DATA ENGINEERING

BUSINESS CASE.

Name: Victor Daniel Velasco Ochoa

—---------------------------------------------------------------------------------------------------------------------------

Welcome to the Atrato challenge for **Data Engineers.** In this challenge, we are looking for ownership on

decisions, data, and business. Imagine you’re already a crucial part of the team and many businesses.

decisions rely on the data you provide.

● As a Data Engineer you should be able to resolve problems with **SQL** and **Python**.

The math scores of each student have been stored in the *STUDENT* table. Write a query to print the *ID* and the *NAME* of each of the three highest scoring students.

Print the *NAME*s in descending order by *SCORE*, then ascending order by *ID* for matching

*SCORE*s.

|  |  |  |
| --- | --- | --- |
| Table | | |
| Name | Type | Description |
| ID | Integer | A student ID in the inclusive range *[1, 1000]*. This field is the primary key. |
| NAME | String | A student's name. This field contains between *1* and *100* characters |
| Score | Float | The Math score of the student. |

|  |  |  |
| --- | --- | --- |
| **STUDENT** | | |
| **ID** | **NAME** | **SCORE** |
| 1 | Bob | 50 |
| 2 | John | 65.5 |
| 3 | Harry | 45 |
| 4 | Edgar | 85 |
| 5 | Dev | 25 |
| 6 | Sid | 98 |
| 7 | Tom | 90 |
| 8 | Julia | 70.5 |
| 9 | Erica | 81 |
| 10 | Jerry | 85 |

**Sample Output**

6 Sid

7 Tom

4 Edgar

—-------------

**EXPLANATION**

The students are arranged in the descending order of their math scores, followed by the ascending order of their ids, as shown below:

Sid > Tom > Edgar > Jerry > Erica > Julia > John > Bob > Harry > Dev

Edgar’s and Jerry’s scores were the same, so they are shown in ID order.

**Can you build a query that returns the expected result?**

First, we need to create the table with the mentioned constraints, for this I will use MySQL executed locally, I create a new Database for this test, and I will use it:

CREATE DATABASE atratoTest;

USE DATABASE atratoTest;

CREATE TABLE Student(

ID INTEGER PRIMARY KEY,

NAME VARCHAR(100) NOT NULL,

SCORE DECIMAL(3,1) NOT NULL,

CONSTRAINT range\_id CHECK(ID BETWEEN 1 AND 1000));

INSERT INTO Student VALUES (1,'Bob',50),

(2,'John',65.5),

(3,'Harry',45),

(4,'Edgar',85),

(5,'Dev',25),

(6,'Sid',98),

(7,'Tom',90),

(8,'Julia',70.5),

(9,'Erica',81),

(10,'Jerry',85);

Now we can Write a query to print the *ID* and the *NAME* of each of the three highest scoring students and print the *NAME*s in descending order by *SCORE*, then ascending order by *ID* for matching

*SCORE*s

R=

**SELECT ID, NAME FROM Student ORDER BY Score DESC, ID ASC LIMIT 3;**

Una captura de pantalla de una computadora

Descripción generada automáticamente

We launched atrato pago to the market. As you can imagine the data is growing and changing fast. Some tables that were put together by skilled data scientists and risk analytics are no longer working properly and are breaking due to data inconsistency.

**How would you propose to keep track of the tables that are breaking in order to provide you**

**teammates the latest information?**

**R=**We can implement many strategies for that but to iteratively solving the main issue I will suggest the next plan:

To mitigate the current issue, we need to establish a system to monitor the logs and events at each stage of the data flow process. If we identify errors or a decline in data quality, we can display them through a user interface (UI) to trigger alerts and understand where and, more importantly, why inconsistencies are occurring. This approach will assist in identifying the tables that are causing most of the issues and determine which ones require investigation or redesign to integrate the data inconsistencies (also, to ascertain if data quality is compromised, necessitating cautious data extraction). Additionally, we can integrate small dashboards with database statistics, such as latency, CPU usage, and pending messages, using tools like Grafana to continuously track potential issues or congestion in the data flow that could lead to errors during data writes and further data inconsistencies.

By leveraging these tools, we can identify tables with issues and attempt to rectify them in case of exceptions. If rectification isn't feasible, we can determine when the most recent and accurate data was written to facilitate extraction, also we can create views from the tables and with a certain rule of validation, we can know if some part of the data has the required quality.

Another approach to enhance data consistency is to establish processes that monitor data movement. For instance, identifying if any of the pipelines are causing congestion and understanding the reasons behind it. Increasing the number of transformations between each step and optimizing them to prioritize data integrity can be beneficial.

Moreover, creating unit tests for pipelines and transformations during the development phase can help identify errors that arise during loading and resolve them before implementation and production.

Ultimately, to minimize these errors, we must refactor and redesign tables. This involves adopting a standardized approach to naming columns, implementing error controls during writing using constraints, optimizing queries for data manipulation, and establishing proper relationships between tables to reduce data redundancy. Error handling mechanisms should also be in place to manage duplicate data and null values. Furthermore, it's essential to implement different databases based on business needs while considering the CAP theorem. For swift data retrieval, NoSQL databases are suitable; for controlled data entries, SQL databases are appropriate. In the case of ingesting substantial data volumes, distributed systems with fail-tolerant mechanisms should be employed.

● It’s a common practice to have many systems scattered all over, imagine we have four:

○ The first one might be hosting the app (ignore the database architecture for now)

■ Every time the user interacts with a screen, clicks a button, or opens the app, this is stored as an event.

○ The second one hosts the risk model for daily operations.

■ Every time a customer asks for a credit, the system retrieves the risk data from the credit bureau and evaluates whether the customer is prone to be a defaulter.

○ The third one host the customer information

■ This database hosts name, email, address, etc.

○ Finally, the fourth and last system hosts all payment information.

■ This database hosts information related to the usage of the credit, payments, recurrent payments, credit line, etc.

All above information is crucial for data exploitation.

**What should we do to centralize the data to display it in charts for KPI monitoring?**

**R=**A good solution can be implementing a data warehouse for centralize and integrate all the data of all the different systems in one place for unique access of information, we can use Snowflake that is a cloud platform for storage and management of big data

Now we need to create ETL for implement systems for extract, transform and load data un some format that should be standardized for scalability pursues, for doing that we can use tools like Apache airflow, PySpark, Hadoop, spark or even basic libraries for data manipulation like pandas or NumPy depending of the volume of data, for that we need some unified model data that tell us how integrate all the different sources of data in one that can be used for KPI depending of the business.

For a better understanding and insight into the data transformed we need to use some visualization tools for create our own dashboard based on the business, we can use tools like matplotlib, seaborn of python or more sophisticated tools like power bi for get the KPI of the business based in the data.

**What would you propose the data governance strategy should be?**

**R**= To establish procedures that guarantee data quality and integrity, it's essential to delve into the specifics of each business. However, some principles can guide us, such as defining metrics to evaluate data quality. This includes determining the acceptable number of rows with special characters, as well as metrics for assessing null values, zero or invalid values, outliers, and strategies for handling them.

Another crucial aspect concerns data anonymity. It's imperative to remove sensitive information and outline a process for data extraction. Additionally, detailing how data will be destroyed once it's no longer required or specifying its storage duration is vital. Encryption, if applicable, should be employed when data is transmitted using specific protocols. Defining access control procedures for the production team and others is a pivotal component of data governance.

Ultimately, documenting each implementation standardization across new features is essential. Establishing well-defined roles within the team can enhance business scalability for the future while mitigating the risk of information loss in case a team member departs.

● Design and implement a data pipeline using Apache Airflow and Snowflake/Redshift.

You can make use of any public available dataset. Write about how you did it, which challenges you encountered, and how you managed to tackle such challenges.

For this kind of integration many challenges were faced, in the beginning it was the first time that I made this kind of integration between Snowflake and Airflow, so I need to investigate for what they were used for and how use it in basic.

1. First I understand the basic architecture of Snowflake and I learn that this is used as a database scalable warehouse, and airflow is used as a DAG scheduler for control all the transformations and steps, so in order to solve it first we install all the dependencies like a linux shell for run airflow, an account for use Snowflake services, for the bash I decided to use zbh, with Astro CLI for control all the commands in Airflow, from this I faced some challenges trying to install an configure all the tools I tried first use first a virtual environment with vsc but it is not allowed to run the airflow server using a windows shell, that’s the reason I investigate and find that using Astro is more simple start the server of airflow.
2. Once I got all configured, I design some simple pipeline for the exercise, following some of the database that you mentioned I will create two databases in Snowflake one **Customer** table is used for save all the basic information of the user register in atrato with one important column “credit” that represents the credit given by atratopago for purchase in eCommerce’s,

|  |  |
| --- | --- |
| Customer | |
| Cid | Integer(4) PK |
| Name | VARCHAR(10) |
| Age | Integer(2) |
| Credit | FLOAT(5,2) |

1. I will simulate an input of data with a csv file **Transaction** for all the purchases made in some eCommerce that uses the service of atratopago this file will display the transactions made by all the customers distinguishing in the TransId, the Customer Id Cid the date of transaction and one important value the amount of the transaction.

|  |  |
| --- | --- |
| Transaction | |
| TransId | Integer(4) |
| Cid | Integer(4) |
| TransDate | Integer(2) |
| Commerce | VARCHAR(30) |
| Amount | FLOAT(5,2) |

1. Now the requirement given by the ML team is detect possible abnormal purchases in the commerce and in the future make some alert to the clients, so for that the ML team wants to know the information of the customers that have make a transaction with an amount more than 85% of the total credit available and in which ecommerce was made it and save it in a temporary table called “Alert” located in our data warehouse in snowflake after pipeline execution.

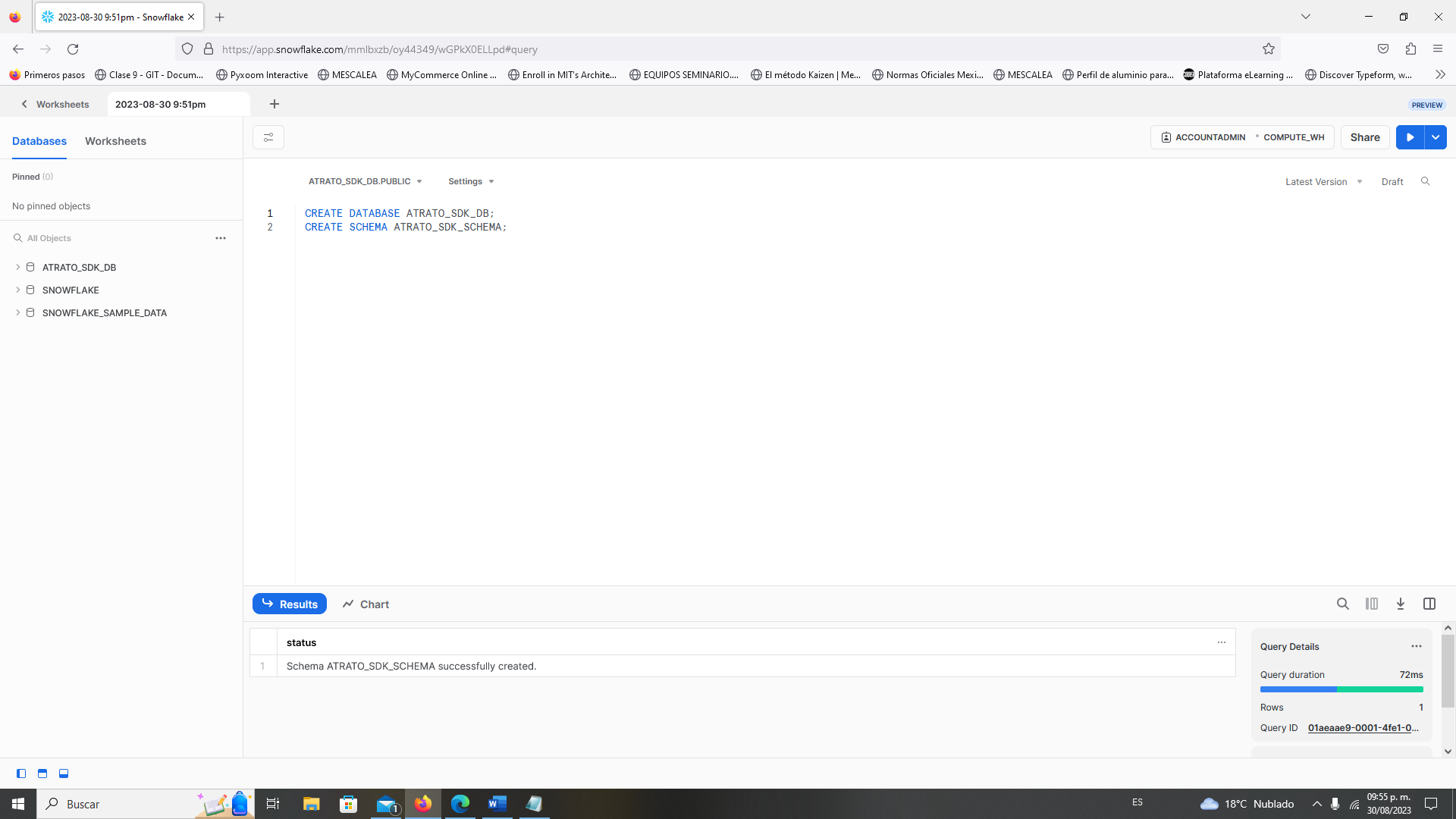
|  |  |
| --- | --- |
| Alert | |
| Cid | Integer(4) |
| Name | VARCHAR(10) |
| Age | Integer(2) |
| TransDate | DATE(dd/mm/yy) |
| Commerce | VARCHAR(30) |
| CreditUsed | FLOAT(5,2) |

1. Now I will configure the Snowflake warehouse for that we will execute the next query’s sets, one challenge here is know how to stablish the Datawarehouse, I search and find a tutorial online for that:

CREATE WAREHOUSE ATRATO\_SDK\_D;

CREATE DATABASE ATRATO\_SDK\_DB;

CREATE SCHEMA ATRATO\_SDK\_SCHEMA;



1. We create the two tables Customer and Alert Table (Extract and Load report) one challenge here is to know what the datatypes are used in Snowflake for that I search in the documentation:

CREATE OR REPLACE TABLE Customer (Cid NUMBER(4), Name VARCHAR(100), Age NUMBER(2), Credit DECIMAL(10,2));

CREATE OR REPLACE TABLE Transaction (TransId NUMBER(4),Cid NUMBER(4), TransDate DATE,Commerce VARCHAR(30),Amount DECIMAL(8,2));

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. I will next insert some example values in the customer table and Transaction table in this case I will insert the next ones.

INSERT INTO Customer (Cid, Name, Age, Credit) VALUES (1,'Victor Velasco',25,10000), (2,'Pedro Perez',60,50000),(3,'Valeria Quiroz',26,5000),(4,'Arturo Hernandez',30,15000.50);

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

INSERT INTO Transaction (TransId,Cid,TransDate,Commerce,Amount) VALUES (1,1,'1/1/2023','Gaia',100),

(2,3,'1/1/2023','Luuna',1000),

(3,1,'2/1/2023','Liv',5000),

(4,1,'2/1/2023','Liv',3500),

(5,2,'3/1/2023','Santandreu',400),

(6,2,'4/12/2023','Luuna',3000),

(7,4,'4/12/2023','dentalia',4500),

(8,3,'5/4/2023','Liv',5000),

(9,4,'5/4/2023','JPmusical',2000),

(10,3,'7/16/2023','dentalia',100);

1. Then I need the url for connect using Snowflake with Airflow so for that I reach the documentation and find that this option is available in one part of the UI of Snowflake:

Tabla

Descripción generada automáticamente

1. Next step is configuring the Airflow, that was one of the major challenges since was the first time using Airflow, I need to search in different areas I found some information for deploying it using VSC and AstroCLI with Linux bash, so I open a new terminal inside the project in VSC and then I create the Airflow project:

astro dev init

Also, we modify the requirements.txt file with the packages needed for snowflake

astro-sdk-python[snowflake]>=1.1.0

Captura de pantalla de computadora

Descripción generada automáticamente

We also make the same for the environment variables for interact with the schema in SQL.

AIRFLOW\_CORE\_ENABL\_XCOM\_PICKLING=True

AIRFLOW\_ATRATO\_SDK\_SQL\_SCHEMA=ATRATO\_SDK\_SCHEMA

Now I executed the command Astro dev start for start the server of airflow, but I got an error since this procedure needs to docker composer running for create a container of this application, so to install docker in the Linux system I execute the next command.

*curl -fsSL https://get.docker.com -o install-docker.sh*

*chmod u+x install-docker.sh*

*sudo ./install-docker.sh*

*sudo usermod -aG docker $USER*

*#for check if docker is running*

sudo systemctl status docker

#for start docker

sudo systemctl start docker

#start airflow webserver

astro dev start

1. Now Airflow is online, once we log in:

Captura de pantalla de computadora

Descripción generada automáticamente

1. Next is to connect the corresponding Snowflake warehouse with our airflow session for that I need to search for connect Snowflake with Airflow and it is very simple just need to go Admin Connections and input the information given by Snowflake in the step 8.

INSERT INTO Transaction (TransId,Cid,TransDate,Commerce,Amount) VALUES (1,1,'1/1/2023','Gaia',100),

(2,3,'1/1/2023','Luuna',1000),

(3,1,'2/1/2023','Liv',5000),

(4,1,'2/1/2023','Liv',3500),

(5,2,'3/1/2023','Santandreu',400),

(6,2,'4/12/2023','Luuna',3000),

(7,4,'4/12/2023','dentalia',4500),

(8,3,'5/4/2023','Liv',5000),

(9,4,'5/4/2023','JPmusical',2000),

(10,3,'7/16/2023','dentalia',100);Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Now I will create all the connections and transformation in a custom DAG for that I basically load the data from transaction then I execute a grouping data based in the same date and sum the amount for know the amount given in the same day and then I join it to the customer table for filter the data with abnormal transaction in the day and save it in Snowflake in a temporary table, one of the biggest challenges here is to connect all the environment with snowflake the documentation and information available is deprecated and was a real challenge get the connection.
2. I built and test the final data pipeline:

Una captura de pantalla de una computadora

Descripción generada automáticamente

1. And then we can check in Snowflake for the final data that correspond to possible alerts of fraud for ML team:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. In instance all the project was full of challenges, the most difficult part was the integration with snowflake how to know use airflow since it has many api and sdk was not very clear how to start but at the end I react documentation ad when the code fails I used the logs section in airflow for debugging and find the solution, if you want to see the code please enter next link pipeline\_airflow\_snowflawe.