# Training Basic ELT using Talent

## Task 0: Prepare Data Source

<https://hub.docker.com/r/aa8y/postgres-dataset/>

```

docker pull aa8y/postgres-dataset

docker run -d --name pg-ds -p 5432:5432 aa8y/postgres-dataset:dellstore

```

## Task 1: Download Talend

Download Talend Openstudio for DataIntegration [Open Source Version]

https://info.talend.com/rs/talend/images/CB\_EN\_DI\_Cookbook\_DataIntegration.pdf

<https://help.talend.com/>

## Task 2: Database Setup

Create new Database “dwh” in the postgresql docker image

Prepare Database schema (according to Excel File: Overview of schemata)

## Task 3: Staging

Create staging tables (1:1 copy plus metadata columns) for the tables customer, product, categories.

Build a Job “stage <tablename>” for each table, which copies the source data to the staging tables using a TRUNCATE INSERT strategy.

Build a central job called “staging”, which includes the child jobs. For now populate the LOAD\_ID variable with the value 1.

Test the jobs, until all tables in staging have the same number of columns as in the source database

## Task 4: Simple Dimension

Build first Dimension (Produkt Dimension)

1. Create table prod\_dim with the columns mentioned in the dellstore\_sourcetotarget\_map in the integration schema as well as in the dwh schema
2. Build a Job “int <tablename” to load the table in the integration schema using a TRUNCATE INSERT method according to the dellstore\_sourcetotarget\_map . Populate the PROD\_ID using a database sequence.
3. Create a central job “integration” and include the “int <tablename>” job in it.
4. Build a Job “load <tablename>” which compares all data in the table in the int schema with the data in the dwh schema and copies the data to the new environment using an INSERT/UPDATE method.
5. Create a central job “load” and include the “load <tablename>” job in it.
6. Test the new jobs by running it multiple times. The number of rows in all product tables should stay the same

## Task 5: Time dimension

Build a constant dimension

1. Create table time\_dim with the columns mentioned in the dellstore\_sourcetotarget\_map in the integration schema as well as in the dwh schema
2. Check, which dates are in the source tables (check orders.orderdate and orderliness.orderdate). Take the minimum date minus 5 years as start and the maximum date plus 10 years as end date for the next step.
3. Build a Job “int <tablename” to load the table in the integration schema using a TRUNCATE INSERT method, based on date values from a date generator.
4. Include the “int <tablename>” job in the central job “integration”.
5. Build a Job “load <tablename>” which compares all data in the table in the int schema with the data in the dwh schema and copies the data to the new environment using an INSERT/UPDATE method.
6. Include the “load <tablename>” job in the central job “load”.
7. Test the new jobs by running it multiple times. The number of rows in all time tables should stay the same

## Task 6: Load Control

1. Create a new central job “daily run”, which includes the central job staging, integration and load. Add dependencies between the three jobs, so that they run in the correct order.
2. Create the DWHLOAD table in the dwh schema (only).
3. Add a new job, which reads the maximum LOAD\_ID from the DWHLOAD table. If no row is stored in the table the value 0 should be returned. Increase the ID by one and store it in the parameter “LOAD\_ID”. Also write a new line to the DWHLOAD table containing the current date, and the updated LOAD\_ID.
4. Update all jobs to use the parameter to populate the LOAD\_ID.
5. Test the central job, test that each execution of the job generates a new LOAD\_ID and a new line in the DWHLOAD Table

## Task 7: Customer Dimension

1. Create table cust\_dim with the columns mentioned in the dellstore\_sourcetotarget\_map in the integration schema as well as in the dwh schema
2. Build a Job “int <tablename” to load the table in the integration schema using a TRUNCATE INSERT method according to the dellstore\_sourcetotarget\_map . Populate the CUST\_ID using a database sequence.
3. Create a central job “integration” and include the “int <tablename>” job in it.
4. Build a Job “load <tablename>” which loads the table

## Task 8: Sales Fakt

## Task 9: Cleaning up

Add the tables orders and orderliness to staging.

Task 8:

Use context variables to replace the hardcoded database names