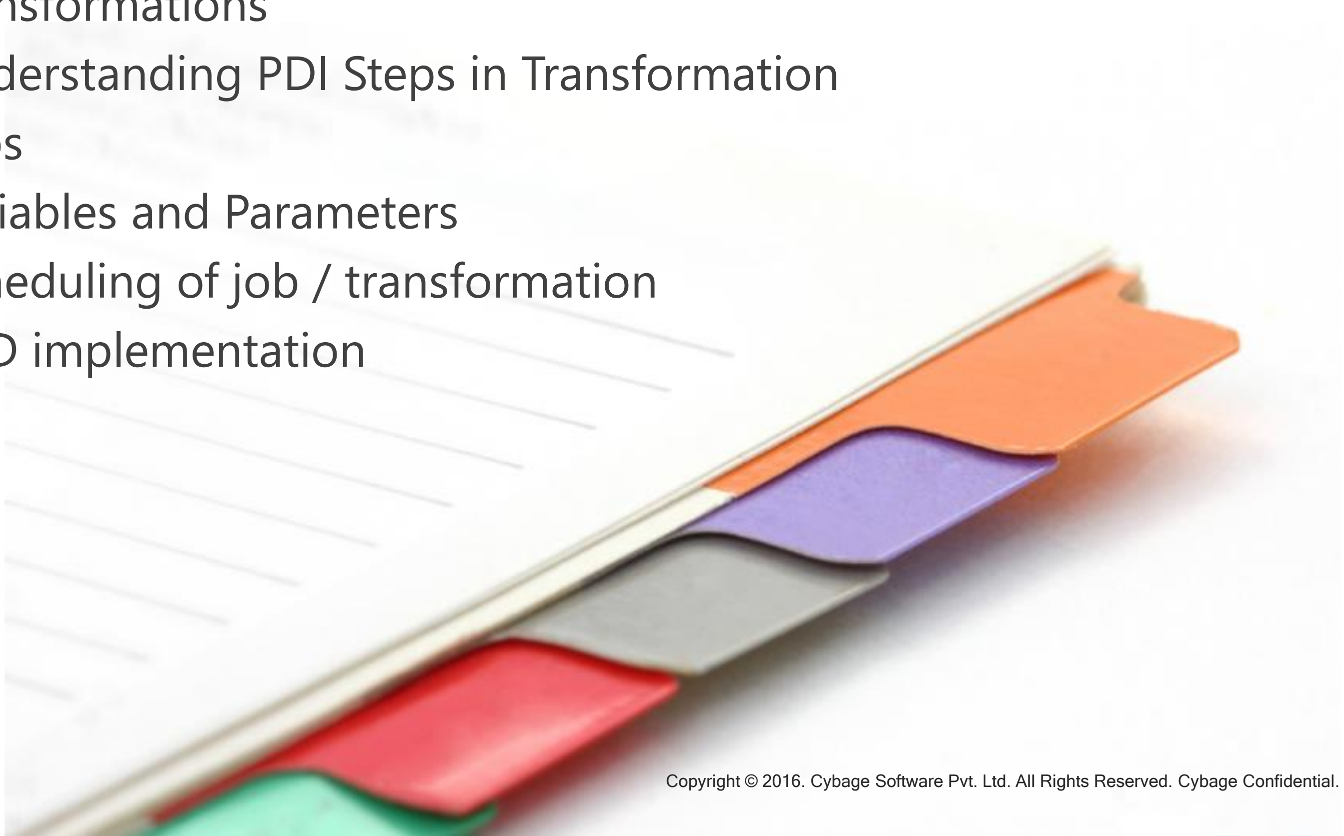
Pentaho Kettle Training

Presented by : Prashant Tikone

Agenda

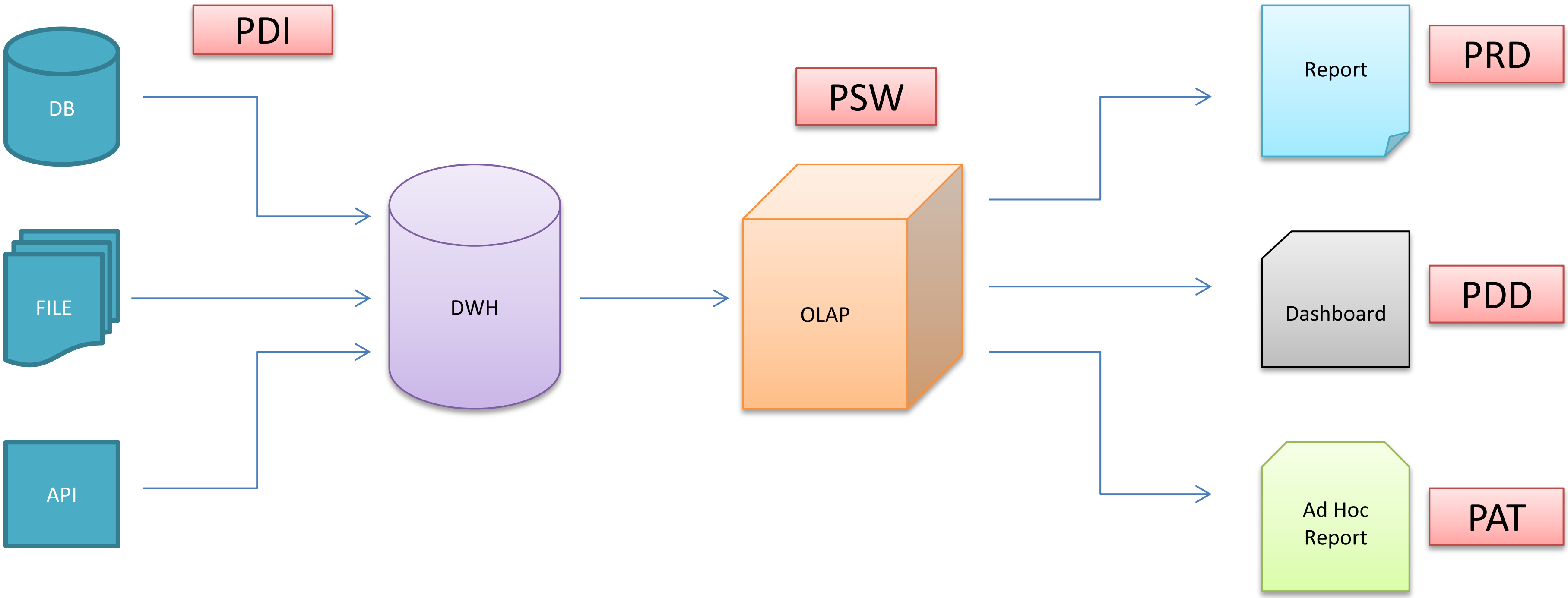
- Introduction to Pentaho BA suite
- Various component of PDI-Kettle
- Dealing with repository
- Transformations
- Understanding PDI Steps in Transformation
- Jobs
- Variables and Parameters
- Scheduling of job / transformation
- SCD implementation



Introduction to Pentaho BA suite

- Pentaho Data Integration
- Pentaho Report Designer
- Pentaho Schema Workbench
- Pentaho BA Server
- Pentaho Dashboard Designer
- Pentaho Analyzer Tool

ETL Architecture and Pentaho



Various components of PDI

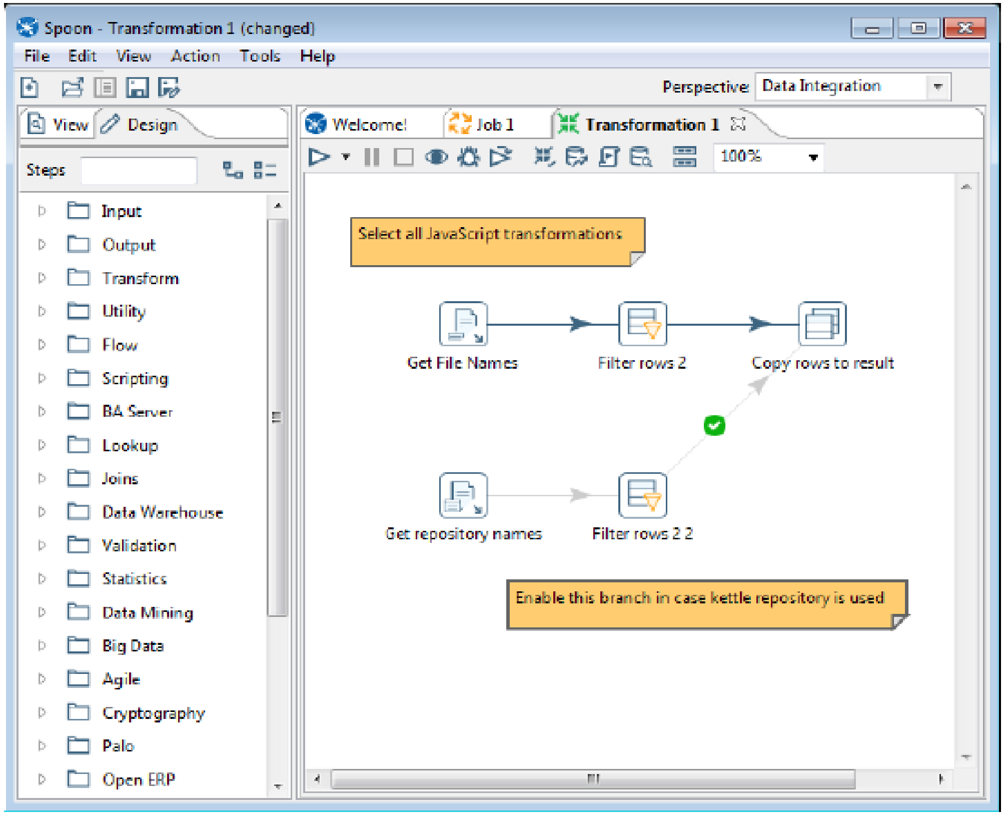
- Data Integration Server
- Spoon
- Pan
- Kitchen
- Carte

Data Integration Server

- Centrally Store for transformations and jobs
- Pentaho Repository
- Processing engine
- Security and authentication,
- Scheduling.

Spoon

- User Interface to design the Pentaho Jobs
- Drag and Drop interface
- Uses library of more than 300 pre-built transformations
- Build Workflows using a series of data integration processing entries.



Select all JavaScript transformations

Enable this branch in case kettle repository is used

Pan, Kitchen & Carte

- Pan - Execute PDI transformations, which represent independent data processing tasks
- Kitchen – Execute PDI jobs, which contain transformations and other job entries as part of a larger business process.
- Carte - Set up cluster of PDI servers. Helps to execute data transformations within a cluster of Carte cluster nodes.

Hardware Requirements

Pentaho Data Integration Server

Hardware—64 bit	Operating System—64 bit
•Processor: Apple Macintosh Pro Quad-Core or Macintosh Mini Quad-Core •Intel EM64T or AMD64 Dual-Core RAM: 8 GB with 4 GB dedicated to Pentaho servers Disk Space: 20 GB free after installation	•Microsoft Windows 2008 Server R2 & 2012 Server •CentOS 6 & 7 •Red Hat Enterprise 6 & 7 •Ubuntu Server 12.04 LTS & 14.04 LTS •OSX 10.10 & 10.11 •Suse Linux SLES 11 (SP3+)

Pentaho Data Integration – Spoon

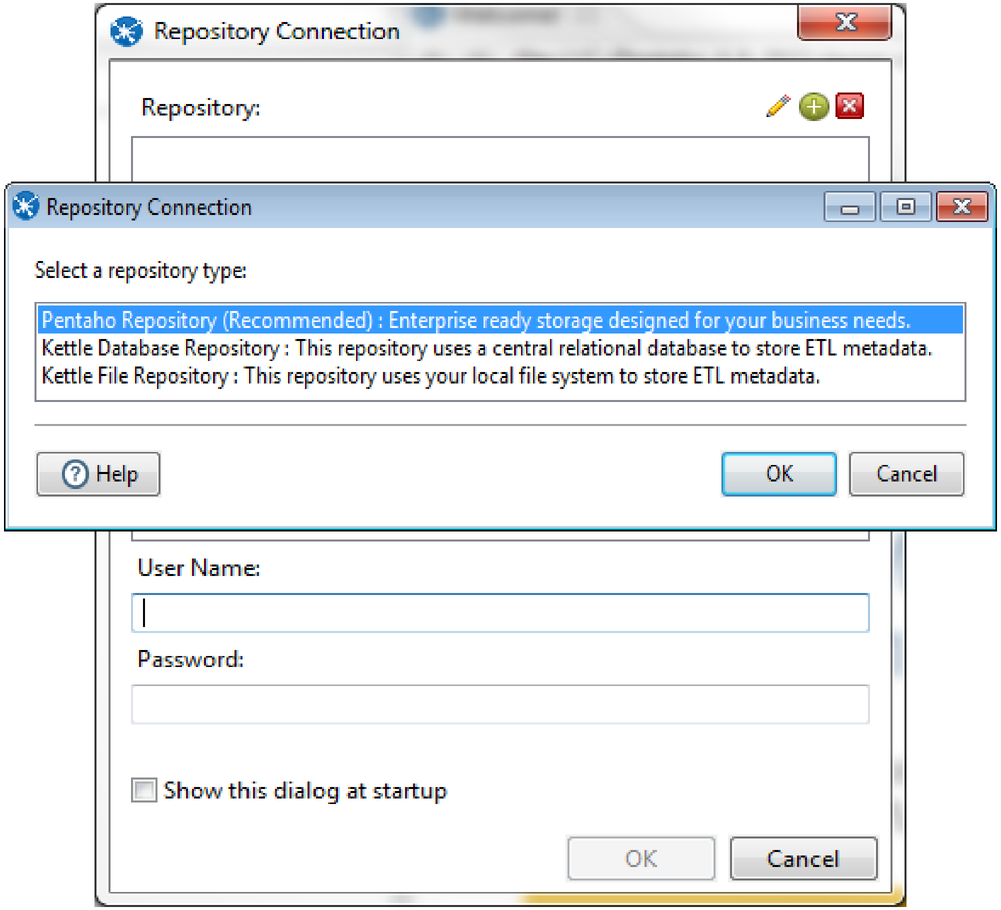
Hardware—64 bit	Operating System—64 bit
•Processors: Apple Macintosh Dual-Core •Intel EM64T or AMD64 Dual-Core RAM: 2 GB RAM for most of the design tools, PDI requires 2 GB dedicated Disk Space: 2 GB free after installation Minimum Screen Size: 1280 x 960	•Microsoft Windows 7 & 10 •Ubuntu Desktop 12.04 LTS & 14.04 LTS •OSX 10.10 & 10.11 •iOS 8.x

Repository

What is repository?

- Meta data Storage
- Provides Revision history
- Track changes
- compare revisions

- Types:
  1. File
  2. Database
  3. Enterprise Repository



Repository Connection

Select a repository type:

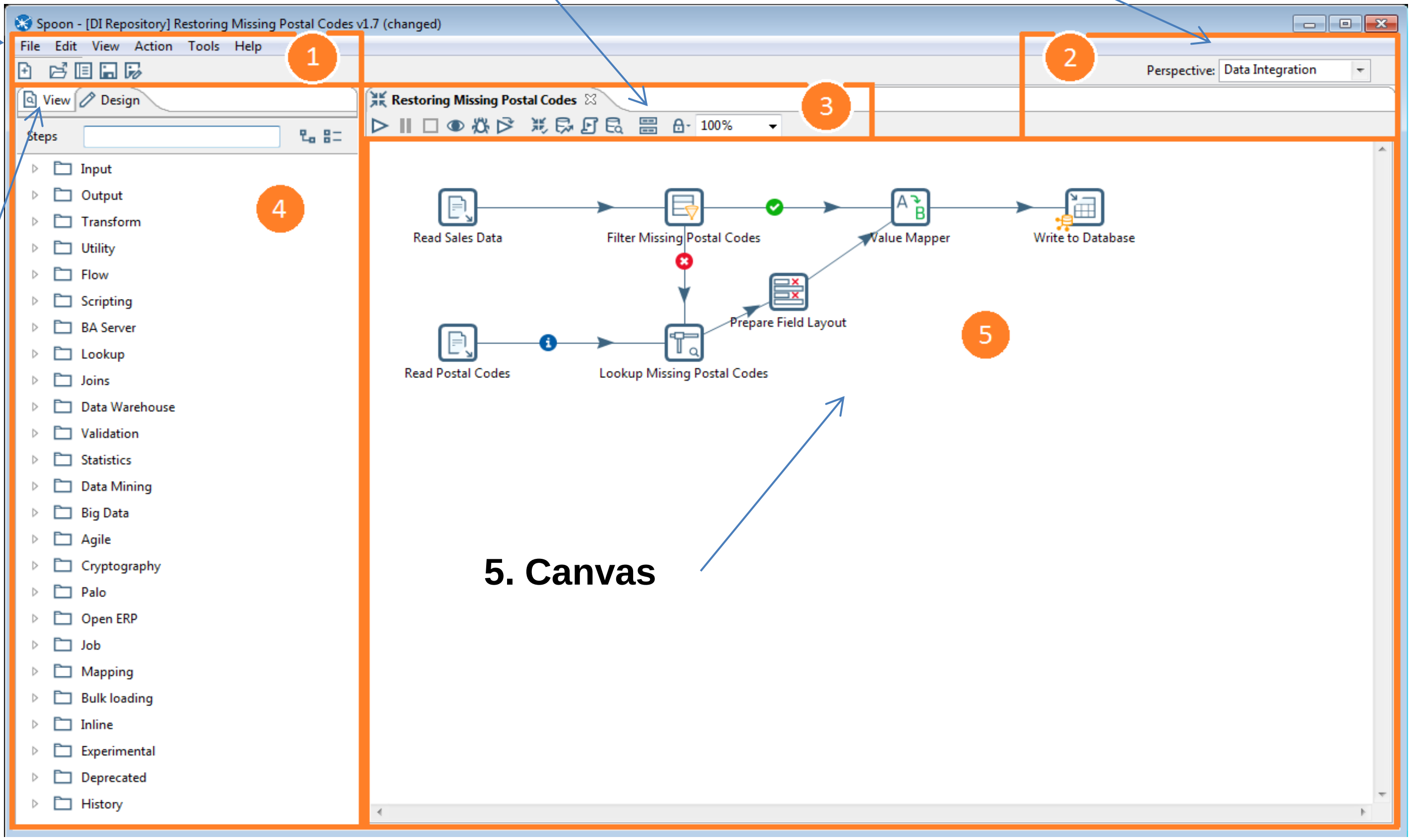
Pentaho Repository (Recommended) : Enterprise ready storage designed for your business needs.
Kettle Database Repository : This repository uses a central relational database to store ETL metadata.
Kettle File Repository : This repository uses your local file system to store ETL metadata.

Spoon

3. Sub- Toolbar

2. Perspective

1. Toolbar



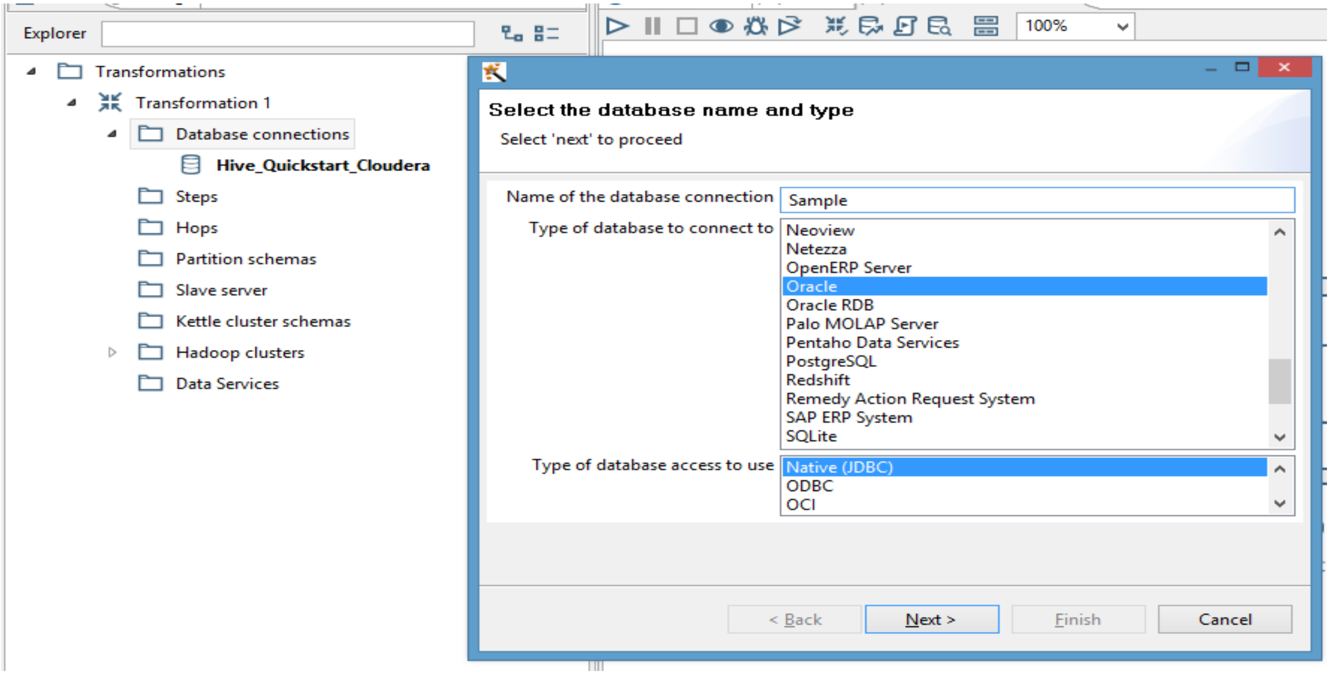
4. Design & View Tabs

5. Canvas

Pentaho Workflow

- Input steps
- Output steps
- Transformation steps
- Flow controls available in PDI
- Lookup data at various sources
- Data Validation

Database Connection



Select the database name and type

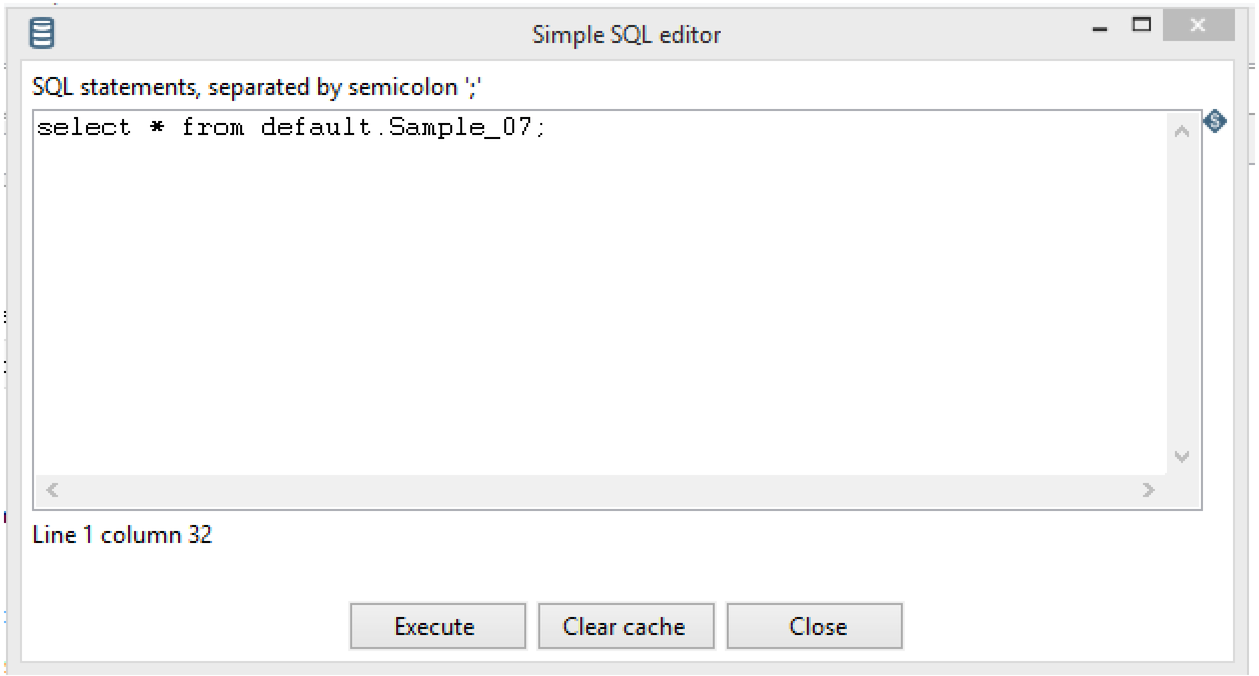
Select 'next' to proceed

Name of the database connection: Sample

Type of database to connect to:
- Neoview
- Netezza
- OpenERP Server
- Oracle
- Oracle RDB
- Palo MOLAP Server
- Pentaho Data Services
- PostgreSQL
- Redshift
- Remedy Action Request System
- SAP ERP System
- SQLite

Type of database access to use:
- Native (JDBC)
- ODBC
- OCI

SQL Editor



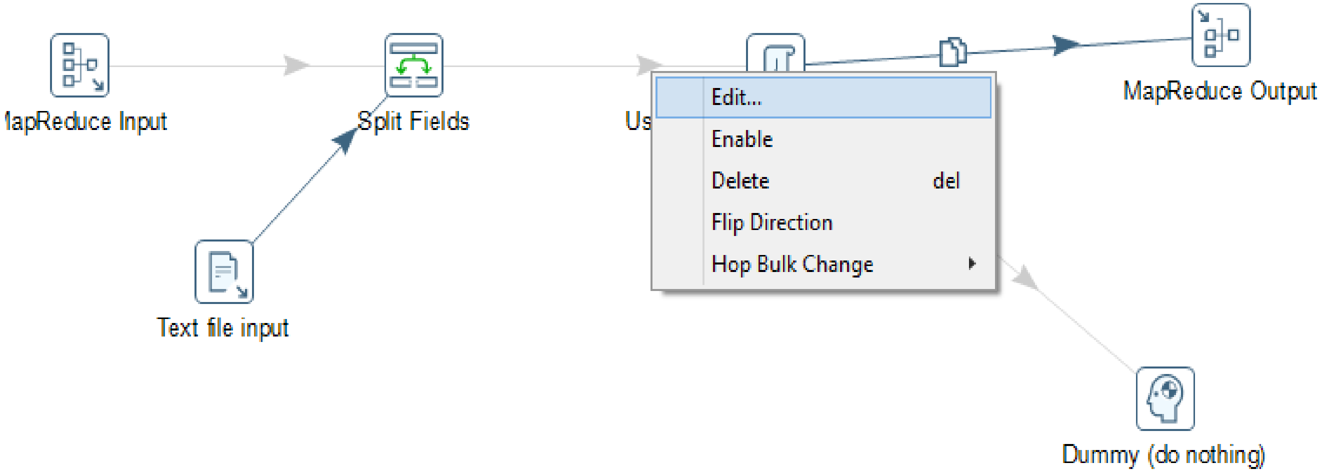
SQL statements, separated by semicolon ';'

```
select * from default.Sample_07;
```

Line 1 column 32

Hops

- Hop connects one transformation step or job with another.
- Direction of the data flow is indicated with an arrow.
- A hop can be enabled or disabled.



Variables

Definition

- Set Variable step in a transformation
- Using kettle.properties file in the directory
- Syntax UNIX - ${VARIABLE} Windows - %%VARIABLE%%

Types

Environment variables

- set an environment variable
- Java Virtual Machine (JVM) with the -D option

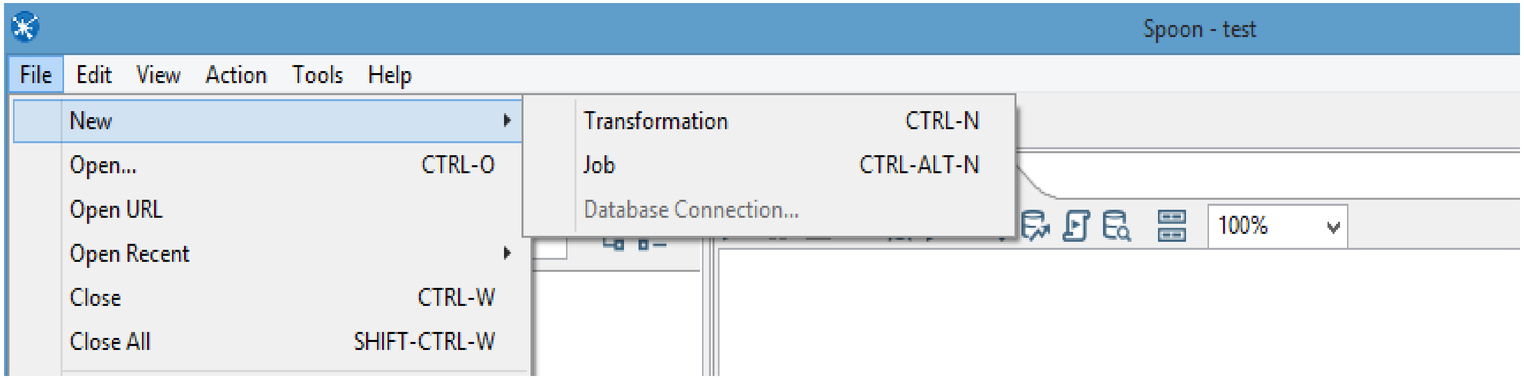
Kettle variables

- local to the job
- "Set Variable" step in a transformation

Internal variables

- Internal.Kettle.Version
- Internal.Job.Name
- Internal.Job.Filename.Name

Transformation



Transformations

Transformation Tab

Parameters

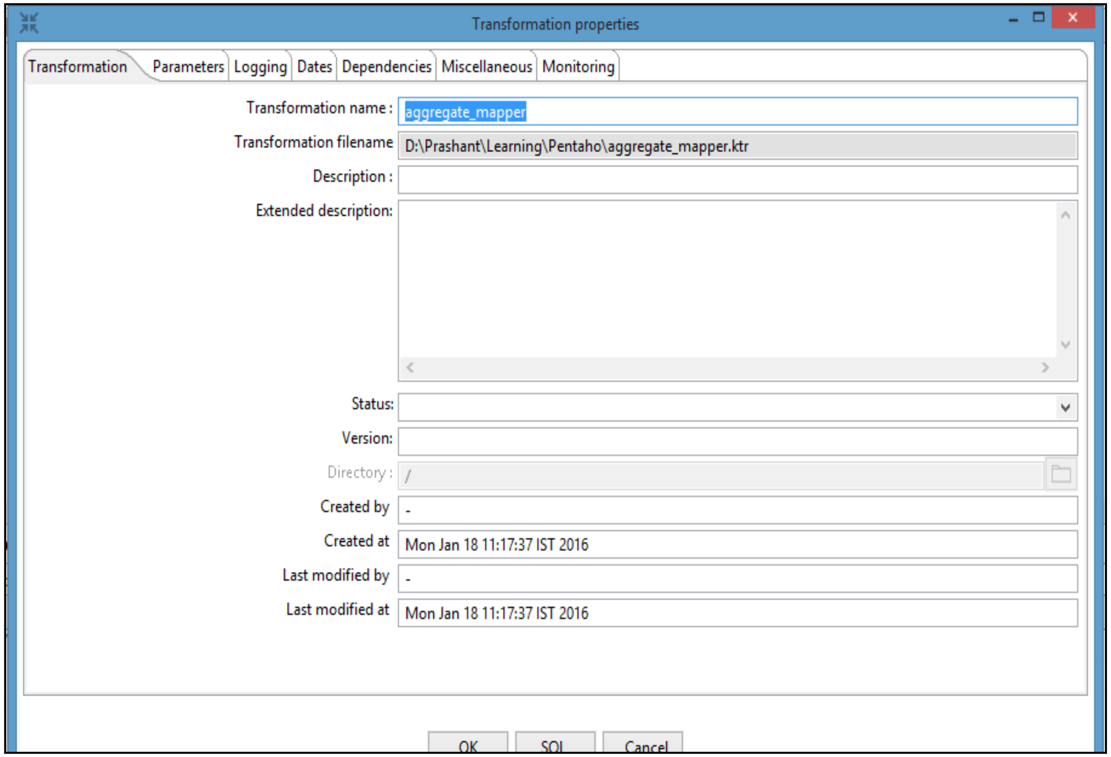
Logging

Dates

Dependencies

Miscellaneous

Monitoring



Transformation name :	aggregate_mapper
Transformation filename	D:\Prashant\Learning\Pentaho\aggregate_mapper.ktr
Description :	
Extended description:	
Status:	
Version:	
Directory :	/
Created by	-
Created at	Mon Jan 18 11:17:37 IST 2016
Last modified by	-
Last modified at	Mon Jan 18 11:17:37 IST 2016

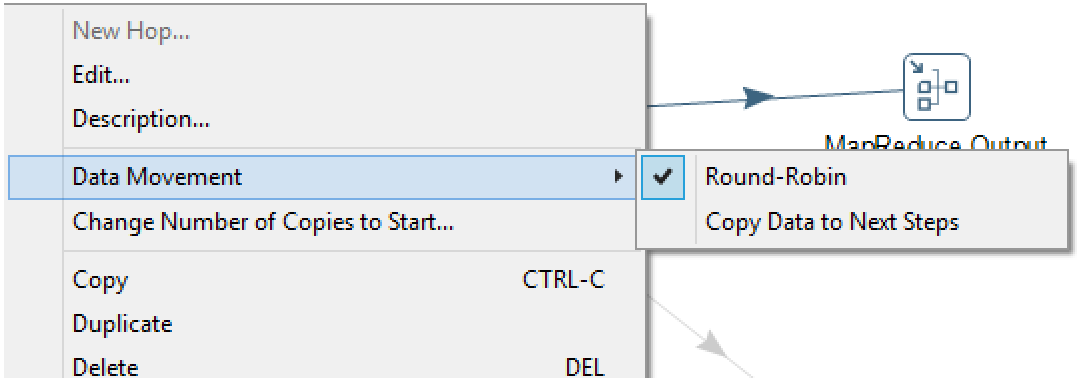
Transformation Steps

Change number of copies to start

- Launch same step several times to minimize the latency.

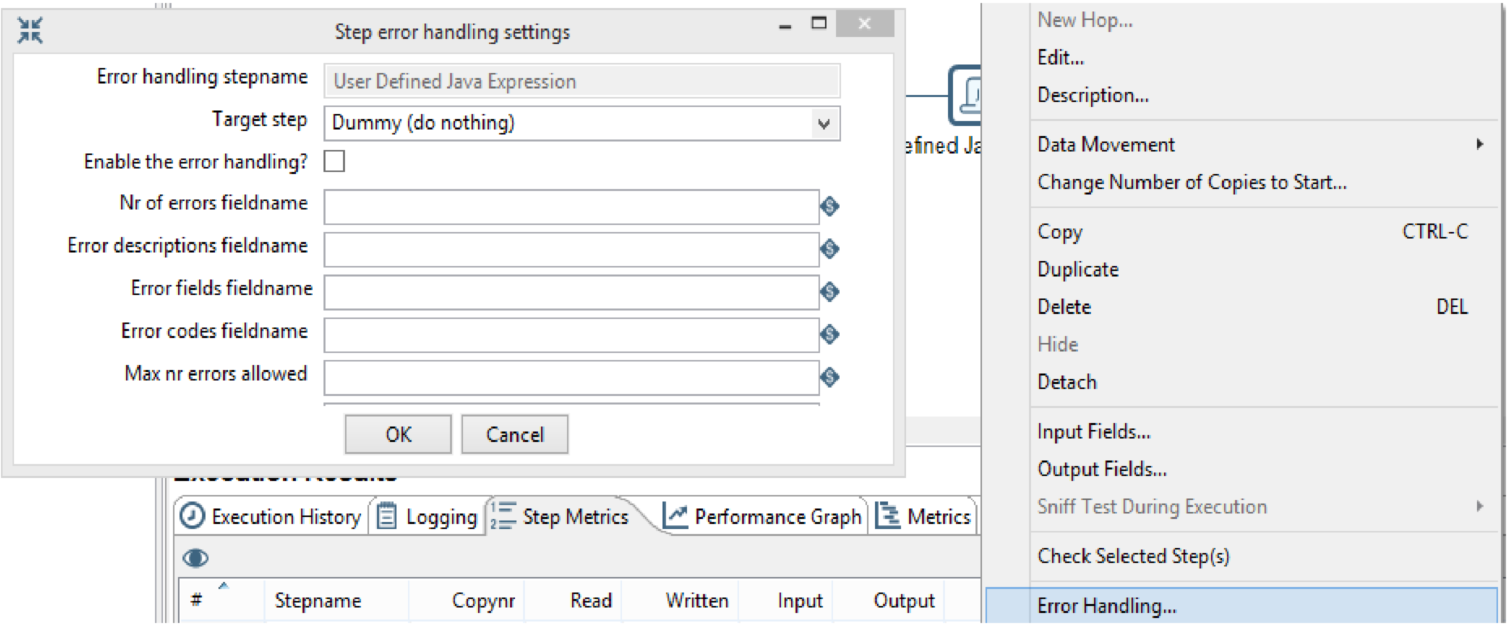
Distribute or Copy the data –

- By Default Round-Robin
- Copy data option will copy the data to all target steps



Transformation Steps

Step error handling settings –
Allows you to configure a step so that instead of halting a transformation when an error occurs, the rows that caused an error are passed to a different step.



Step error handling settings

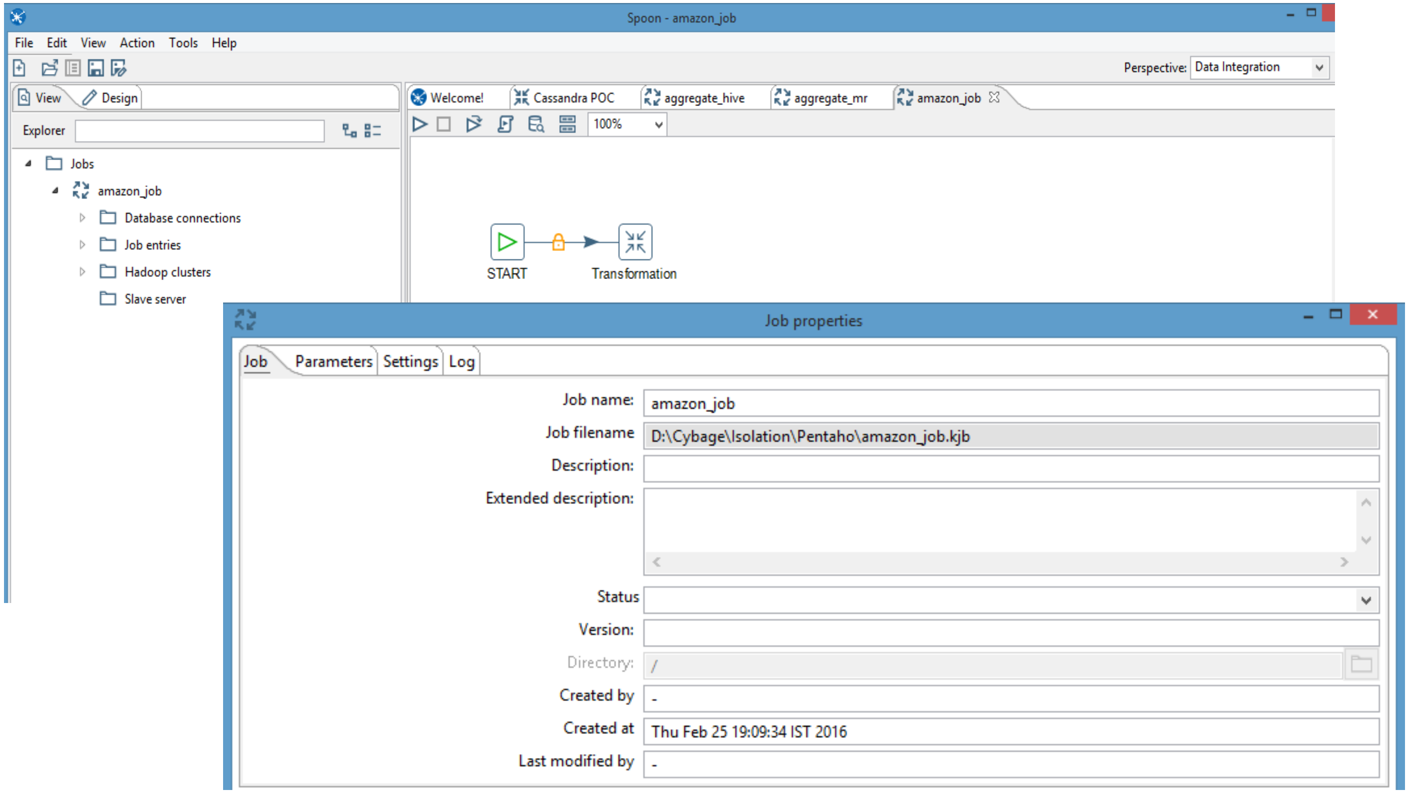
Error handling stepname	User Defined Java Expression
Target step	Dummy (do nothing)
Enable the error handling?	□
Nr of errors fieldname	
Error descriptions fieldname	
Error fields fieldname	
Error codes fieldname	
Max nr errors allowed	

Demos

- Transformations
- Joins
- Lookup

Jobs

Job Settings



Job name: amazon_job

Job filename D:\Cybage\Isolation\Pentaho\amazon_job.kjb

Description:

Extended description:

Status

Version:

Directory: /

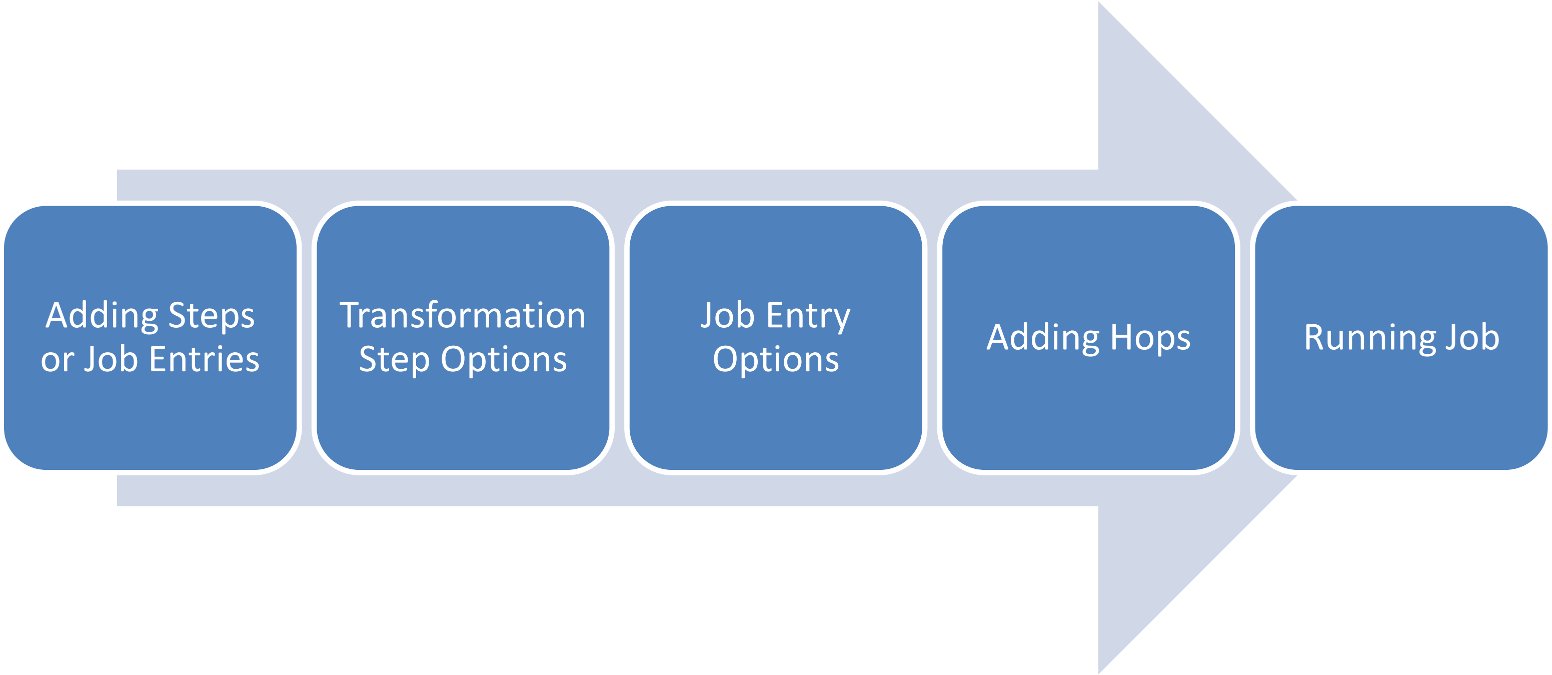
Created by -

Created at Thu Feb 25 19:09:34 IST 2016

Last modified by -

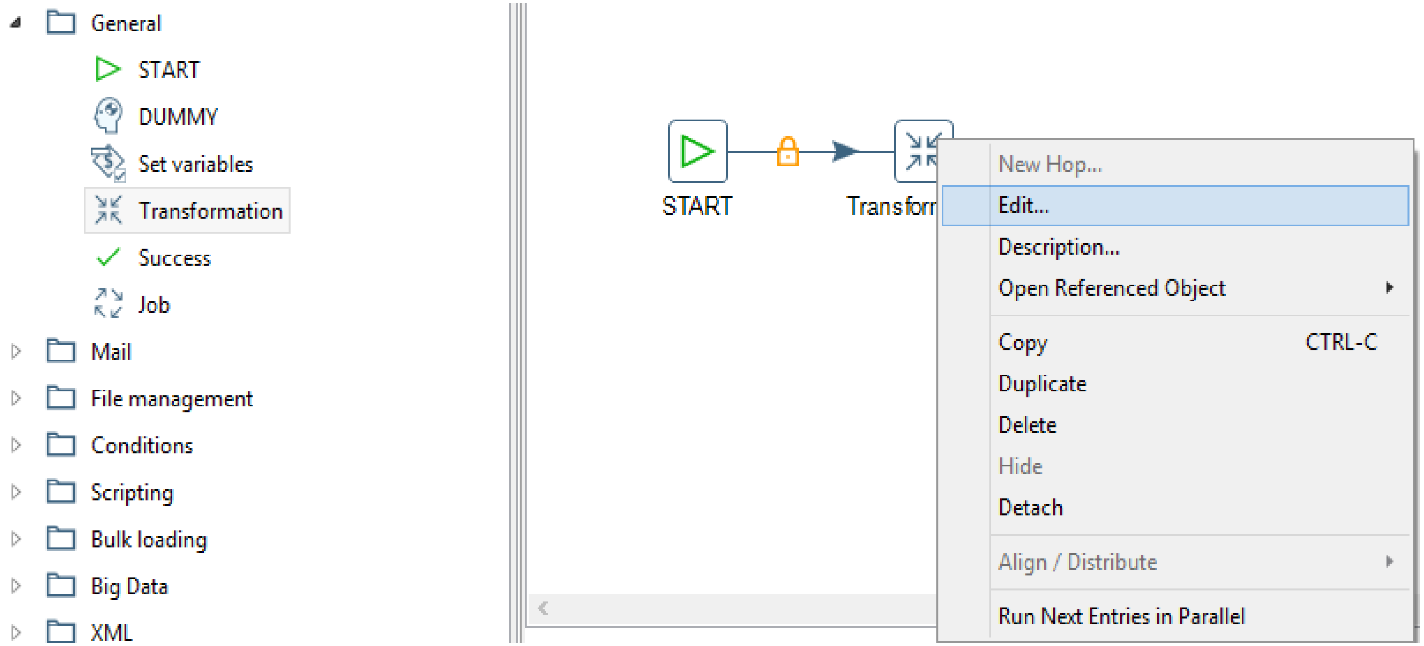
Jobs

Job Design Process



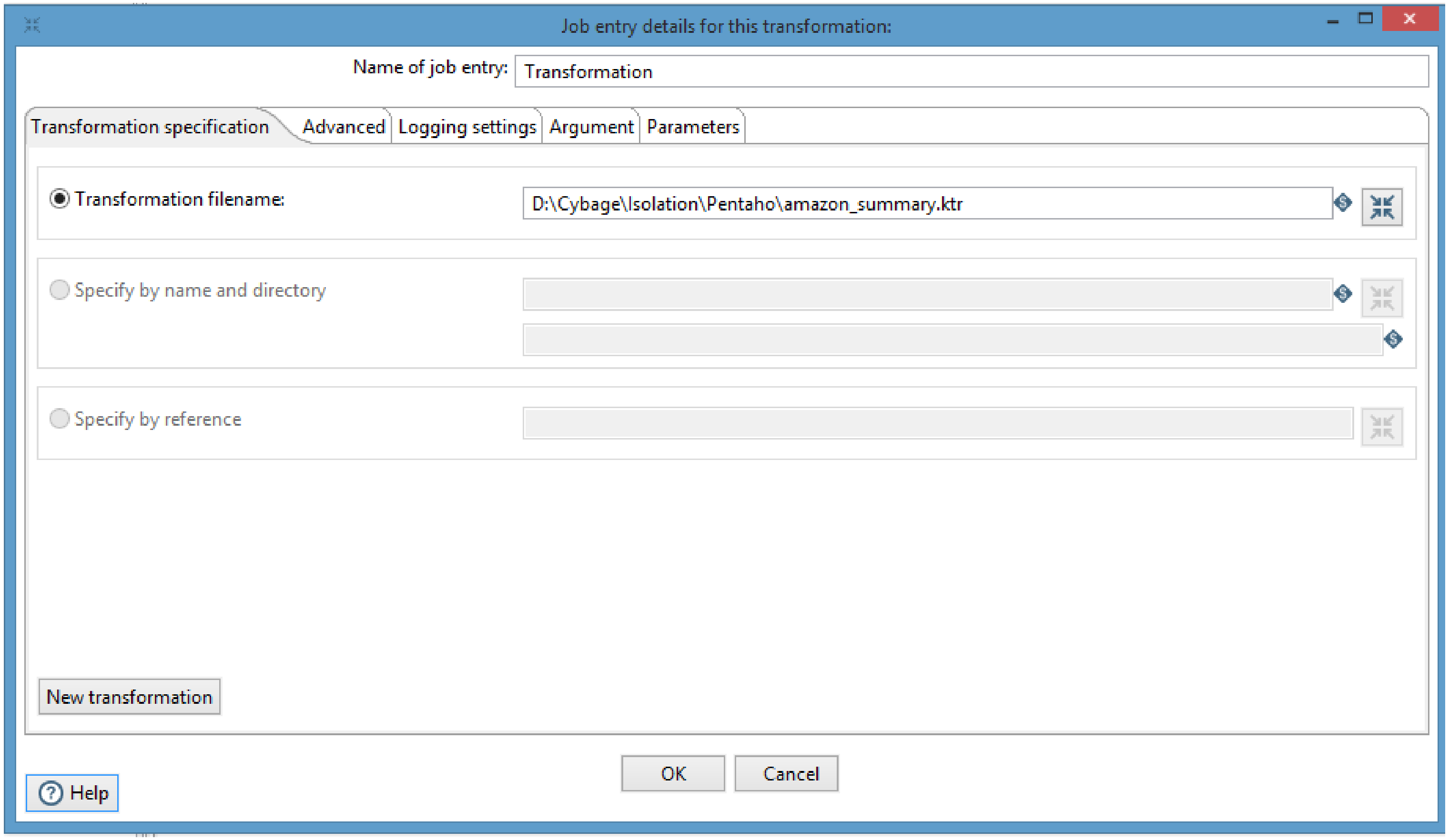
Jobs

Adding Transformation Steps



Jobs

Transformation Options



Job entry details for this transformation:

Name of job entry: Transformation

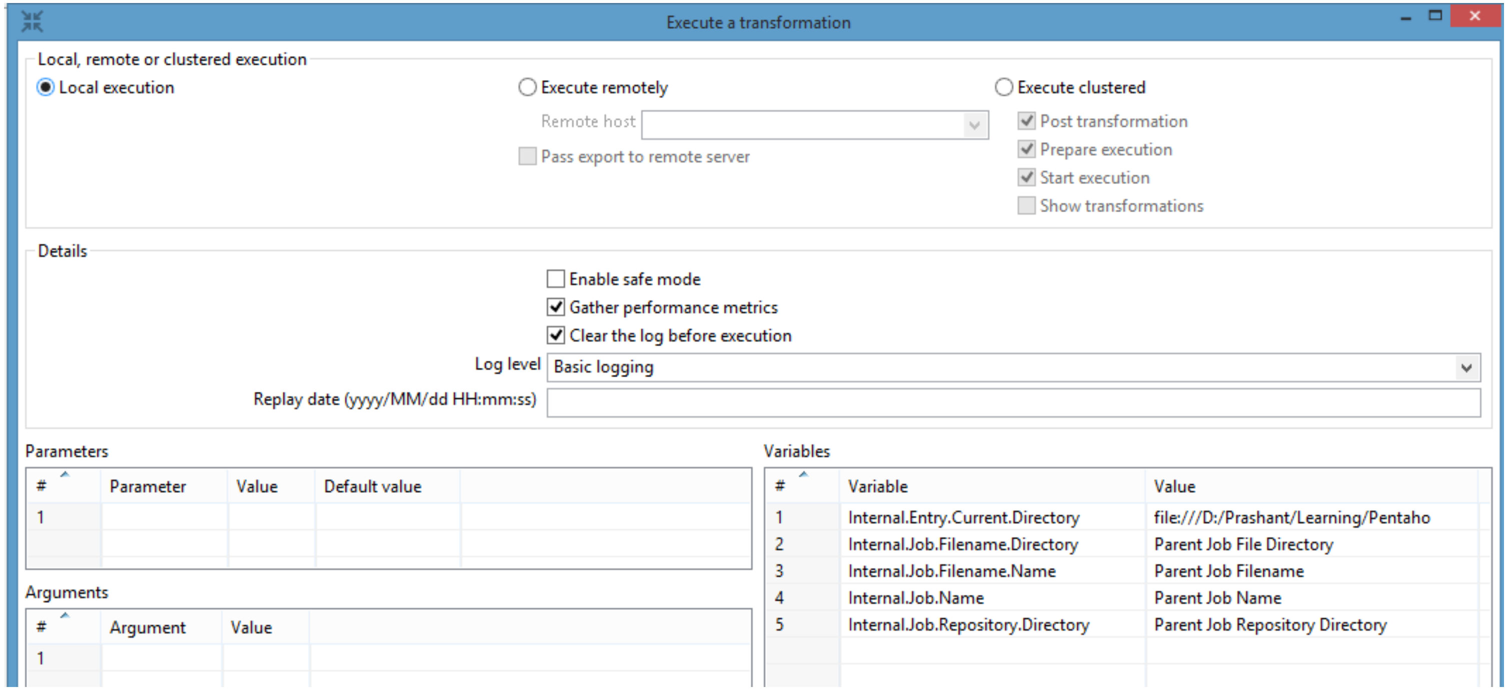
● Transformation filename: D:\Cybage\Isolation\Pentaho\amazon_summary.ktr

○ Specify by name and directory

○ Specify by reference

Run Transformations / Jobs

- **Local Execution**
- **Execute remotely**
- **Execute clustered**



Execute a transformation

Local, remote or clustered execution

● Local execution

○ Execute remotely

Remote host

□ Pass export to remote server

○ Execute clustered

☑ Post transformation

☑ Prepare execution

☑ Start execution

□ Show transformations

Details

□ Enable safe mode

☑ Gather performance metrics

☑ Clear the log before execution

Log level: Basic logging

Replay date (yyyy/MM/dd HH:mm:ss):

Parameters

#	Parameter	Value	Default value
1			

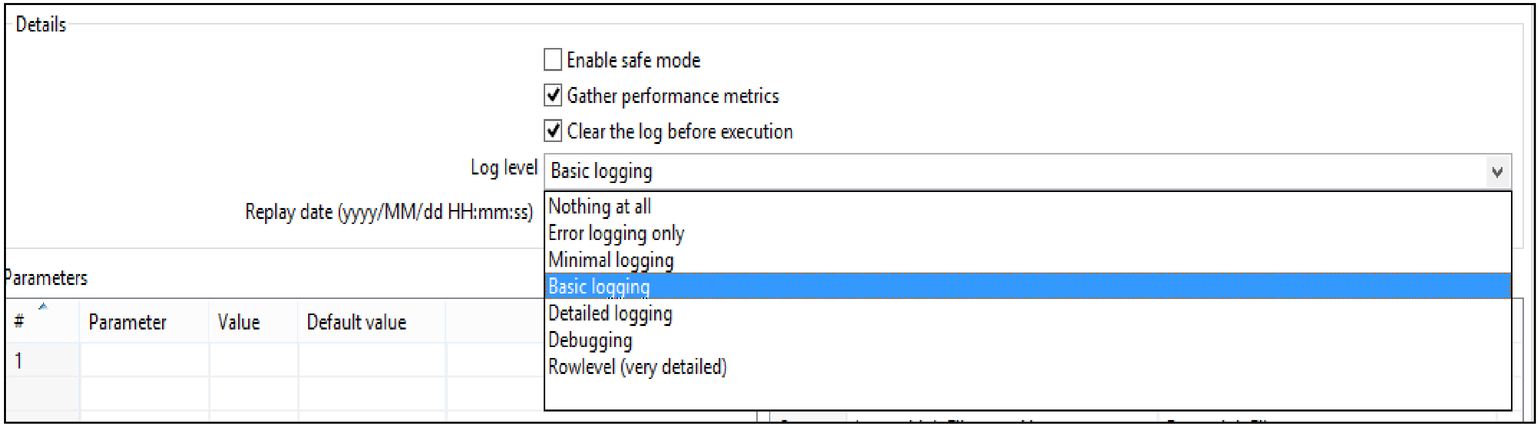
Arguments

#	Argument	Value
1		

Variables

#	Variable	Value
1	Internal.Entry.Current.Directory	file:///D:/Prashant/Learning/Pentaho
2	Internal.Job.Filename.Directory	Parent Job File Directory
3	Internal.Job.Filename.Name	Parent Job Filename
4	Internal.Job.Name	Parent Job Name
5	Internal.Job.Repository.Directory	Parent Job Repository Directory

Logging



Details

☐ Enable safe mode

☑ Gather performance metrics

☑ Clear the log before execution

Log level: Basic logging

Replay date (yyyy/MM/dd HH:mm:ss)

- Nothing at all
- Error logging only
- Minimal logging
- Basic logging
- Detailed logging
- Debugging
- Rowlevel (very detailed)

Parameters

#	Parameter	Value	Default value
1			

- Nothing: Don't show any output
- Error: Only show errors
- Minimal: Only use minimal logging
- Basic: This is the default basic logging level
- Detailed: Give detailed logging output
- Debug: For debugging purposes, very detailed output.
- Row level: Logging at a row level, this can generate a lot of data.

Demos

- Jobs

Building Dimensional Model

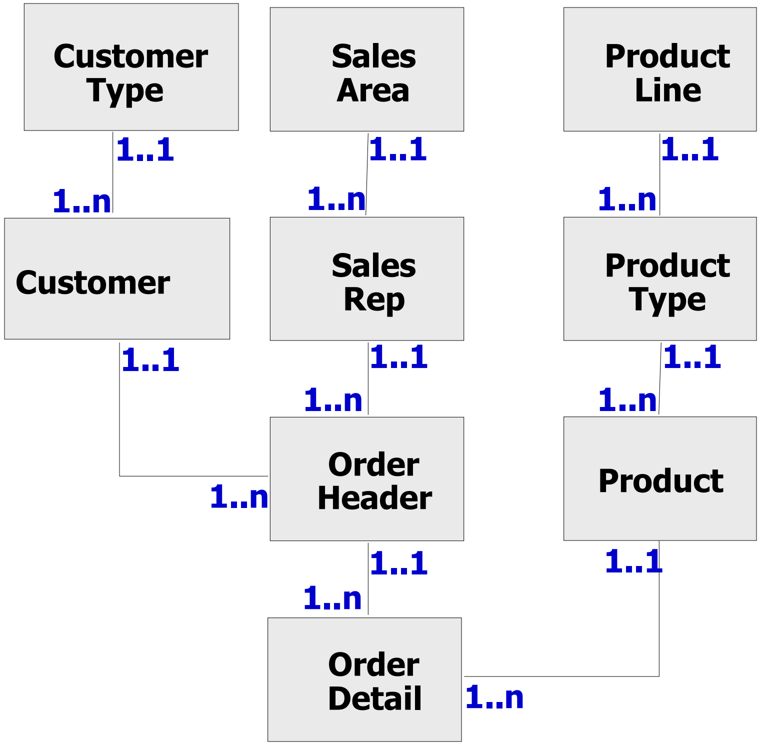
The four key decisions made during the design of a dimensional model

- Identify the Source Data for business process.
- Define the grain of data .
- Identify the dimensions.
- Identify the facts.

Operational vs Reporting Databases

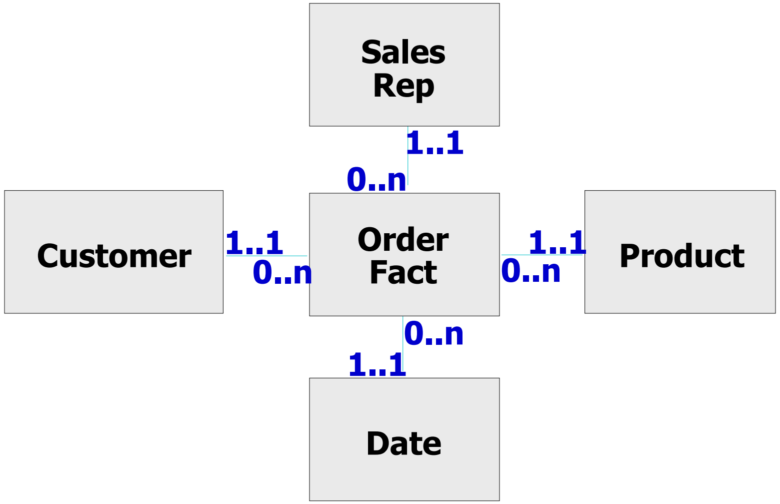
- Relational databases are typically either:

Operational



Customer Type
1..1
1..n
Customer
1..1
1..n
Sales Area
1..1
1..n
Sales Rep
1..1
1..n
Order Header
1..1
1..n
Order Detail
1..n
Product Line
1..1
1..n
Product Type
1..1
1..n
Product
1..1

Reporting



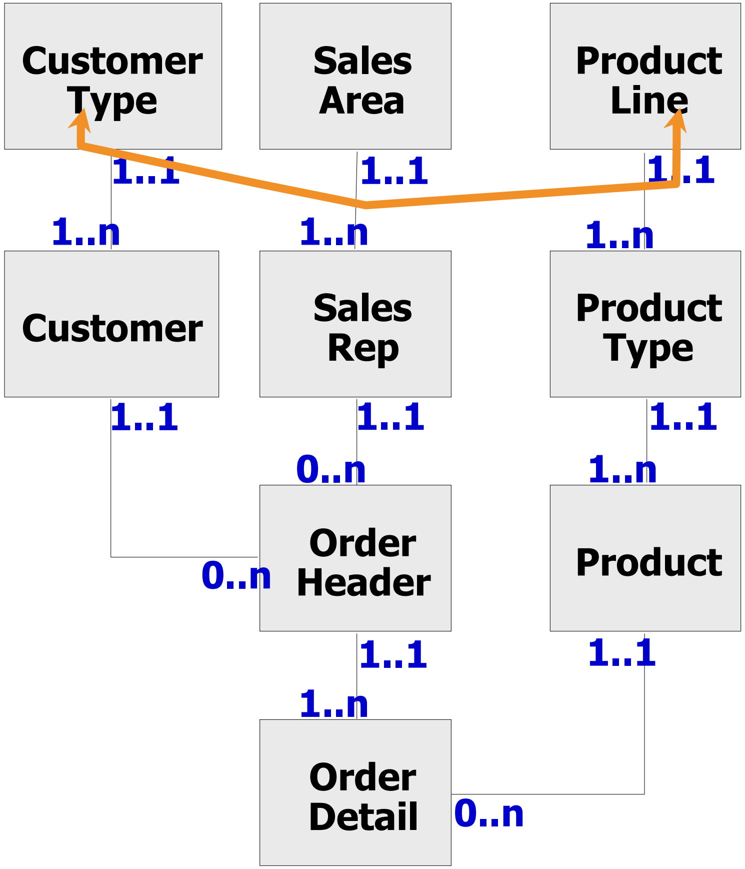
Sales Rep
1..1
0..n
Customer
1..1
0..n
Order Fact
0..n
1..1
Product
0..n
1..1
Date

Features of an Operational Database

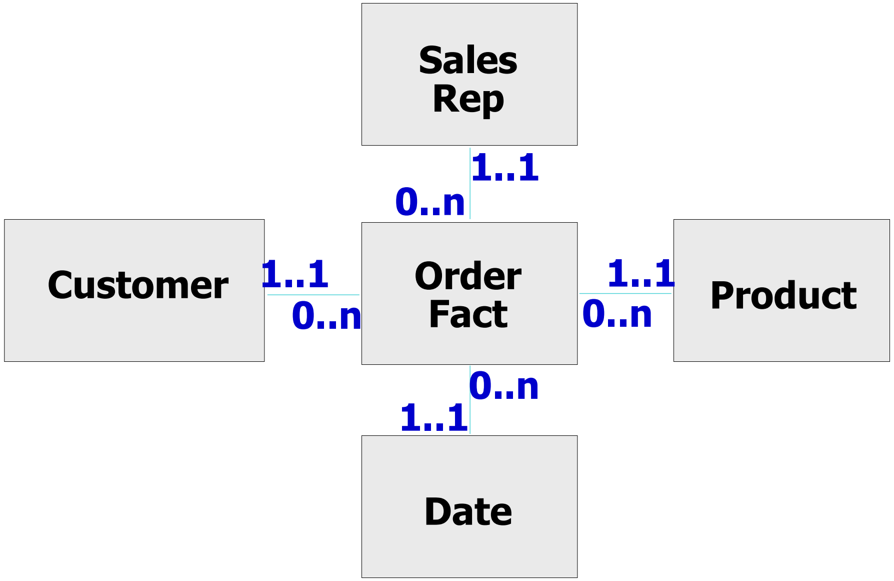
- Operational databases:
  - –are designed to maximize accuracy and minimize redundancy
  - –are optimized for writing/updating data rather than reading data
  - –often result in monolithic designs with multiple joins
  - –Large queries can perform slowly.

Identify Issues with Operational Databases

- “Show all customer types that bought from a product line.”
- The query must check data in seven tables before returning a result set.



Reporting Databases (Star Schema Design)

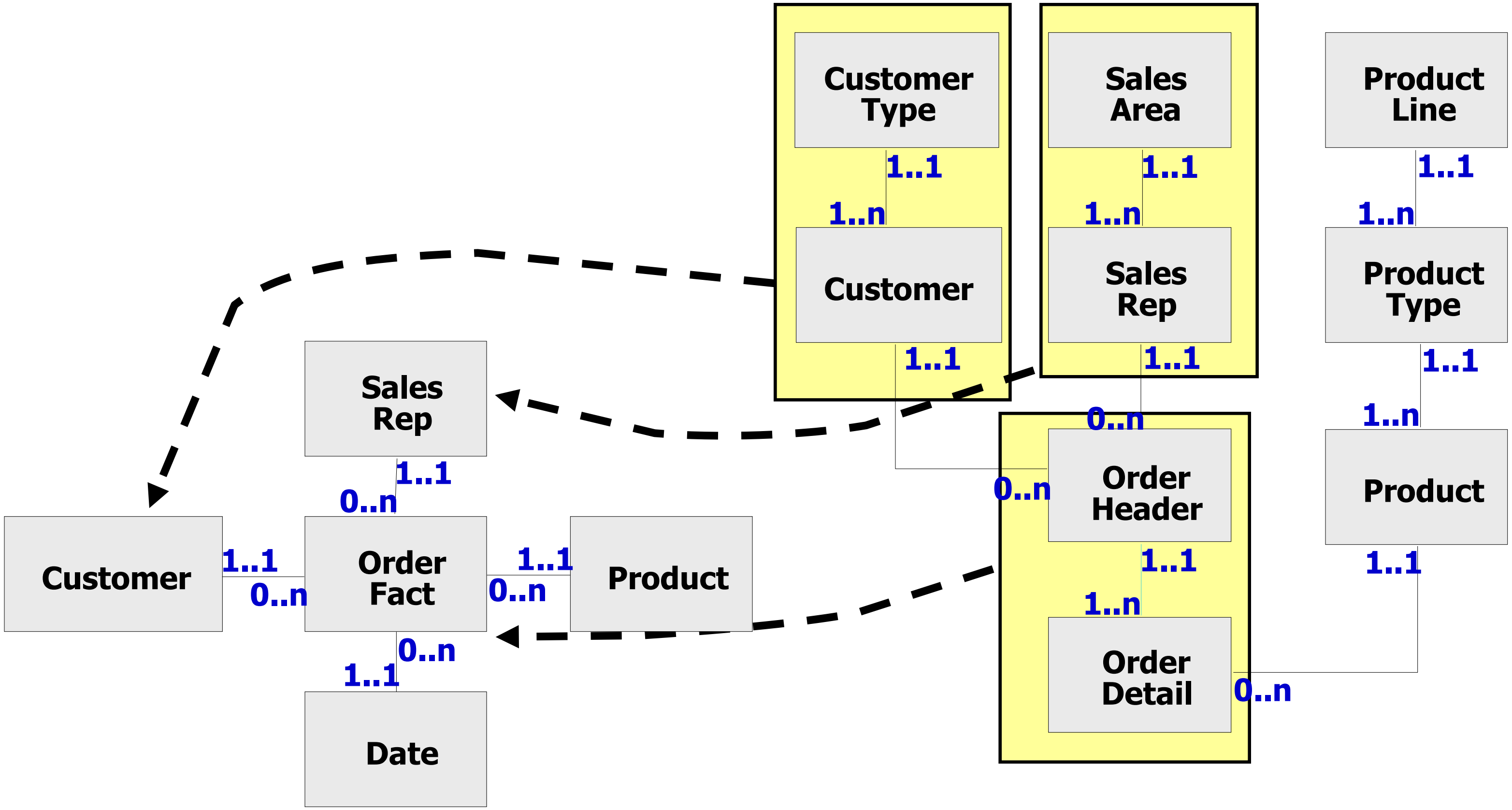


- Transactional data is stored in a fact table
- Reference data is stored in separate dimension tables

- **same information, but five tables instead of nine**

Create a Star Schema

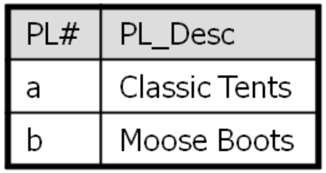
- Collapse the relationships to form dimensions (perspectives).



Examine Operational Data

- Data is normalized

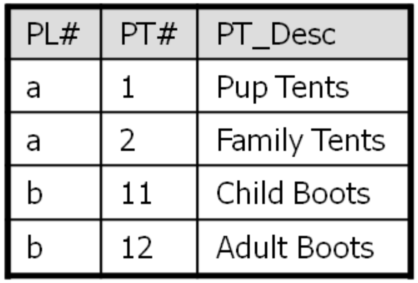
Product Line Table



PL#	PL_Desc
a	Classic Tents
b	Moose Boots

2 rows

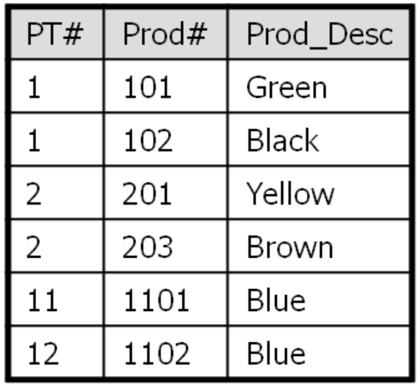
Product Type Table



PL#	PT#	PT_Desc
a	1	Pup Tents
a	2	Family Tents
b	11	Child Boots
b	12	Adult Boots

4 rows

Product Table



PT#	Prod#	Prod_Desc
1	101	Green
1	102	Black
2	201	Yellow
2	203	Brown
11	1101	Blue
12	1102	Blue

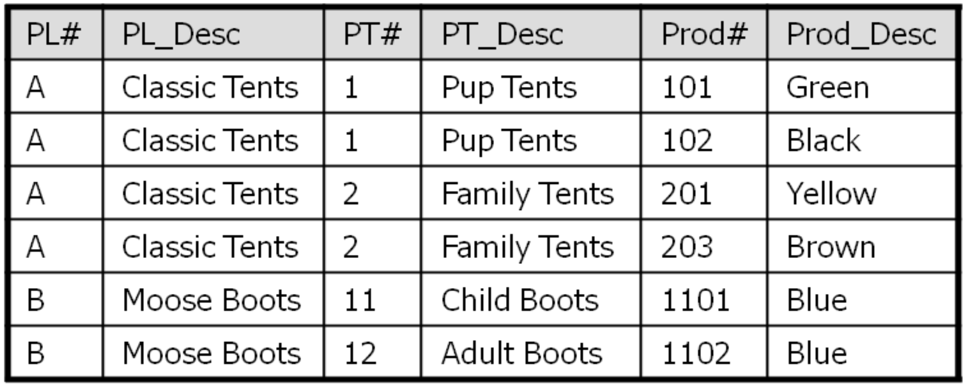
6 rows

Before collapsing into a star schema dimension

Examine Reporting Data

- Data is de-normalized

Product Dimension Table



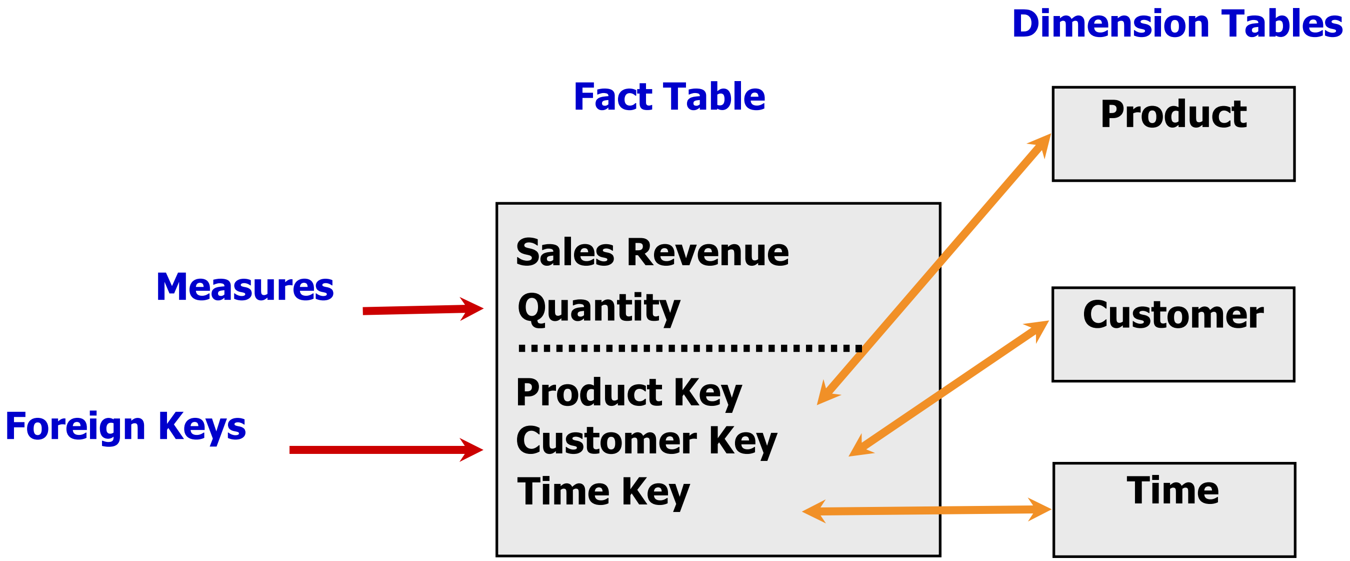
PL#	PL_Desc	PT#	PT_Desc	Prod#	Prod_Desc
A	Classic Tents	1	Pup Tents	101	Green
A	Classic Tents	1	Pup Tents	102	Black
A	Classic Tents	2	Family Tents	201	Yellow
A	Classic Tents	2	Family Tents	203	Brown
B	Moose Boots	11	Child Boots	1101	Blue
B	Moose Boots	12	Adult Boots	1102	Blue

6 rows

After collapsing into a star schema dimension

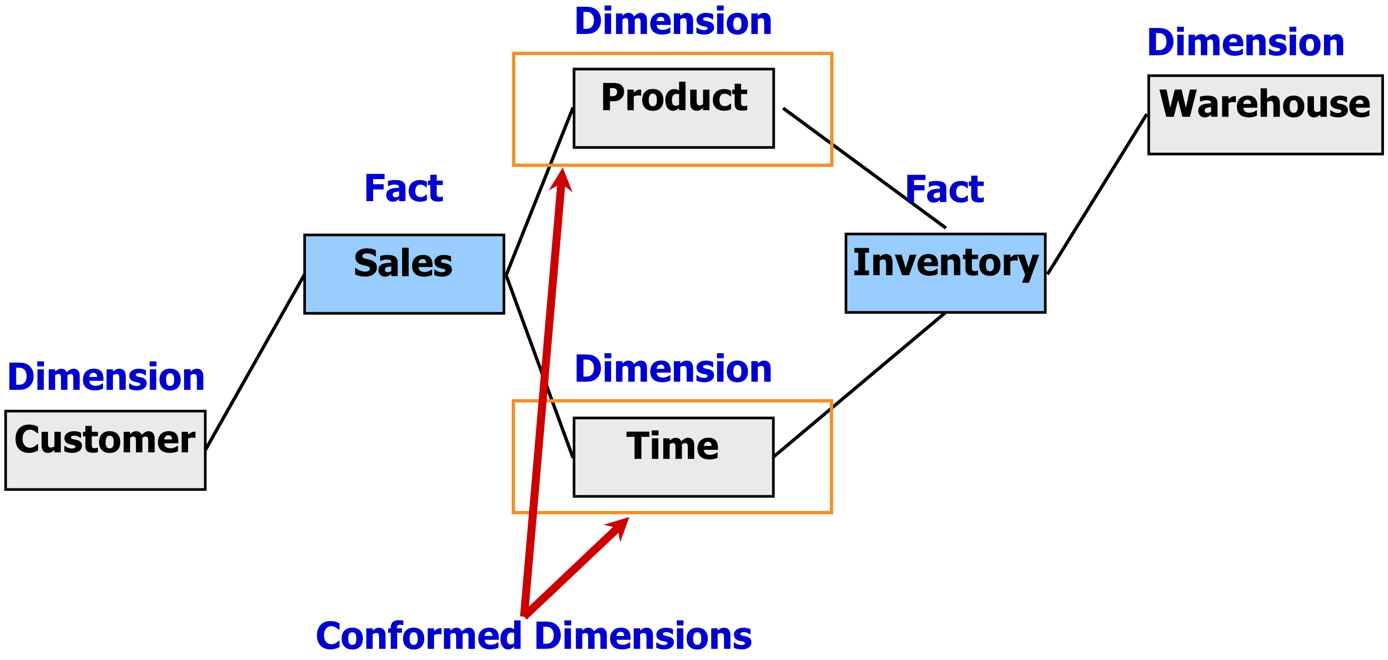
Fact Tables

- Fact tables contain the (usually additive) numbers by which a company measures itself:
  - –Standard Selling Price - not additive
  - –Sale Amount - additive



Dimension Tables

- Dimension tables provide descriptive information.
- Dimension tables may be “conformed” so that they are applicable to multiple fact tables.



Dimension Types

- What is SCD?
- SCD Type 1
- SCD Type 2
- SCD Type 3

Dimension Types

Slowly Changing Dimension –
Type 1: Overwrite

Before:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

After:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	IL

Dimension Types

Slowly Changing Dimension –

Type 2: Add new row

Before:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

After:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State	Start_Date	End_Date
123	ABC	Acme Supply Co	CA	01-Jan-2000	21-Dec-2004
124	ABC	Acme Supply Co	IL	22-Dec-2004	

Dimension Types

Slowly Changing Dimension –

- Type 3: Add new attribute

Before:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

After:

Supplier _Key	Supplier_ Code	Supplier_ Name	Original_Supplier _State	Effective_Date	Current_Supplier _State
123	ABC	Acme Supply Co	CA	22-Dec-2004	IL

Fact Types

Factless fact tables

Most Fact Tables are used to capture numerical results, but it is possible that the event merely records a set of dimensional entities coming together at a moment in time.

Such Fact table will have foreign keys from all related dimension tables without having any particular fact entry.

Example, an event of a student attending a class on a given day may not have a recorded numeric fact

Fact Types

Aggregate fact tables

- *Aggregate fact tables* are simple numeric rollups of atomic fact table data.

- Achieve improved query performance.

- Materialized views can serve as aggregate facts

- BI tools can choose appropriate (aggregated or atomic) aggregate level at query time.

Demos

Datawarehouse

- Building Dimension
- Building Fact Tables

Important Links

- Download Link: https://sourceforge.net/projects/pentaho/
- Documentation: http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation

Any Questions?

Thank you!