Data Engineer Case Study

On behalf of the growth and data team at Ounass, we would like to thank you for considering dedicating the time to complete this take-home exam.

The exam consists of 2 tasks:

- Building a batch data pipeline
- Data Analysis (Your own exploration)

You are expected to produce artifacts including but not limited to:

- Commented SQL scripts & python DAG's
- Notebook for Data Analysis
- A presentation with a tool like Powerpoint or Canva

Provided:

- Source Data
    - https://docs.google.com/spreadsheets/d/1P2rpXt0ZGn6B5FyRbq6wJXKJpc9HynCLvkOrKcnUmYY/edit?usp=sharing
- Docker compose file to setup airflow, mysql & postgresql
    - Above docker-compose is not configured to run on ARM processor, you are free to modify according to your system needs
    - Create network mysql_defult before running script
    - Volumes need to updated according to your workspace
    - Source.zip file consists of
        - docker-compose.yml for airflow infra
        - variables.json for creds

Please store all artifacts in a private github repository and share it with us. You can share it with the following email address: mokumar@altayer.com

We would also expect you to demonstrate and present the complete solutions to us

Task 1: Building a Batch Data Pipeline

In this task, you will design and build an ETL pipeline to satisfy a business requirement as described below.

your-luxury-goods.com has its sales data stored in a transactional database system. As a Data Engineer, you are asked to bring these data into a Data Warehouse for further analysis. Your goal is to build a data pipeline to satisfy this requirement by adhering to the best practices. It is also important to make sure that the data in the destination tables is in the best shape. The pipeline should be scheduled daily so that the warehouse stores data up to the previous date on any given date.

An important point to consider here is that the source database is a production OLTP system that uses MySQL. Hence, the extraction process must not put any unwanted stress on it.

Please find below the artifacts that you will use to build the pipeline:

1. The schemas for the source tables
2. An Entity Relationship Diagram (ERD) of the tables
3. A link to a google sheet that contains extracts from the tables

You will be working with 3 tables in the source system:

- Customer: a table that stores customer data
- Salesorder: a table that stores order data
- Salesorderitem: a table that stores item data at the order level

Tables Schemas

Table Name : customer

Column Name	Data Type	Rules	Description
id	integer	Unique, NOT NULL	Primary Key for thecustomer
first_name	string	NOT NULL	Customer's First Name
last_name	string	NOT NULL	Customer's Last Name
email	string	NOT NULL, shouldbe a valid email address	Customer's email address
gender	string	Either "Female" or "Male"	Customer's Gender
billing_address	string		Default Billing Address of the Customer
shipping_address	string	NOT NULL	Default Shipping Address of the Customer

Table Name : salesorder

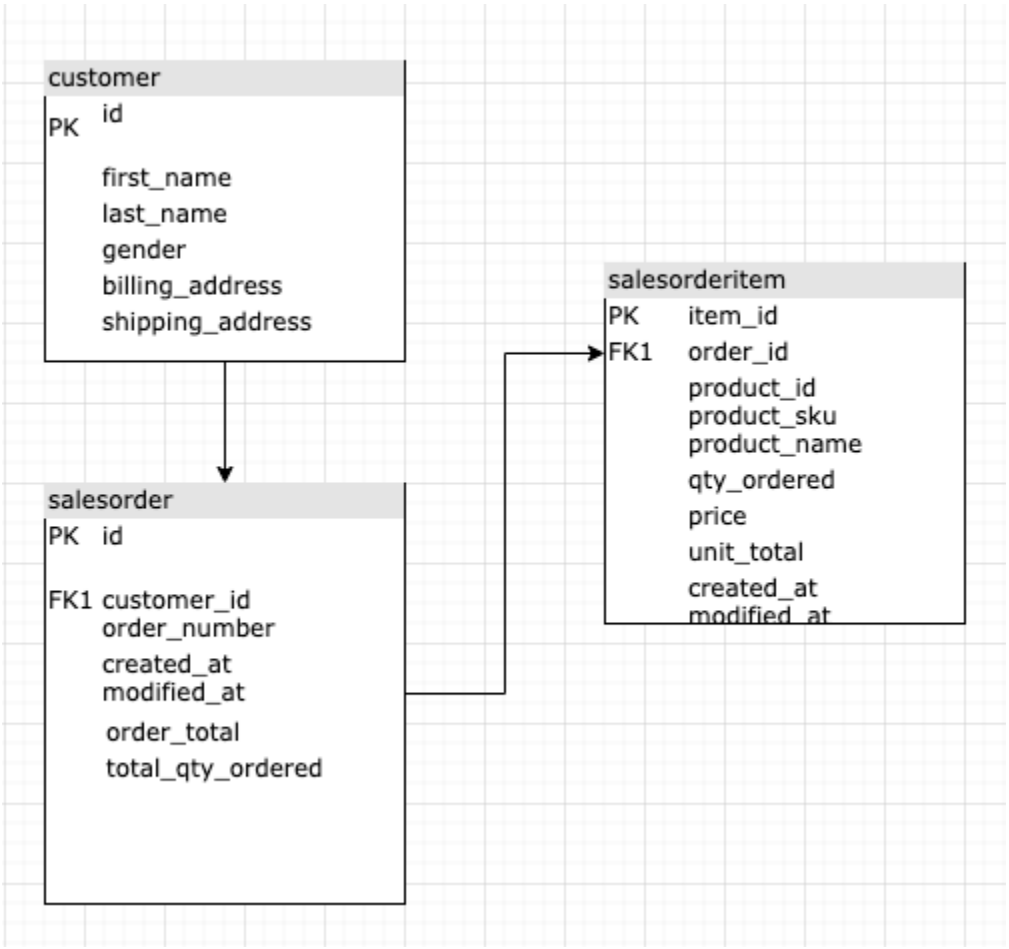
Column Name	Data Type	Rules	Description
id	integer	Unique, NOT NULL	Unique identifier of the Order record
customer_id	integer		Customer ID of the customer who placed the order
order_number	string	NOT NULL, UNIQUE	Unique ReadableOrder Number which is used by the business
created_at	datetime	NOT NULL. Valid DateTime	Created Time of theOrder
modified_at	datetime	NOT NULL. Valid DateTime	Last Modified time ofthe order
order_total	decimal	NOT NULL	Total Value of the Order
total_qty_ordered	integer	NOT NULL	Total No of Items in the Order

Table Name : salesorderitem

Column Name	Data Type	Rules	Description
item_id	integer	Unique, NOT NULL	Unique identifier forthe order item record
order_id	integer	NOT NULL	Related Order Record id slaesorder.id
product_id	integer	NOT NULL	Related Product's ID
product_sku	string	NOT NULL	Product's SKU, whichuse to identify the product uniquely by the business
product_name	string		Name of the Product
qty_ordered	integer	NOT NULL	Quantity Ordered
price	decimal	NOT NULL	Unit Price of the Product
line_total	decimal	NOT NULL	Qty_ordered * price
created_at	datetime	NOT NULL. Valid Datetime	Created Time of the Order Line Item Records
modified_at	datetime	NOT NULL. Valid Datetime	Last Modified time ofthe Line Item Record

Entity Relationship Diagram for the Source Data

The ER Diagram below describes the relationships of the three main entities related to this task.



Link to Data Extracts

The link below contains a google sheet of data extracts from the 3 tables described above.

https://docs.google.com/spreadsheets/d/12bkmPhzl6t3jqgBWEpLA4KroQVmdjpzLiKw6lutqk5o/edit?usp=sharing

Rubrics

1. Create the source OLTP system and the source tables on a MySQL database instance using the sample data set provided to you. Include the DDL scripts in your submission.
2. The schema for the destination Data warehouse table you need to populate is shown below. You can use it to create a data warehouse using Postgresql. Include the DDL scripts in your submission.

Table Name : sales_order_item_flat

Column Name	**Data Type**
item_id	int
order_id	int
order_number	string
order_created_at	datetime
order_total	double
total_qty_ordered	int
customer_id	int
customer_name	string
customer_gender	string
customer_email	string
product_id	int
product_sku	string
product_name	string
item_price	double

item_qty_order	int
item_unit_total	double

Using Apache Airflow, please author the following batch data pipelines to run daily:

3. A pipeline that pulls the data from the production database and stores it in a landing zone in the Postgresql warehouse.
4. A pipeline that pulls the data from the landing zone into an Operational Data Store where the data is cleansed, and the tables are normalized. Include the corresponding ERD in your submission.
5. A pipeline that pulls the data from the ODS into a Data Mart that has the destination table as described in the table above.

You are expected to perform below checks on the table:

a. For the final destination table, order_id and item_id should be unique across all the records
b. Date Time fields should contain date and time in the correct format
c. Double and Integers should not be null
d. Email should be checked for correct patterns
e. Bonus - Any other data validations you think that will help to have a quality dataset

We encourage you to adhere to the best development practices in your solution such as:

- Modularity in code structure that enhances re-use and extendibility
- Adhering to PEP-8 standards for Python and to the SQL formatting standards described in this link
- Using Pandas to handle data processing in the ETLs
- Including unit tests for your DAGs

Finally we would expect you to demonstrate and present the complete solution to us.