder of PostgreSQL
5. Import data from this file using the Copy command

Merging Streams of Data

Considerations

1. Prefer specialized merging steps such as Append Stream or Sorted Merge

A	B	C	D
1	2	3	4
1	2	3	4
1	2	3	4

A	B	C	D
5	6	7	8
5	6	7	8
5	6	7	8

Append

A	B	C	D
1	2	3	4
1	2	3	4
1	2	3	4
5	6	7	8
5	6	7	8
5	6	7	8

A	B	C	D
6	2	3	7
7	2	2	1
3	4	5	2

A	B	C	D
1	6	1	5
2	4	3	6
4	2	5	7

Sorted Merge

A	B	C	D
1	6	1	5
2	4	3	6
3	4	5	2
4	2	5	7
6	2	3	7
7	2	2	1

Merging Streams of Data

Considerations

1. Merged streams may have duplicates
2. Ensure unique occurrence of primary key after merging
3. Sort data before deduplicating
4. The most important thing: Metadata of merging streams must be same

Data Cleansing

What

1. Correcting small mistakes such as typing mistakes or data format related issues. Examples: 5 vs 5.0, 13th July vs 13/07/2020, duplicate entries, etc.
2. After Data extraction we should do data cleansing
3. We cleaned data while extracting also: setting format of dates, sales and profit value in sales data, removing duplicates from product data etc. is part of data cleansing

Data Cleansing

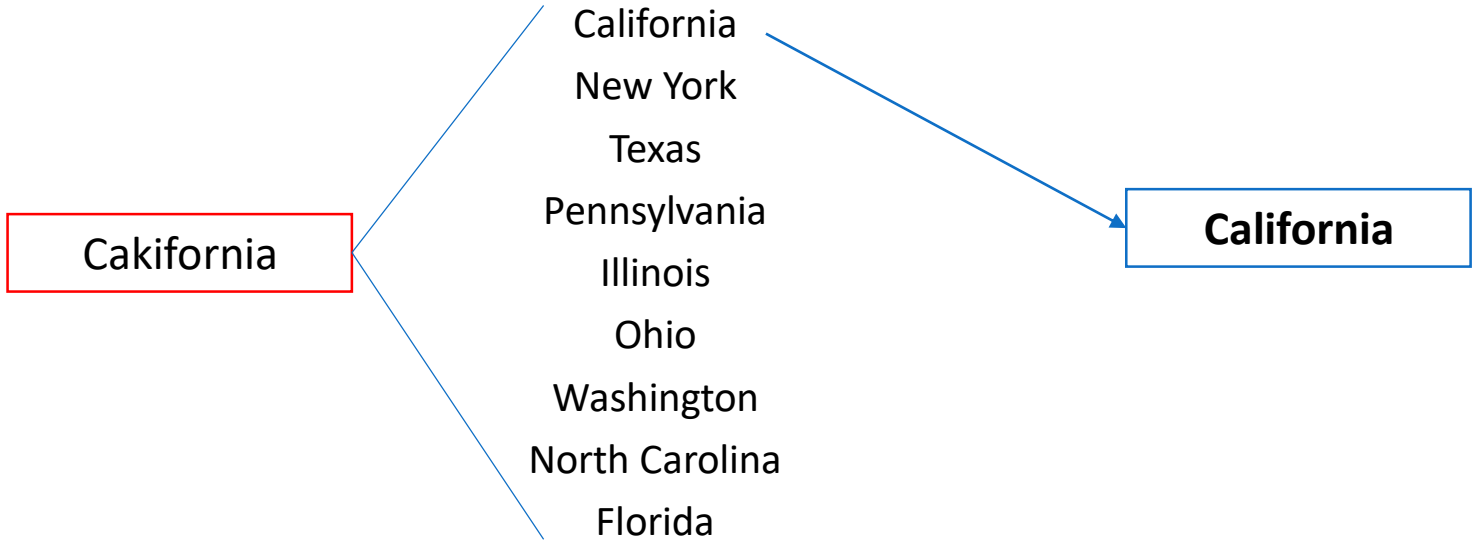
Upcoming examples

1. Country field contains US, USA, United States and United States of America to represent one country
2. Special character # is present in City column
3. Manual entry mistakes in State column such as California is sometimes written as Cakifornia and californis etc.
4. Changing discount value in sales table to percentage value
5. Change date format from mm/dd/yyyy to dd-mm-yyyy for order and ship dates

Data Cleansing

Fuzzy Match

1. Measures closeness with Lookup values
2. Assign value which is most similar



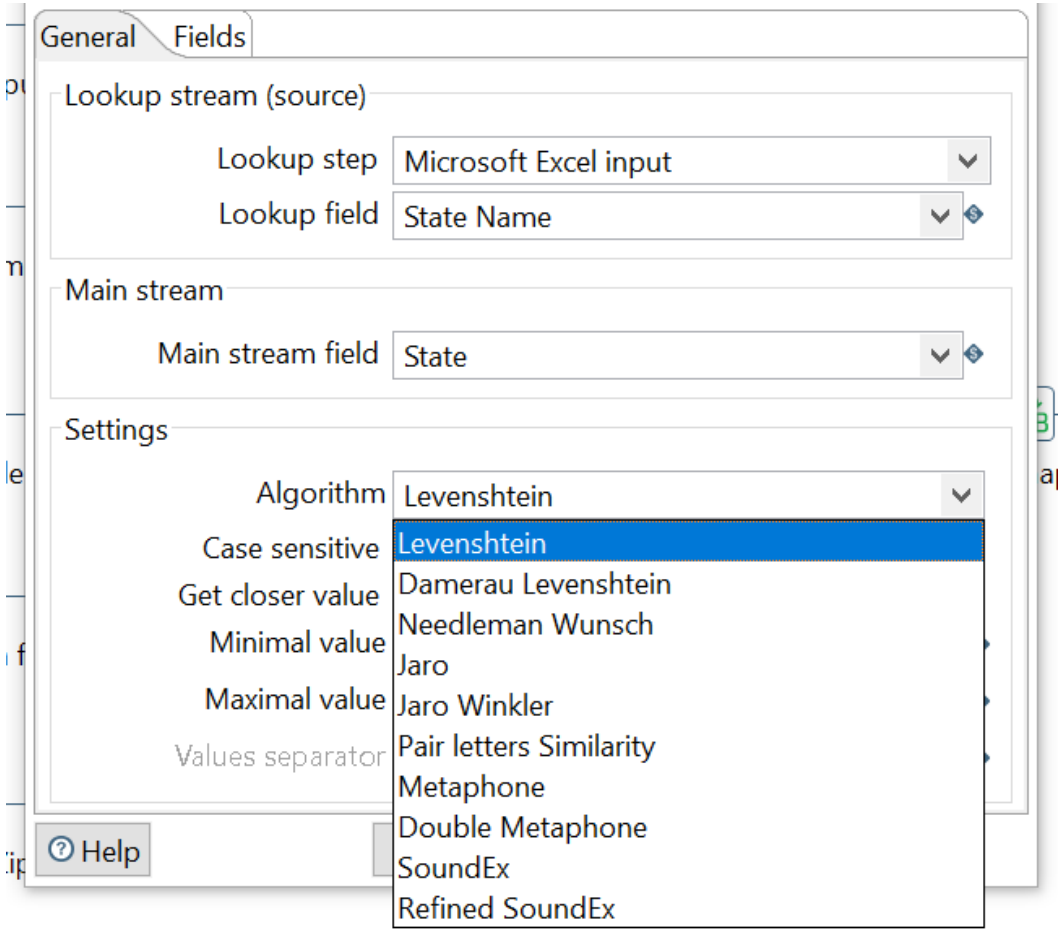
Main value

Lookup values

Closest Value is assigned

Data Cleansing

Fuzzy Match Algorithms



Data Cleansing

Fuzzy Match Algorithms

1. **Levenshtein and Damerau-Levenshtein**
    - Calculates distance by calculating the edit steps
    - Steps – Insert, Delete, Replace (Transpose for Damerau-Levenshtein)
    - Cakifornia -> California only one step, replace ‘k’ with ‘l’
    - Akiforina -> California needs two steps, add ‘c’ and replace ‘k’ with ‘l’
    - Cailfornia -> California needs two replace steps as per Levenshtein or one transpose step (il -> li) as per Damerau-Levenshtein

Data Cleansing

Fuzzy Match Algorithms

2. Needleman-Wunsch

- Score is calculated as penalty
- Cakifornia -> California will have a score of -1
- Different mismatches can have different weights

3. Jaro and Jaro-Winkler

- Calculates similarity index between 0 and 1
- 0 – no similarity and 1 – completely similar
- How similar are CALIFORNIA and FLORIDA?
  - Levenshtein distance – 7 (try yourself to find out how)
  - Jaro similarity score of 0

Data Cleansing

Fuzzy Match Algorithms

4. Pair letters similarity

Ex- find similarity between FLORIDA and FLOTISA

Step 1. Make two character pairs from both strings

FL, LO, OR, RI, ID, DA

FL, LO, OT, TI, IS, SA

Step 2. Calculate score using the formula

Score = Total Pairs Matched/ Total Pairs = 4/ 12 = 0.33

Data Cleansing

Fuzzy Match Algorithms

5. Metaphone, Double Metaphone, Soundex, and RefinedSoundEx

- ‘Phonetic’ Algorithms, try to match the sound of words
- Commonly used for deduplication
- Only applicable on English language

Data Cleansing

Common steps

Scenario	Step
Value must have a particular format	Select values
Values in multiple columns are to be combined into a single column	Concat fields
Assign new value basis the value of a field containing number	Number range
Assign new value basis the value of a field containing a string	Value mapper
Remove duplicates	Unique rows
Remove/ change special characters or part of strings	Replace in string

Data Validation

What

1. **Data Cleansing** is to correct mistakes and format for data to improve its quality.
2. **Data Validation** is to ensure that the Data complies with the business rules
3. **Examples**:
    - Age field should contain integer type values
    - Customer age should be more than 18 years
    - Product ID in the sales table should be available in the Product table as well
    - Credit Card number/ email ID/ Phone number should have a predefined format

Data Validation

Error Handling

1. If some rows do not respect the data validation rules, those are the error rows.
2. We need to properly handle error rows.
3. Error can be handled in these four ways:
   - Discarding the error rows
   - Separating error rows, processing them and remerging them with the main stream
   - Reporting the error rows to the log
   - Writing the error rows in a file or a dedicated table for further revision

Data Validation

Common steps

Scenario	Step
Value must have a given data type such as String or Date	Select values
Value cannot be null	Filter rows
Numbers or dates should fall inside an expected range	Filter rows (>, <, or = functions)
Values must belong to list found in an external source such as a file or a database	Stream Lookup or Data base Lookup
Text should not contain certain terms or substrings	Replace in string step

Transformation

Normalizing

States	Consumer	Corporate	Home Office
California	86	45	26
New York	44	24	19
Texas	42	25	7
Pennsylvania	23	18	9
Illinois	14	11	16
Ohio	24	9	5
Washington	27	9	2
North Carolina	17	9	4

State	Segment	Number of Customers
California	Consumer	86
California	Corporate	45
California	Home Office	26
New York	Consumer	44
New York	Corporate	24
New York	Home Office	19
Texas	Consumer	42
Texas	Corporate	25
Texas	Home Office	7
Pennsylvania	Consumer	23
Pennsylvania	Corporate	18
Pennsylvania	Home Office	9
Illinois	Consumer	14
Illinois	Corporate	11
Illinois	Home Office	16
Ohio	Consumer	24
Ohio	Corporate	9
Ohio	Home Office	5

PDI – SQL DB connection

Preparing SQL DB

1. Create a new table in postgresql by running this command in PGAdmin

create table science_class (Enrollment_no INT, Name VARCHAR, Science_Marks INT);

2. Insert some sample values

insert into science_class values (1,'Popeye',33); insert into science_class values (2,'Olive',54); insert into science_class values (3,'Brutus',98);

PDI – SQL DB connection

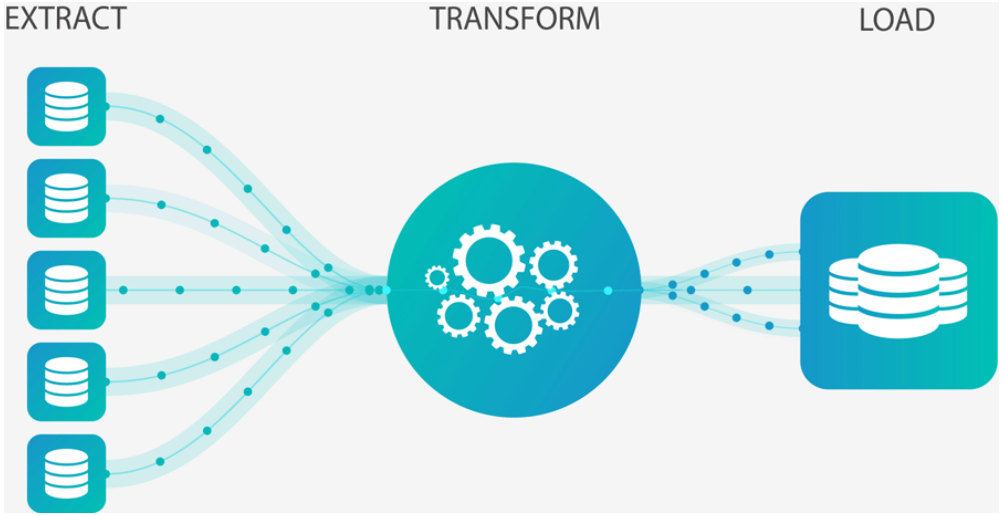
Tasks

1. Retrieve all data from the table ‘Science_Class’ **(Read)**
2. Retrieve the name of students who have scored more than 60 marks **(Read with conditions)**
3. Update the marks of Popeye to 45 **(Update)**
4. Insert a new row with “Wimpy” who has scored 75 marks **(Insert)**
5. Delete the record of Wimpy **(Delete)**

ETL

What

1. **ETL** stands for Extract, Transform and Load
2. **Extract** is reading data from data sources
3. **Transform** is processing the data
4. **Load** is writing the data to the destination



Data Warehouse

What

1. A **data warehouse** collates data from a wide range of sources within an organization.
2. **NOT** operational database.
3. **NOT** updated frequently
4. For the purpose of **analytics**

Data Warehouse

Differences with OLTP

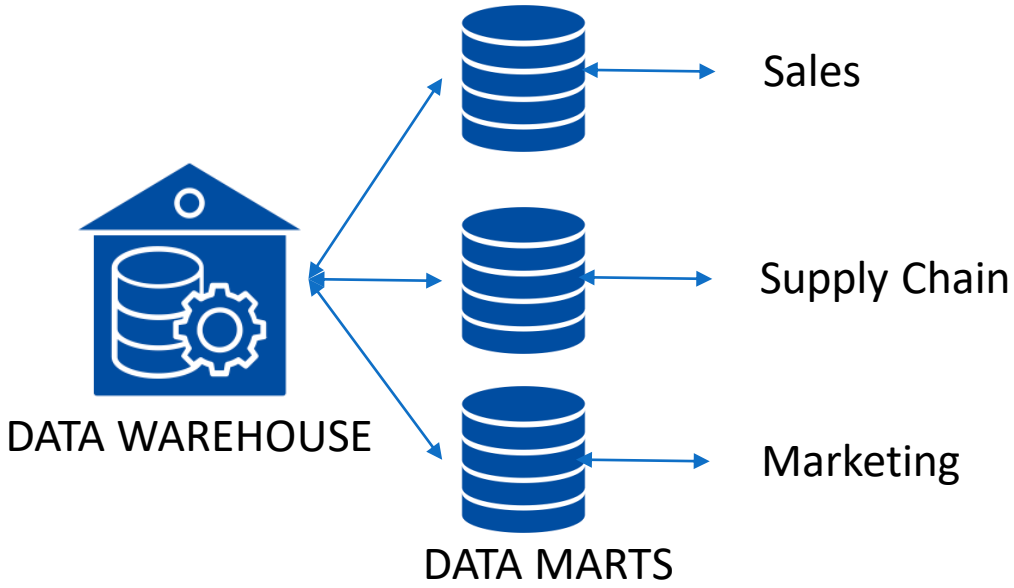
OLTP stands for OnLine Transaction Processing. Used for adding/ updating one/ few rows of data at a time

CHARACTERISTIC	OLTP	DATA WAREHOUSE
System scope/view	Single business process	Multiple business subjects
Data sources	One	Many
Data model	Static	Dynamic
Dominant query type	Insert/update	Read
Data volume per transaction	Small	Big
Data volume	Small/medium	Large
Data currency	Current timestamp	Seconds to days old
Bulk load/insert/update	No	Yes
Full history available	No	Yes
Response times	< 1 second	< 10 seconds
System availability	24/7	8/5
Typical user	Front office	Staff
Number of users	Large	Small/medium

Data Warehouse

Differences with DataMart

Data mart contain repositories of summarized data collected for analysis on a specific unit within an organization, for example, the sales, finance, operations, marketing department.

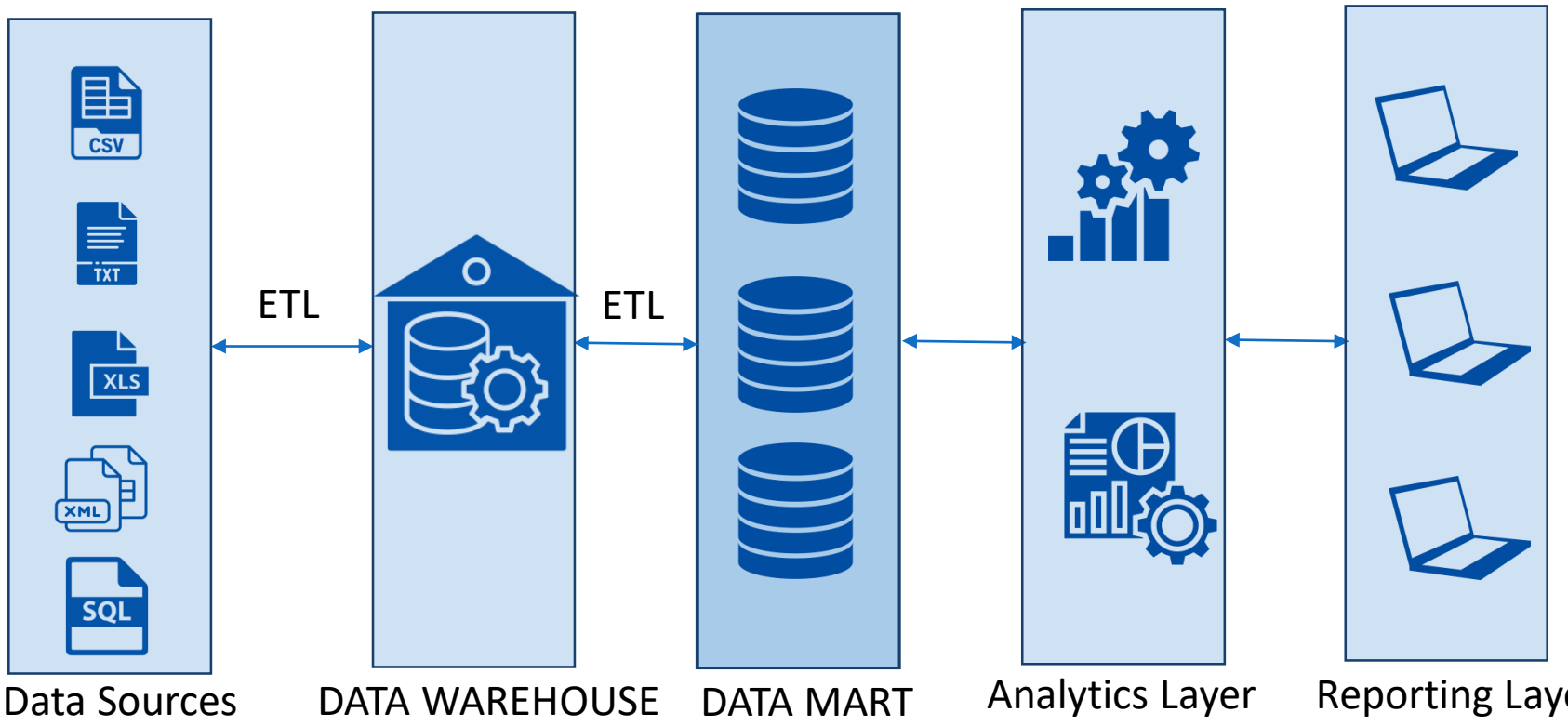


Data Warehouse

Inmon vs Kimball

Inmon model: Organizations create DW, Data marts are created from DW,

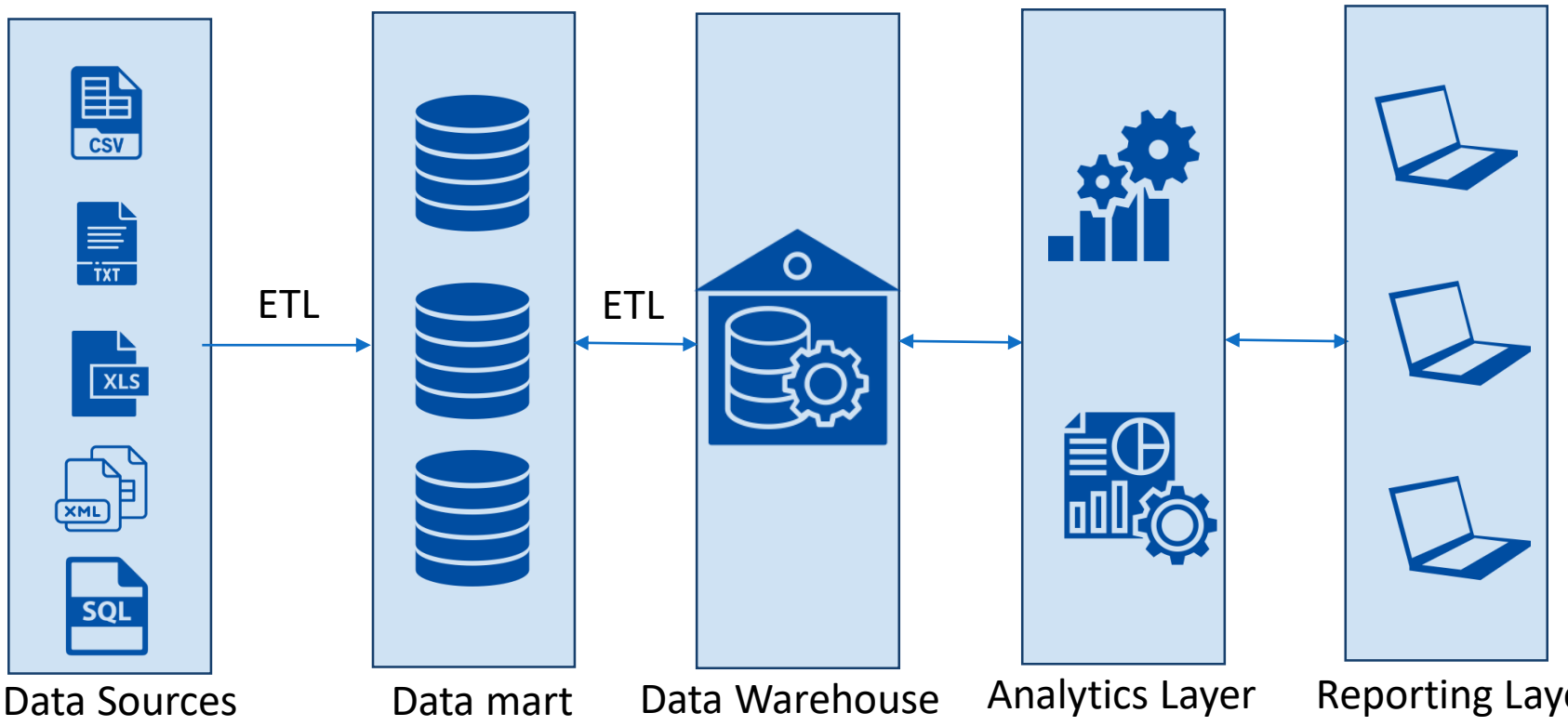
Analytics is applied on Data marts



Data Warehouse

Inmon vs Kimball

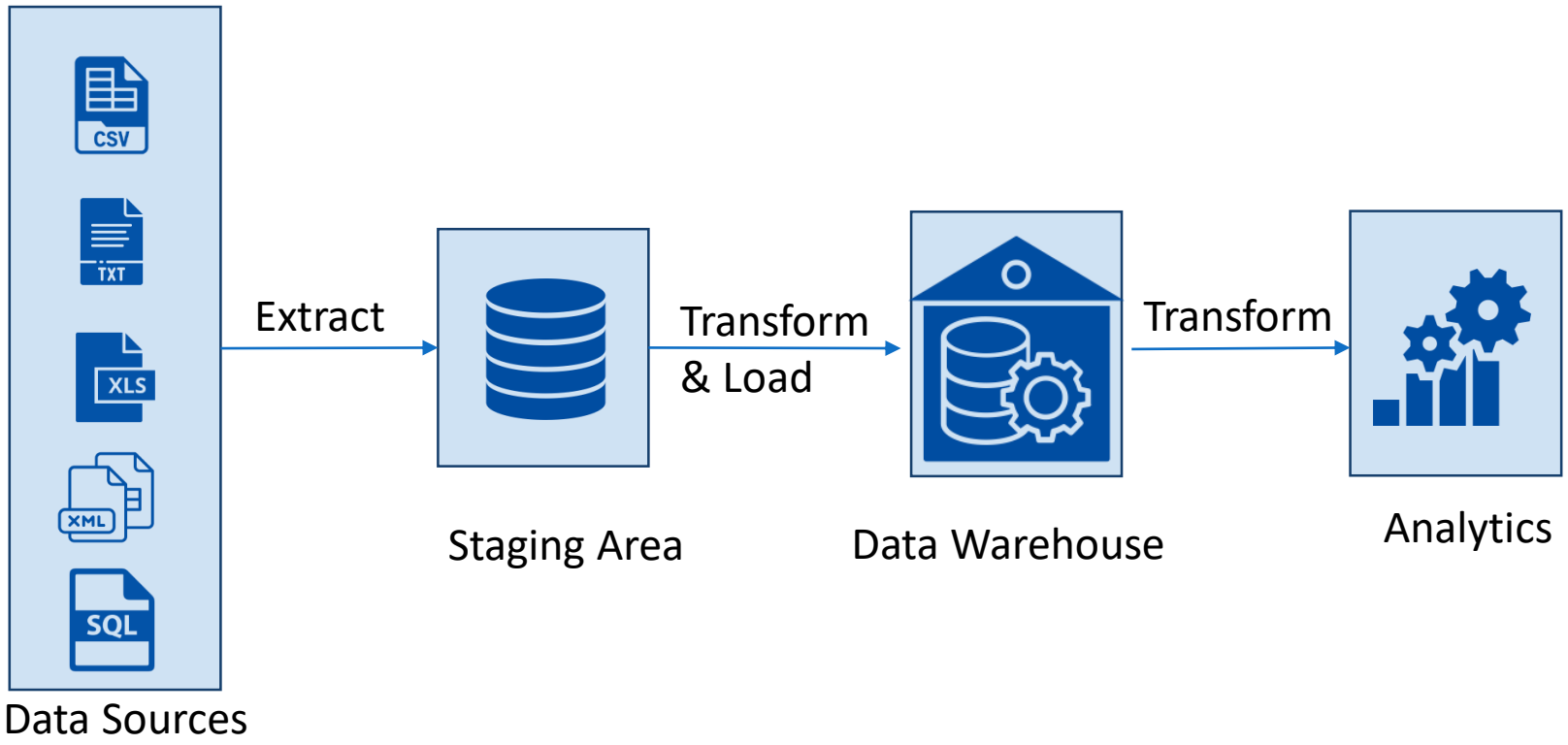
Kimball model: Organizations create Data marts, DW are created from data marts, Analytics is applied on DW



Data Warehouse

ETL vs ELT

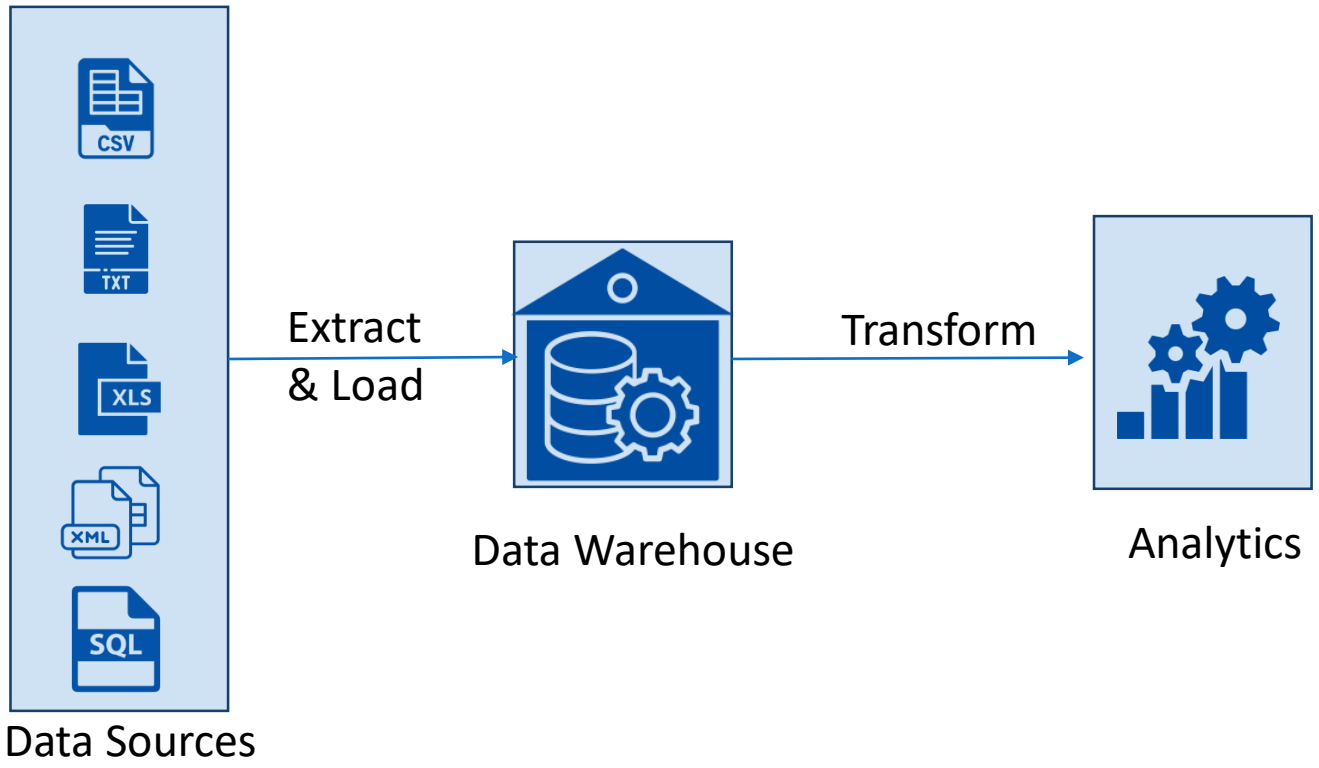
ETL – Extract, Transform, Load



Data Warehouse

ETL vs ELT

ELT – Extract, Load, Transform



Data Loading

Facts and Dimensions

- A **fact table** stores numerical measurements of the business as a quantity of products sold, discounts, taxes, number of invoices, and anything that can be measured.
- These measurements are referred to as **facts**.

- **Dimension tables** contain the textual descriptors of the business.
- Typical dimensions are product, time, customers, and regions.

In our Example,

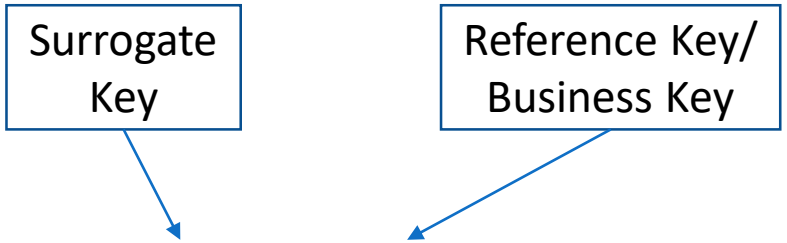
- Sales table is a fact table
- Customer and Product table are Dimension tables

Data Loading

Dimensions Technical Details

- A dimension table must have a technical key also known as a **surrogate key**. Surrogate keys are always integers.
- Dimensions should have a special record for the unavailable data.
- The **business key/ reference key** is also stored to match the data in the dimension table with the data in the source database.

Dimensions

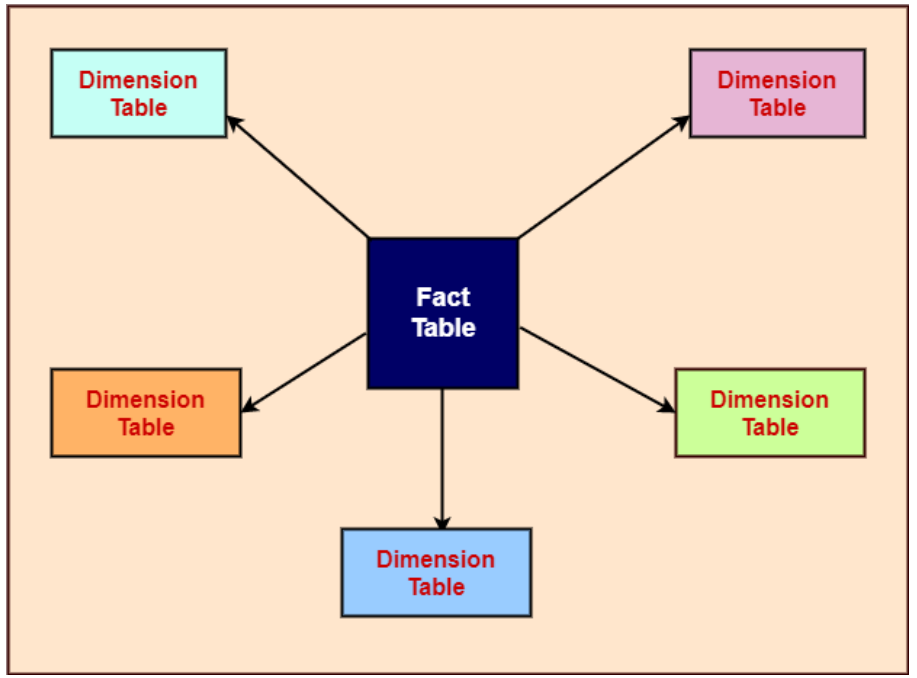


ID	Product ID	Category	Sub-Category	Product Name
1	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase
2	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs Rounded Back
3	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters by Universal
4	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table
5	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold N Roll Cart System

Data Warehouse

Star Schema

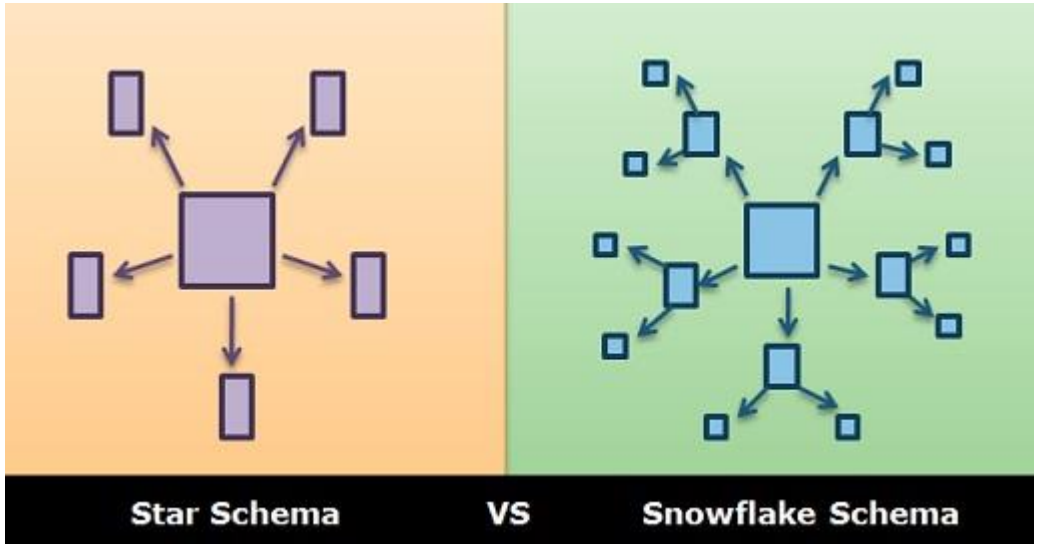
A fact table is at the centre which is connected with several dimension tables



Data Warehouse

Snowflake Schema

- Dimension tables are further connected with sub-dimension tables



Dimensions

Slowly changing dimensions- SCD

- Dimensions such as Region, Product, Customer
- Changes may occur from time to time
- Type 1 SCD: Old value is not saved and the new value is overwritten
- Type 2 SCD: The whole history of changes in dimension data is stored

Loading Data

Product table

```
create table product (
        surr_id int primary key,
        product_id varchar default 'N/A' NOT NULL,
        category varchar default 'N/A' NOT NULL,
        sub_category varchar default 'N/A' NOT NULL,
        product_name varchar default 'N/A' NOT NULL,
        start_date date,
        end_date date,
        version int default 1 NOT NULL,
        current varchar default 'Y' NOT NULL,
        lastupdate date
);
```

Loading Data

Customer table

```
create table customer (
        surr_id int primary key,
        customer_id varchar default 'N/A' NOT NULL,
        customer_name varchar default 'N/A' NOT NULL,
        segment varchar default 'N/A' NOT NULL,
        age int default '0' NOT NULL,
        city varchar default 'N/A' NOT NULL,
        state_name varchar default 'N/A' NOT NULL,
        country varchar default 'N/A' NOT NULL,
        postal_code varchar default 'N/A' NOT NULL,
        region varchar default 'N/A' NOT NULL
);
```

Loading Data

Final sales table

```
create table finalsales (
        order_line int primary key,
        order_id varchar default 'N/A' NOT NULL,
        order_date date default '1900-01-01' NOT NULL,
        ship_date date default '1900-01-01' NOT NULL,
        ship_mode varchar default 'N/A' NOT NULL,
        s_cust_id int default '0' NOT NULL,
        s_prod_id int default '0' NOT NULL,
        sales numeric default '0' NOT NULL,
        quantity int default '0' NOT NULL,
        discount numeric default '0' NOT NULL,
        profit numeric default '0' NOT NULL
);
```

PDI Job vs Transformation

Differences

1. Transformations are about moving and transforming rows from source to target. Jobs about high level flow control: executing transformations, sending mails on failure, transferring files via FTP, ...

2. Transformation execute in parallel, but the steps in a job execute in order.

Regex

Wildcards

Wildcard	Explanation
|	Denotes alternation (either of two alternatives).
*	Denotes repetition of the previous item zero or more times
+	Denotes repetition of the previous item one or more times.
?	Denotes repetition of the previous item zero or one time.
{m}	denotes repetition of the previous item exactly m times.
{m,}	denotes repetition of the previous item m or more times.
{m,n}	denotes repetition of the previous item at least m and not more than n times
^,$	^ denotes start of the string, $ denotes start of the string
[chars]	a *bracket expression*, matching any one of the chars
~*	~ means case sensitive and ~* means case insensitive

Regex

Examples

```
SELECT * FROM customer
WHERE customer_name ~* '^a+[a-z\s]+$'

SELECT * FROM customer
WHERE customer_name ~* '^(a|b|c|d)+[a-z\s]+$'

SELECT * FROM customer
WHERE customer_name ~* '^(a|b|c|d)[a-z]{3}\s[a-z]{4}$' ;

SELECT * FROM users
WHERE name ~* '[a-z0-9\.\-\_]+@[a-z0-9\-]+\.[a-z]{2,5}';
```