

|  |
| --- |
| EPAM Systems, RD Dep.  Talend Data Integration Exit Task |
| **BI. Talend Data Integration** |

Content

[Talend Data Integration Exit Task 1](#_Toc504039681)

[1 TASK Description 3](#_Toc504039682)

[1.1 Preparation 3](#_Toc504039683)

[1.2 Tasks 3](#_Toc504039684)

[2 TASK implementation 4](#_Toc504039685)

[2.1 LOADING CUSTOMERS 4](#_Toc504039686)

[2.2 LOADING PRODUCTS 4](#_Toc504039687)

[2.3 LOADING PAYMENTS 4](#_Toc504039688)

[2.4 ADDING isinterstate column 6](#_Toc504039689)

[2.5 ADDING Foreign keys 7](#_Toc504039690)

# TASK Description

## Preparation

1. Download and unarchive Talend Open Studio for Data Integration.
2. Create project.
3. Created jobs and other repository elements should be Exported and pushed to separate repo at github.
4. Link to the repo need to be added to google sheet <https://docs.google.com/spreadsheets/d/1F_kgL2a0ZjAUrg4_LNdugsvSfIJEHfWZ-ckC3yEhFeU/edit?usp=sharing>
5. Optionally you can add text document with a description of your jobs.

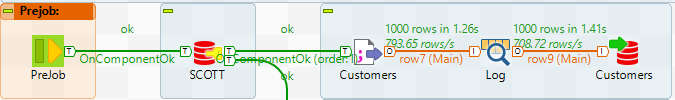
## Tasks

Three files provided customers.csv, products.csv, Payments\_yyyymmdd.csv. Your goal is to create Data Mart based on this data. Create dimensions at your choice. Add SK where needed.  
Additionally calculate and add to a table KPI isInterState = 1 in case byer and warehouse State differ, or 0 if State is the same.

# TASK implementation

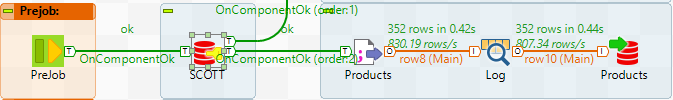
Данные о клиентах и продуктах были оставлены в корне папки, для оплат была дополнительно создана папка, куда были вынесены необходимые файлы.

## LOADING CUSTOMERS



Первым шагом были загружены клиенты. Загрузка клиентов осуществлялась в masterjob с помощью компонента prejob. В таблице был установлен ключ по полю customerid. Чтобы избежать дублирование записей при повторной загрузке был выбран метод update-insert.

## LOADING PRODUCTS

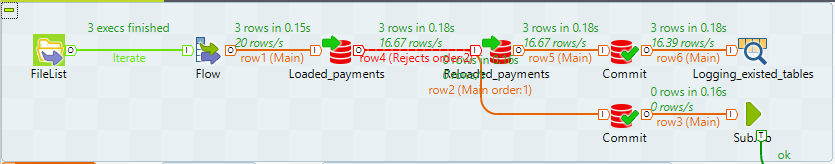


Вторым шагом были загружены продукты. Загрузка продуктов также осуществлялась в masterjob с помощью компонента prejob. В таблице был установлен ключ по полю productid. Чтобы избежать дублирование записей при повторной загрузке был выбран метод update-insert.

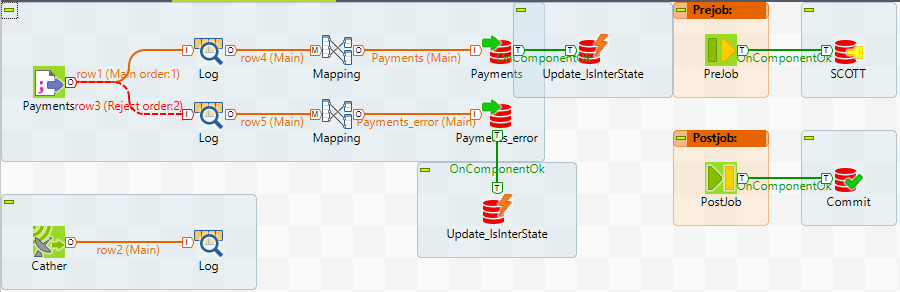
Таблицы клиентов и продуктов выступают в качестве дименшенов.

## LOADING PAYMENTS

Третьим шагом загружались оплаты, которые выступают в качестве фактовой таблицы. Загрузка оплат осуществлялась в несколько шагов с помощью subjob. Изначально в таблице не было установлено никаких ключей. Дублирование файлов при загрузке проверяется с помощью дополнительной таблицы loaded\_payments. Если файла в таблице нет, то он записывается, если есть – пропускается.

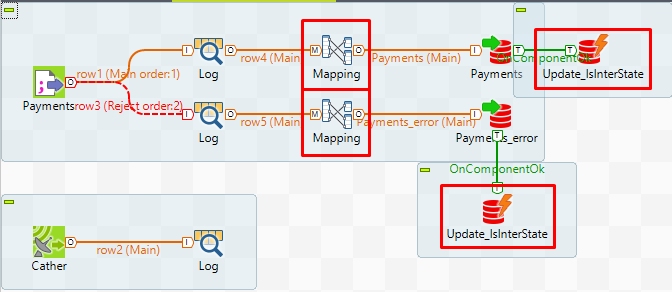


В subjob в загрузка разделяется на два потока. Записи с ошибками будут отправлены в таблицу с некорректными данными.



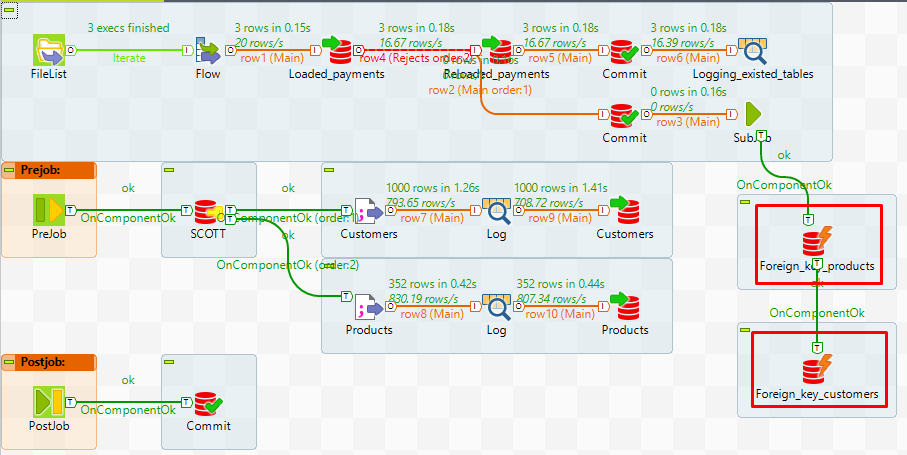
Корректные записи будут отправлены в таблицу с payments. Также в subjob учавствует cather, который позволяет отслеживать ошибки.

## ADDING isinterstate column



1. Добавление колонки в фактовую таблицу происходит в subjob.
2. В элементе mapping добавляется колонка и ей присваивается нулевое значение.
3. В элементе toraclerow реализуется обновление колонки. При выполнение условия customer.state = product.warehouselocationstate значение isinterstate колонки обновляется и переходит в 1, в противном случае нет.

## ADDING Foreign keys



1. Добавление внешних ключей в фактовую таблицу происходит в masterjob. Primary key в фактовую таблицу не добавлялся, поскольку это не обязательное требование при проектировании DWH.

После загрузки данных в таблицу происходит обращение к элементу toraclerow. Там реализуется alter table и добавление внешних ключей в фактовую таблицу по полям customerid и productid.