**Git\_link:** [**https://github.com/OksanaKrangach/DQE\_Intermedia\_Final.git**](https://github.com/OksanaKrangach/DQE_Intermedia_Final.git)

**Final Tasks**

# Business requirements

Input data set contains 3 csv files: airports, carriers, flights



Business description and requirements for source data can be found in Flights\_Data\_Set\_Requirements.xlsx.



1. Data set that should be processed is uploaded to ‘source’ folder in GSI data lake (csv)
2. Target data is stored in ‘Raw’ folder (parquet)

**Task\_1. Provide a check list for required data quality checks.**

**Task\_1:**

Check list – in the separate file: **Flights\_Check\_List\_Oksana\_Krangach.xlsx**

**A screenshot of a computer

Description automatically generated**

**Task\_2. Implement DQ using Jupyter Notebook**

For the final task my set of DQ checks are the next:  
Carrier: Completeness by records  
Airports: Consistency check for state

Flights: Consistency for CancellationCode  
Flights: Consistency check for CRSElapsedTime

* 1. Create new notebook in your environment (you should receive a personal link to environment)
  2. Implement required DQ checks
     1. ***For the final task your set of DQ checks are the next:  
        Carrier: Completeness by records  
        Airports: Consistency check for stateFlights: Consistency for CancellationCode  
        Flights: Consistency check for CRSElapsedTime***

Approach for implementing DQ checks:

Create and use share functions for similar DQ checks.

E.g. create a function to check PK uniqueness with the following arguments

* path to parquet file
* list of columns to be unique.

Call this function as many times as many tables we do require to pass this check.

Do the same for other checks if they are similar.

In case you have a unique DQ check, you can implement it in any way you’d like.

* 1. As a result, please print a table with DQ results. It should look like table below (slight changes acceptable)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Table | DQ check | Column | Status | Bad Data |
| 1 | Carrier | Uniqueness | Code | Failed | List of values that did not pass your check.  E.g. code= ‘ABC’ |
| … | … | … |  | … | … |

**Task\_2:**

1. Log into dlab using **EPAM SSO**

A screenshot of a computer

Description automatically generated

1. Go to List of resources and select required project

A screenshot of a computer

Description automatically generated

1. Click on Create New

A screenshot of a computer

Description automatically generated

1. Enter details for your environment, select SHARED IMAGE as shown on a screenshot below and click on create button. Note that shared image has already all required libraries installed. In case you find any gaps – pls contact us directly.

Give 1st letter of first name + last name as a Name (Volha Melnikava = vmelnikava)

A screenshot of a computer

Description automatically generated

1. Wait for a while until status of your environment becomes Running

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

1. Click on environment name and select Jupyter link to access it

A screenshot of a computer

Description automatically generated

1. Click on Jupyter link -> you will be redirected to Jupyter UI

A screenshot of a computer

Description automatically generated

1. Click on new -> terminal to open it

A screenshot of a computer

Description automatically generated

1. Name of bucket where all the data for the task is stored: i**skldl01-projectby-local-bucket**
2. You should be able to see 2 folders under the bucket:

* ‘source’ containing 3 source files required for processing data
* ‘raw’ –contains target data

A screen shot of a computer program

Description automatically generated

Files in ‘source’ and ‘raw’ folders:

A screenshot of a computer program

Description automatically generated

In the ‘raw’ folder:

A computer screen with white text

Description automatically generated

1. Copy ‘source’ and ‘raw’ folders from the common buckets to the local environment:

A screen shot of a computer program

Description automatically generated

1. On the Jupyter UI you can see data stored locally for dlab user. Default location is /home/dlab-user. This data can also be accessed through the terminal as well as through the UI.

A black screen with white text

Description automatically generated

A white and grey line

Description automatically generated with medium confidence

1. Tests according to the Check list for the Task\_2 were implemented.

DQ\_Checks\_okrangach.ipynb file in the attachment.

Tests results:

A screenshot of a computer

Description automatically generated