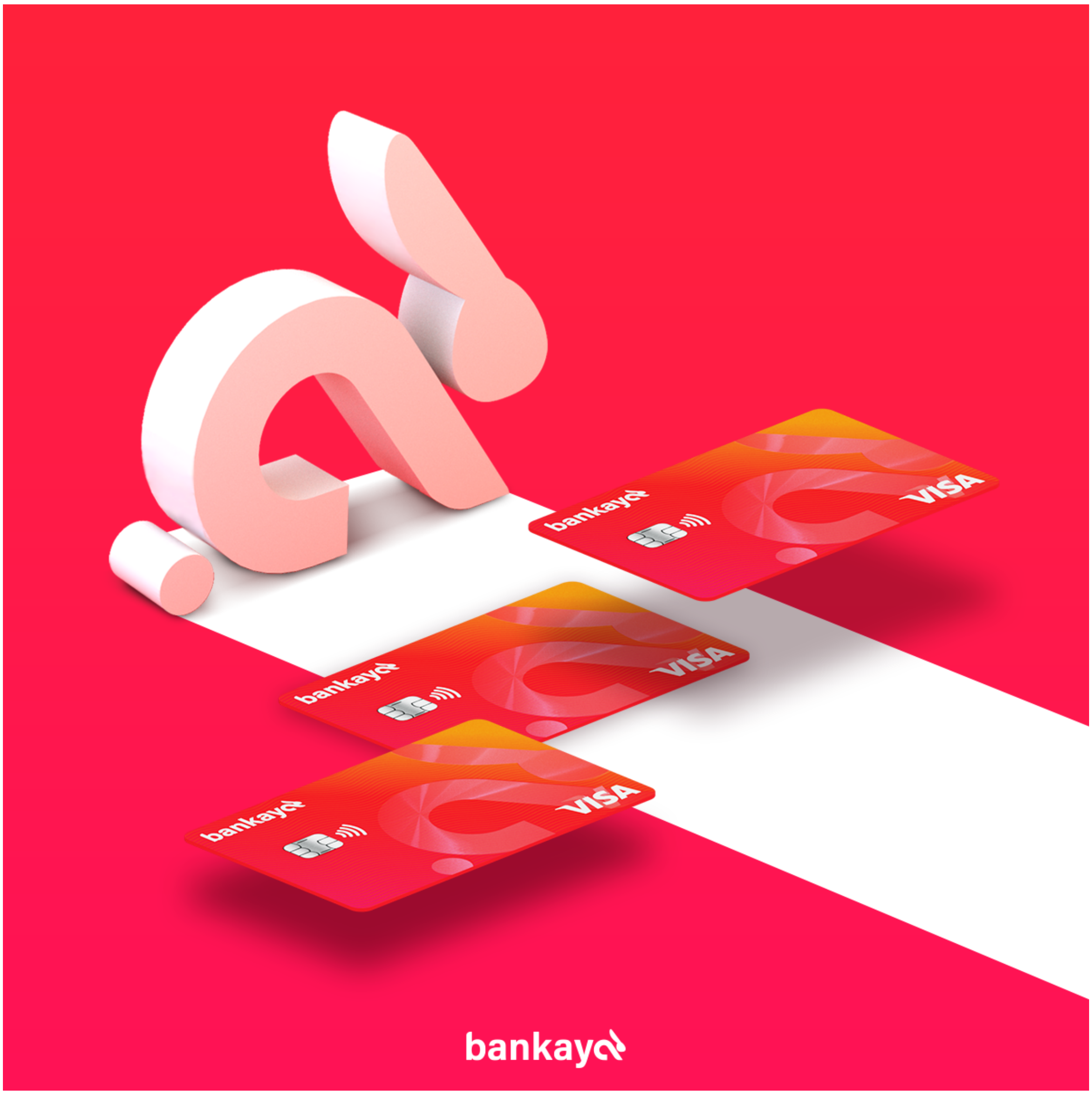


Data Engineer Test

Canseco García Edgar Jesús

. Julio de 2021

caption float

What is data Engineering?

We are experiencing a new technological transformation, now more than ever in history, thousands and thousands of data are produced per second, and with the arrival of new ways of creating infrastructure, such as cloud computing, it is no longer crazy to store large amounts of information, now data has become a fundamental asset in the growth and operation of the vast majority of companies, and this trend is only growing with new technologies and ways of finding value and knowledge in data, as in the case of the use of interactive dashboards or deep learning models.

The problem arises that the data comes in different forms and from very different sources, the speed with which it arrives and the memory size that it will occupy when stored also varies, in addition, this data is not ready to be used for a report or ML model, this data must be cleaned, unified, transformed, etc, and finally put them somewhere where they can be available so that they can be used. This is where data engineering comes in. It is the discipline in charge of all this process described, in addition to having responsibilities such as data management, data infrastructure, and data security.

What are the main responsabilities of a data Engineer?

The main responsibilities of a data engineer could be divided into two branches, technical and communication responsibilities. The technical responsibilities would be focused on the construction of data pipelines for the different challenges that exist today, in addition to security, infrastructure, and data management aspects that must be known for better data quality.

Due to these processes impact various areas in an organization, it is important to maintain clear communication about the various characteristics that these data pipelines must have to improve their quality.

Explain ETL?

ETL is the acronym for Extract, Transform and Load Data, it is the process of compiling data from a diverse number of sources, its subsequent transformation, and organization to finally be stored in a repository.

In more detail on this process, the first part (E) extracts the data from the source systems, and this task causes minimal impact on the system. It is important to take into account the different formats in which the data could appear, and define whether it is a real-time or batch ingest. The second part (T) applies a series of business rules or functions to the extracted data to turn it into useful and meaningful data. In the third part (L) the data from the previous phase is loaded into the destination system. Usually, this repository is a Data Lake or Data Warehouse.

Also in the ETL, you can define the periodicity with which the data can be loaded, for example, there will be data that is required to be loaded every week or data that is very transactional and it is necessary to apply the ETL process every certain amount of hours.

How you build a Data Pipeline?

In the projects I have worked on, it is essential to begin to listen and understand the client and thus have a better knowledge about their line of business to make a work plan that specifies how the architecture solution to be implemented will be. This work plan also defines the processes and tasks that describe the Data Pipeline, which includes how it will be executed, schedule its execution tasks, monitor the Pipeline, and make quality tests verifying the output of the Pipeline.

Once the work plan and its scope are completed, the tasks are scheduled and divided among the different members of the data team, to begin with, the planned construction of the Data Pipeline.

In a RDBMS Joins are your friends. Explain Why

By definition on RDBMS, the data is structured in tables, where each table covers a concept or definition of the business, but if analysis or a report of the data is required, it will be necessary to work with different tables, and this split data is not very useful.

This is how joins help us in this part of data union, where selecting the correct type of join (inner, left, right, outer) and the correct union filters, depending on the use case, helps to unify that information from different tables answering various questions about the business.

What are the main features of a production pipeline.

The production pipeline must be safe and stable mainly, in addition to having the good practices of the DevOps culture and having an environment of continuous integration and continuous delivery. They must be monitored and configured with metrics to quickly know when something is not working correctly or there are anomalies, in addition to having mapped how failures will be handled and having the necessary logs for analysis.

It must also have a reliable security infrastructure that does not jeopardize access to the information it handles and have an estimate of how long it takes to run with different workloads and thus be able to make it scalable or apply autoscaling.

How do you monitor this data pipelines?

There are cloud services that are linked to the different services they offer to be directly monitored, such as Stackdriver, in addition to storing the logs (normally for 30 days) to be analyzed. To correctly use these services, metrics are configured to monitor and configure alarms, for example, a metric for the use of a VM CPU, and establish a threshold of use of 70 %, and if the VM exceeds that capacity of use, send a mail to administrator and data engineer.

Due the logs are also stored, these can be analyzed to find useful insights and complement the information that the metrics provide, for example, what was the workload that caused the CPU to increase up to 80 %?.

Give us a situation where you decide to use a NoSQL database instead of a relational database. Why did you do so?

The first time I opted for a NoSQL database was in college. The prototype of a taxi services app had to be made, the way it was defined implementation logic and how to store the different reservation records with their respective information, it was very complicated to want to model it in a relational database because the data did not have a defined schema and the normalizations were not clear, so a better approach was a non-relational database and save the JSON files as documents. Additionally, the way the data was retrieving was easy to consume.

What are the non technical soft skills that are mos valuable for data engineers?

One of them is the ability to communicate, since it not only depends on speaking appropriately, it is the ability to break down technical problems and explain them correctly and understandably to other people who often do not have that technical background.

Have the ability to work as a team and based on defined work methodologies, for a correct harmony with all the people involved in the project or job.

Diagram of the solution on GCP Infrastructure

Supose you have to design an Anomaly Detection Solution for a client in real or near real time.

A platform for anomaly detection is about finding patterns of interest (outliers, exceptions, peculiarities, etc.) that deviate from expected behavior within dataset(s). Given this definition, it's worth noting that anomaly detection is, therefore, very similar to noise removal and novelty detection. Though patterns detected with anomaly detection are actually of interest, noise detection can be slightly different because the sole purpose of detection is removing those anomalies - or noise - from data. Which technologies do you apply for real time ingestion and stream for an anomaly detection system? Diagram the solution in AWS or GCP Infrastructure.

Solution: The architecture suggested in the diagram assumes that the data can come from on-premises sources, or the infrastructure could be in the cloud and would be deployed in a PaaS service such as App Engine or a VM or Vm´s fleets orchestrated with Kubernetes for example.

For the part of ingesting in real-time or near real-time, the Pub / Sub service is used, which is capable of ingesting millions of data with a latency of milliseconds, this is managed with a topic to which multiple instances of the system can send their messages. Cloud Dataflow integrates natively with Pub / Sub through a subscription, in addition, it contains templates for this purpose that can be adjusted if you want to do this part quickly.

The PCollection structures in Dataflow allow reading this data and making transformations distributed with the Apache Beam framework with python, so it may be able to detect anomalies and save the results in a required format to deposit them in a bucket on Cloud Storage or transfer it to a DW handled by BigQuery for analysis. There is also the possibility that for this analysis of the anomalies detected work with python and one of its many libraries, so an instance of Jupyter Lab with Datalab is suggested.

Finally, these anomalies may be of interest to developers and that is presented in a system, in the form of a report or a future step, have an interest in making ML models and predicting scenarios, so the architecture must be capable of scaling to new services.

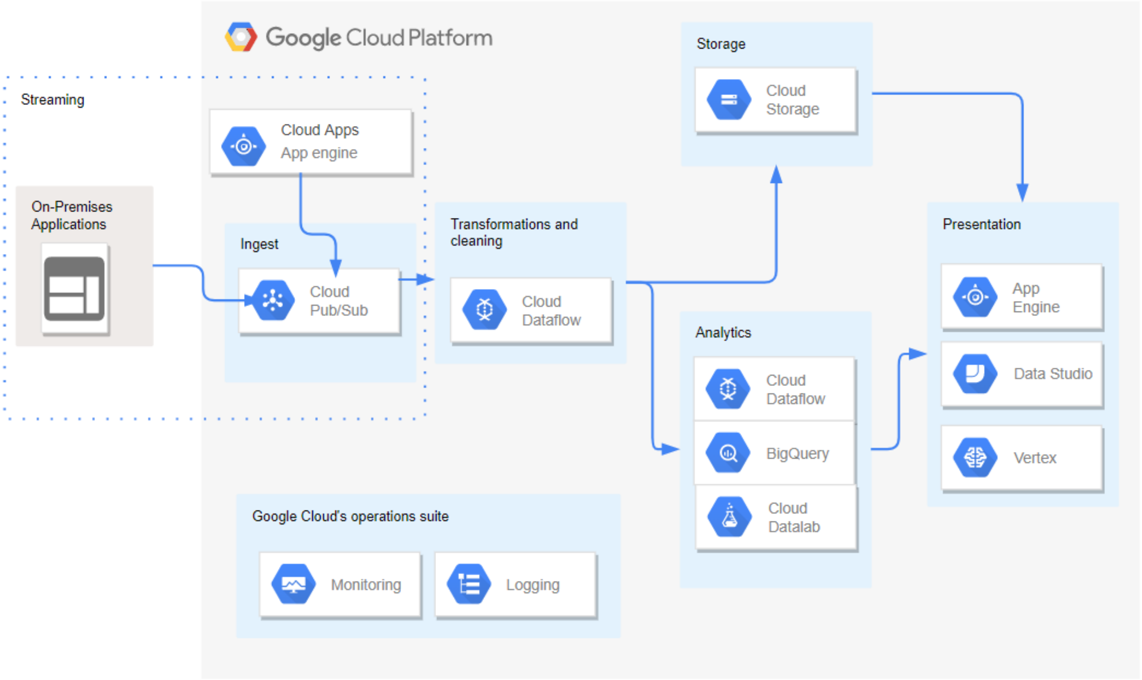


Figura 1- Architecture on GCP

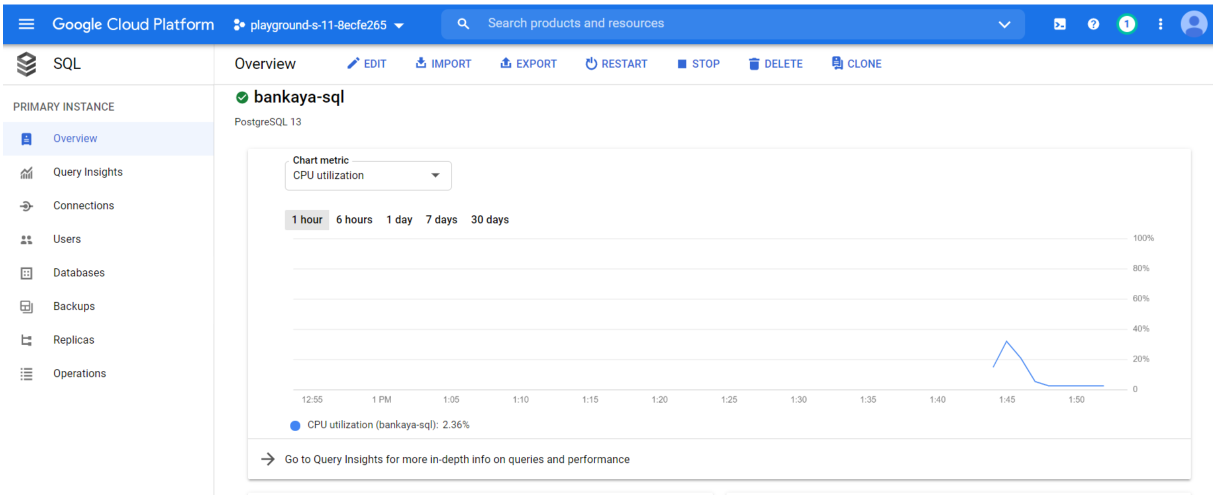
Differences between OLAP and OLTP Systems.

In its essence, OLTP is a database system oriented to transaction processing, so access to data is optimized for frequent reading and writing tasks since its transactionality is high and queries are not usually complex, while OLAP systems are databases oriented to analytical processing, access to the data is usually read-only. The most common action is retrieving data, with additions and joins to consult historical information and obtain useful information in its analysis.

Create the following tables in a SQL Database

For the challenges, a temporary test laboratory instance was created to do the challenge.

A project on GCP was used to configure a Cloud SQL instance with postgresql.



bankaya-sql

PostgreSQL 13

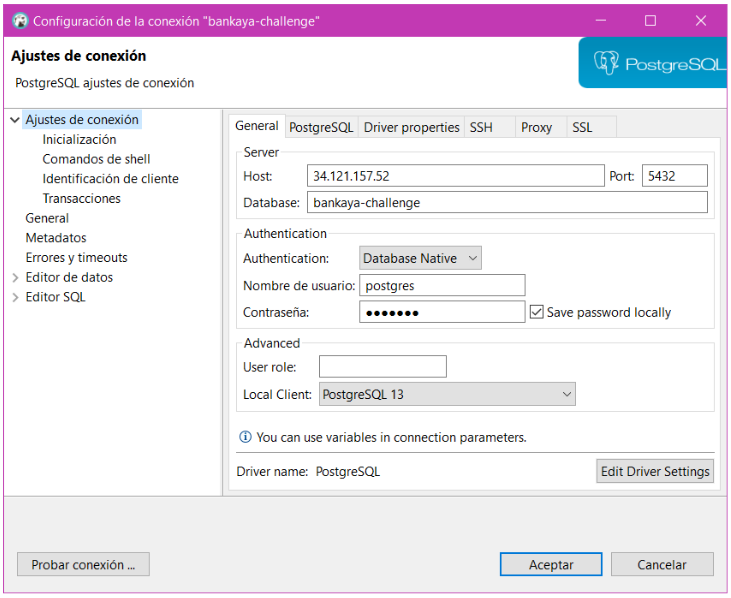
Chart metric: CPU utilization

CPU utilization (bankaya-sql): 2.36%

Go to Query Insights for more in-depth info on queries and performance

Figura 2- Cloud SQL

It was configured to be able to work in Dbeaver.



Ajustes de conexión

PostgreSQL ajustes de conexión

Server

Host: 34.121.157.52 Port: 5432

Database: bankaya-challenge

Authentication

Authentication: Database Native

Nombre de usuario: postgres

Contraseña: ••••••• ☑ Save password locally

Advanced

User role:

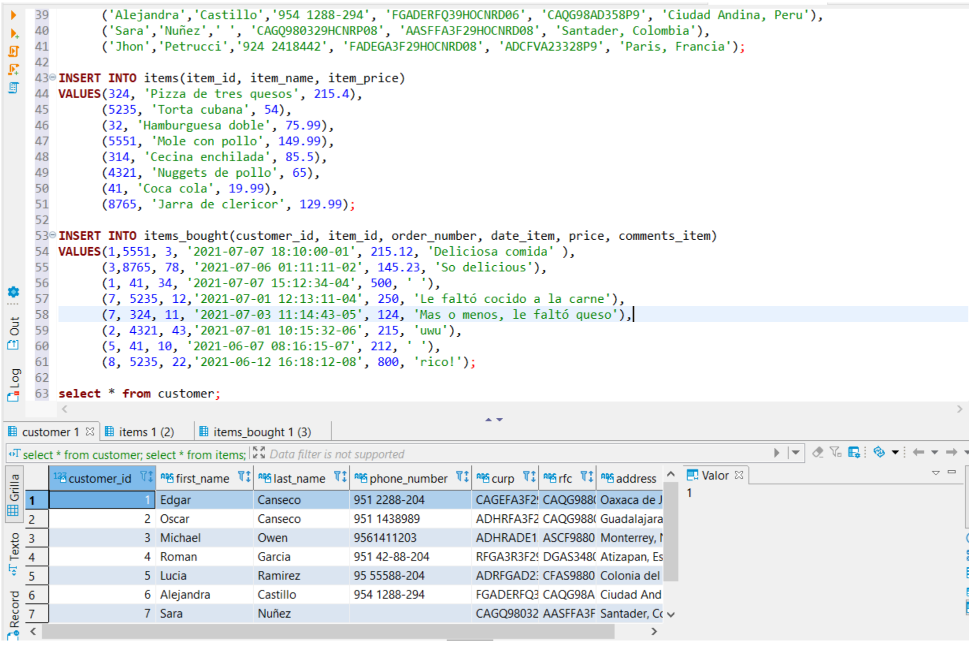
Local Client: PostgreSQL 13

You can use variables in connection parameters.

Driver name: PostgreSQL

Figura 3- Dbeaver configured

Finally the tables were created and the data was inserted.



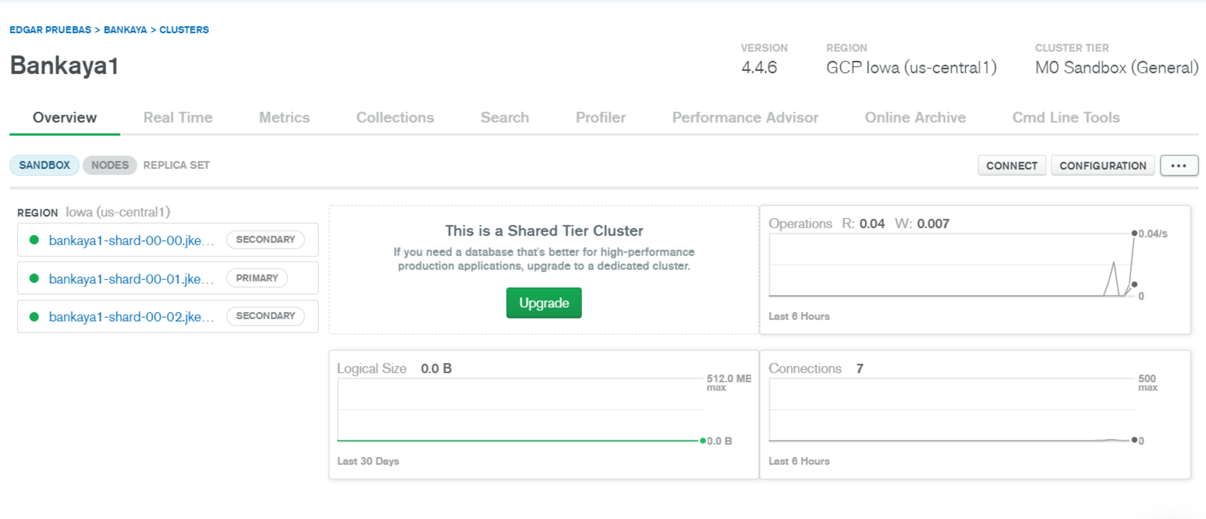
```
        ('Alejandra','Castillo','954 1288-294', 'FGADERFQ39HOCNRD06', 'CAQG98AD358P9', 'Ciudad Andina, Peru'),
        ('Sara','Nuñez',' ', 'CAGQ980329HCNRP08', 'AASFFA3F29HOCNRD08', 'Santader, Colombia'),
        ('Jhon','Petrucci','924 2418442', 'FADEGA3F29HOCNRD08', 'ADCFVA23328P9', 'Paris, Francia');

INSERT INTO items(item_id, item_name, item_price)
VALUES(324, 'Pizza de tres quesos', 215.4),
        (5235, 'Torta cubana', 54),
        (32, 'Hamburguesa doble', 75.99),
        (5551, 'Mole con pollo', 149.99),
        (314, 'Cecina enchilada', 85.5),
        (4321, 'Nuggets de pollo', 65),
        (41, 'Coca cola', 19.99),
        (8765, 'Jarra de clericor', 129.99);

INSERT INTO items_bought(customer_id, item_id, order_number, date_item, price, comments_item)
VALUES(1,5551, 3, '2021-07-07 18:10:00-01', 215.12, 'Deliciosa comida' ),
        (3,8765, 78, '2021-07-06 01:11:11-02', 145.23, 'So delicious'),
        (1, 41, 34, '2021-07-07 15:12:34-04', 500, ' '),
        (7, 5235, 12,'2021-07-01 12:13:11-04', 250, 'Le faltó cocido a la carne'),
        (7, 324, 11, '2021-07-03 11:14:43-05', 124, 'Mas o menos, le faltó queso'),
        (2, 4321, 43,'2021-07-01 10:15:32-06', 215, 'uwu'),
        (5, 41, 10, '2021-06-07 08:16:15-07', 212, ' '),
        (8, 5235, 22,'2021-06-12 16:18:12-08', 800, 'rico!');

select * from customer;
```

select * from customer; select * from items; Data filter is not supported

	customer_id	first_name	last_name	phone_number	curp	rfc	address
1	1	Edgar	Canseco	951 2288-204	CAGEFA3F2	CAQG988	Oaxaca de J
2	2	Oscar	Canseco	951 1438989	ADHRFA3F2	CAQG988	Guadalajara
3	3	Michael	Owen	9561411203	ADHRADE1	ASCF9880	Monterrey, I
4	4	Roman	Garcia	951 42-88-204	RFGA3R3F2	DGAS348	Atizapan, Es
5	5	Lucia	Ramirez	95 55588-204	ADRFGAD2	CFAS9880	Colonia del
6	6	Alejandra	Castillo	954 1288-294	FGADERFQ3	CAQG98A	Ciudad And
7	7	Sara	Nuñez		CAGQ98032	AASFFA3F	Santader, Co

Figura 4- Data inserted

Now, for the NoSQL database I used mongodb, using the Atlas service configured on a GCP instance



EDGAR PRUEBAS > BANKAYA > CLUSTERS

Bankaya1

VERSION 4.4.6

REGION GCP Iowa (us-central1)

CLUSTER TIER M0 Sandbox (General)

REGION Iowa (us-central1)

- bankaya1-shard-00-00.jke... SECONDARY
- bankaya1-shard-00-01.jke... PRIMARY
- bankaya1-shard-00-02.jke... SECONDARY

This is a Shared Tier Cluster

If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.

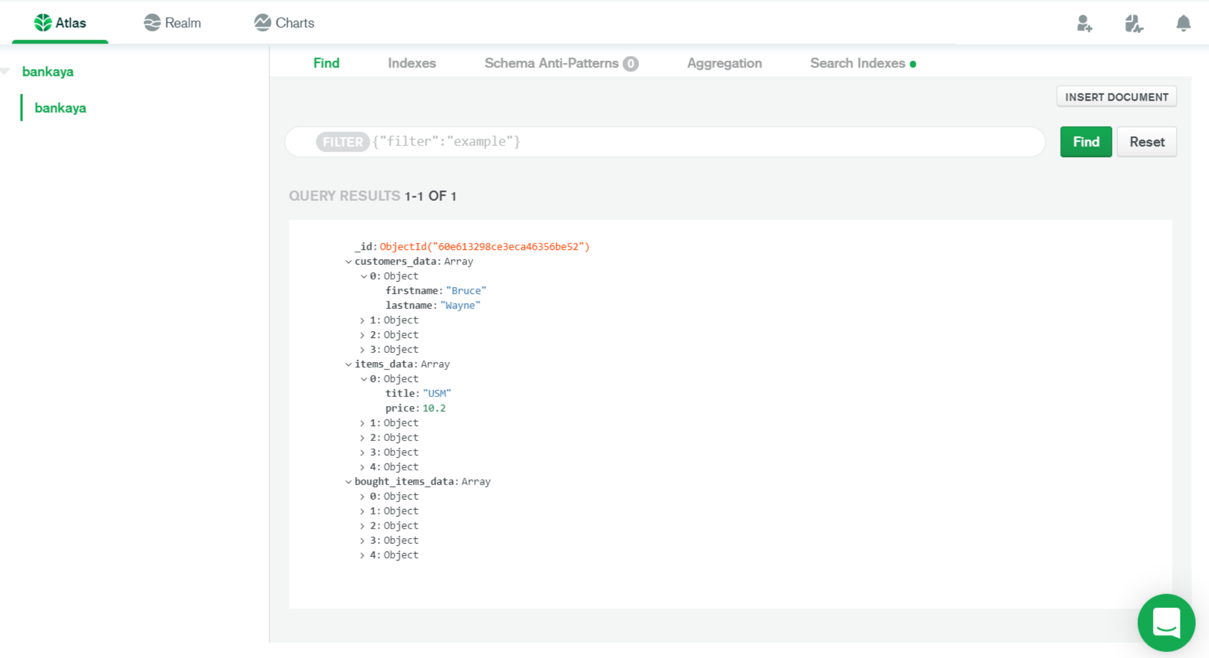
Operations R: 0.04 W: 0.007

Logical Size 0.0 B

Connections 7

Figura 5- MongoDB cluster

The data was inserted into a document as proposed in the challenge.



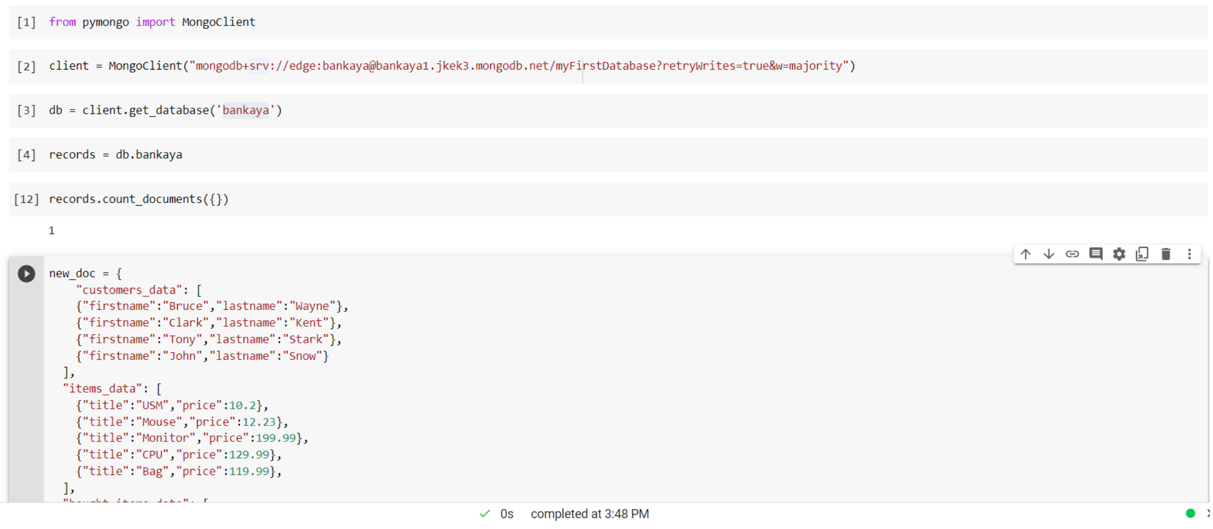
{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60e613298ce3eca46356be52")
⌄ customers_data: Array
  ⌄ 0: Object
      firstname: "Bruce"
      lastname: "Wayne"
  › 1: Object
  › 2: Object
  › 3: Object
⌄ items_data: Array
  ⌄ 0: Object
      title: "USM"
      price: 10.2
  › 1: Object
  › 2: Object
  › 3: Object
  › 4: Object
⌄ bought_items_data: Array
  › 0: Object
  › 1: Object
  › 2: Object
  › 3: Object
  › 4: Object
```

Figura 6- MongoDB data

And this part was done through a python script



```
[1]  from pymongo import MongoClient

[2]  client = MongoClient("mongodb+srv://edge:bankaya@bankaya1.jkek3.mongodb.net/myFirstDatabase?retryWrites=true&w=majority")

[3]  db = client.get_database('bankaya')

[4]  records = db.bankaya

[12] records.count_documents({})

     1

     new_doc = {
         "customers_data": [
         {"firstname":"Bruce","lastname":"Wayne"},
         {"firstname":"Clark","lastname":"Kent"},
         {"firstname":"Tony","lastname":"Stark"},
         {"firstname":"John","lastname":"Snow"}
       ],
       "items_data": [
         {"title":"USM","price":10.2},
         {"title":"Mouse","price":12.23},
         {"title":"Monitor","price":199.99},
         {"title":"CPU","price":129.99},
         {"title":"Bag","price":119.99},
       ],
       "bought_items_data": [
```

Figura 7- script in python

Create a ETL Script pipeline

Dataprep is an interesting service that fits very well with the proposed task, it is capable of extracting various sources of information, combining them into elements called -ecipe", cleaning with a very complete suite of tools, selecting different loading destinations, among They BigQuery, and above all, it is a managed and scalable service, since in the end all the transformation flows are transformed into a dataflow pipeline, to distribute the workload, and finally, it can schedule the frequency of execution of the flows, making a merge of the records, to only update the most current data once the history is loaded.

The first part consists of adding the data sources, which in this case is Mongo and Postgresql database.

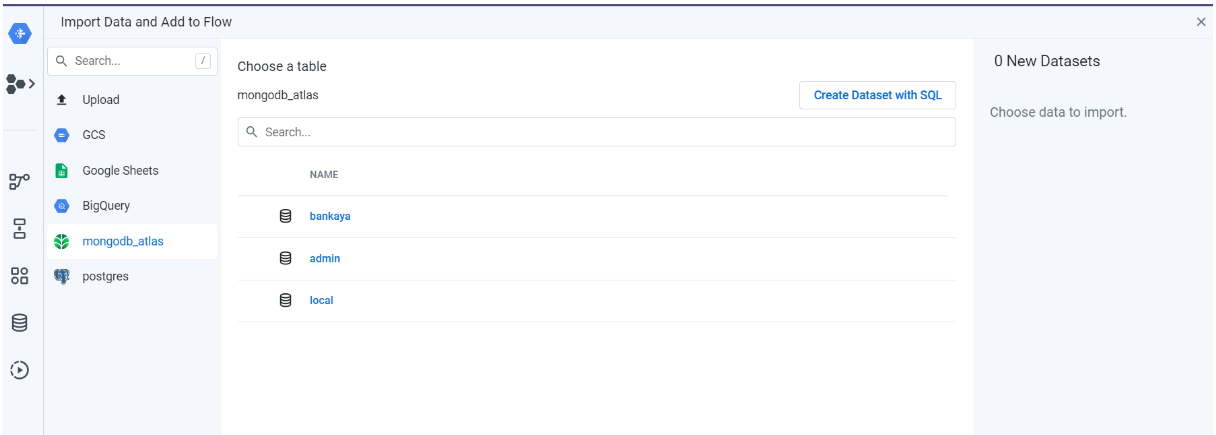
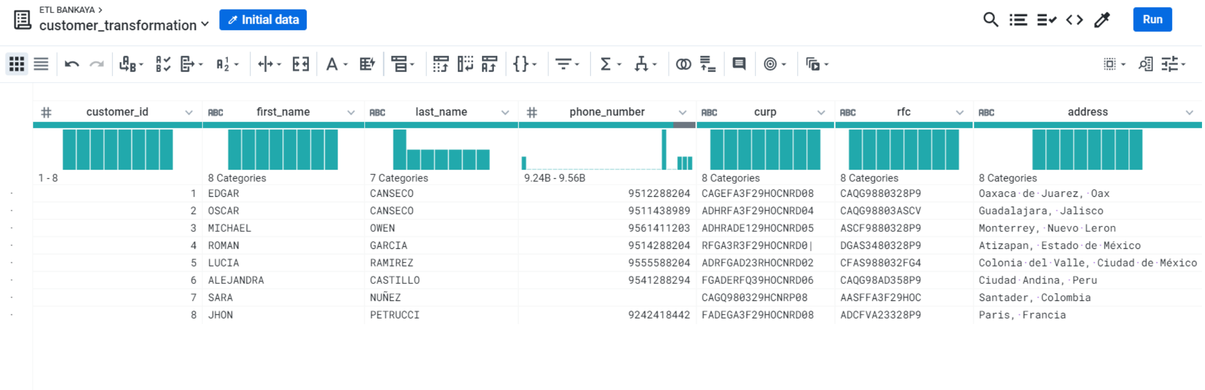


Figura 8- adding sources

You can select the preprocessing and cleaning tasks in steps.



customer_id	first_name	last_name	phone_number	curp	rfc	address
1 - 8	8 Categories	7 Categories	9.24B - 9.56B	8 Categories	8 Categories	8 Categories
1	EDGAR	CANSECO	9512288204	CAGEFA3F29HOCNRD08	CAQG9880328P9	Oaxaca de Juarez, Oax
2	OSCAR	CANSECO	9511438989	ADHRFA3F29HOCNRD04	CAQG98803ASCV	Guadalajara, Jalisco
3	MICHAEL	OWEN	9561411203	ADHRADE129HOCNRD05	ASCF9880328P9	Monterrey, Nuevo Leron
4	ROMAN	GARCIA	9514288204	RFGA3R3F29HOCNRD0|	DGAS3480328P9	Atizapan, Estado de México
5	LUCIA	RAMIREZ	9555588204	ADRFGAD23RHOCNRD02	CFAS988032FG4	Colonia del Valle, Ciudad de México
6	ALEJANDRA	CASTILLO	9541288294	FGADERFQ39HOCNRD06	CAQG98AD358P9	Ciudad Andina, Peru
7	SARA	NUÑEZ		CAGQ980329HCNRP08	AASFFA3F29HOC	Santader, Colombia
8	JHON	PETRUCCI	9242418442	FADEGA3F29HOCNRD08	ADCFVA23328P9	Paris, Francia

Figura 9- transformations

Finally the load will be to the BigQuery data warehouse where a dataset called Bankaya was created

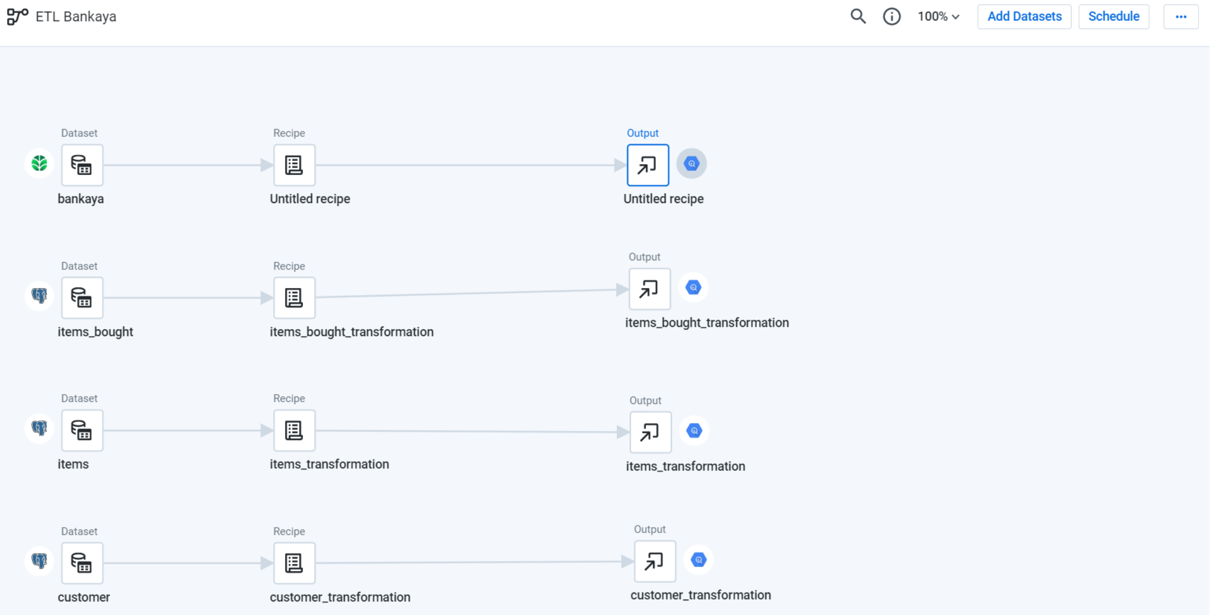
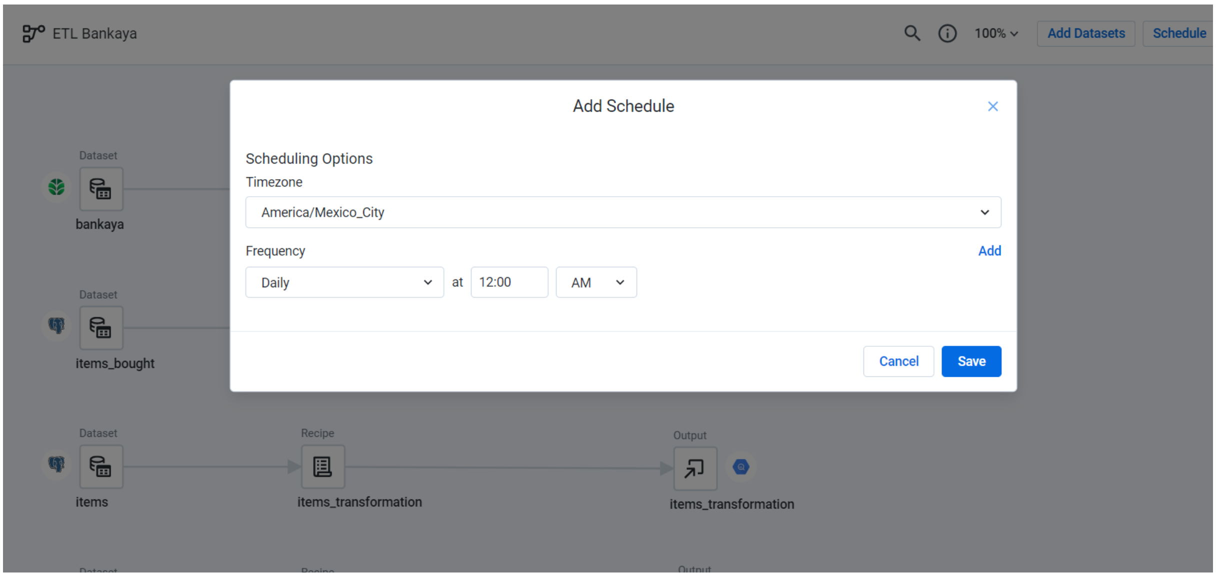


Figura 10- pipeline

It is scheduled so that the load is daily at 12:00 am with respect to the Mexico area (GMT -5)



Add Schedule

Scheduling Options

Timezone

America/Mexico_City

Frequency

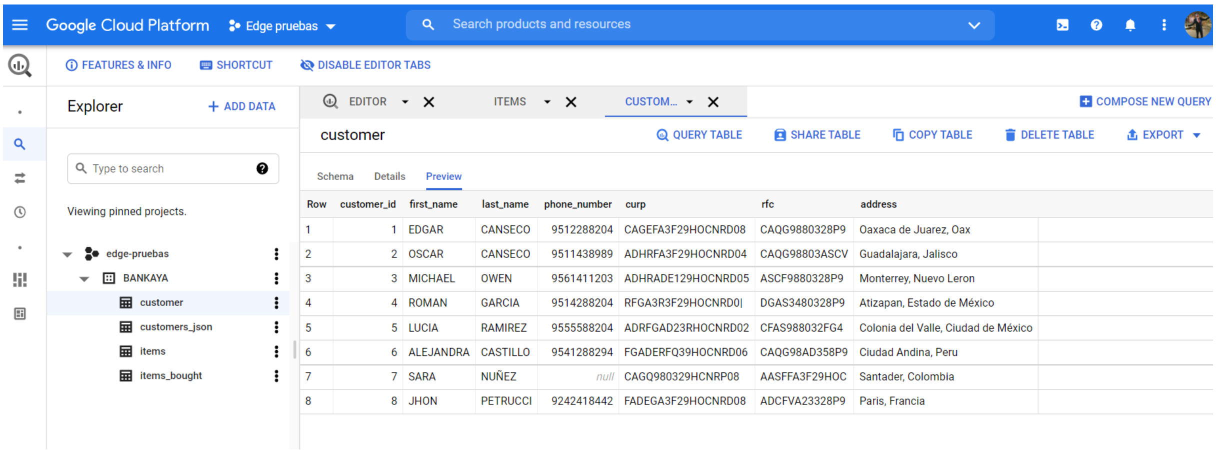
Add

Daily at 12:00 AM

Cancel Save

Figura 11- scheduling

This is the result of the pipeline with the tables in Bigquery

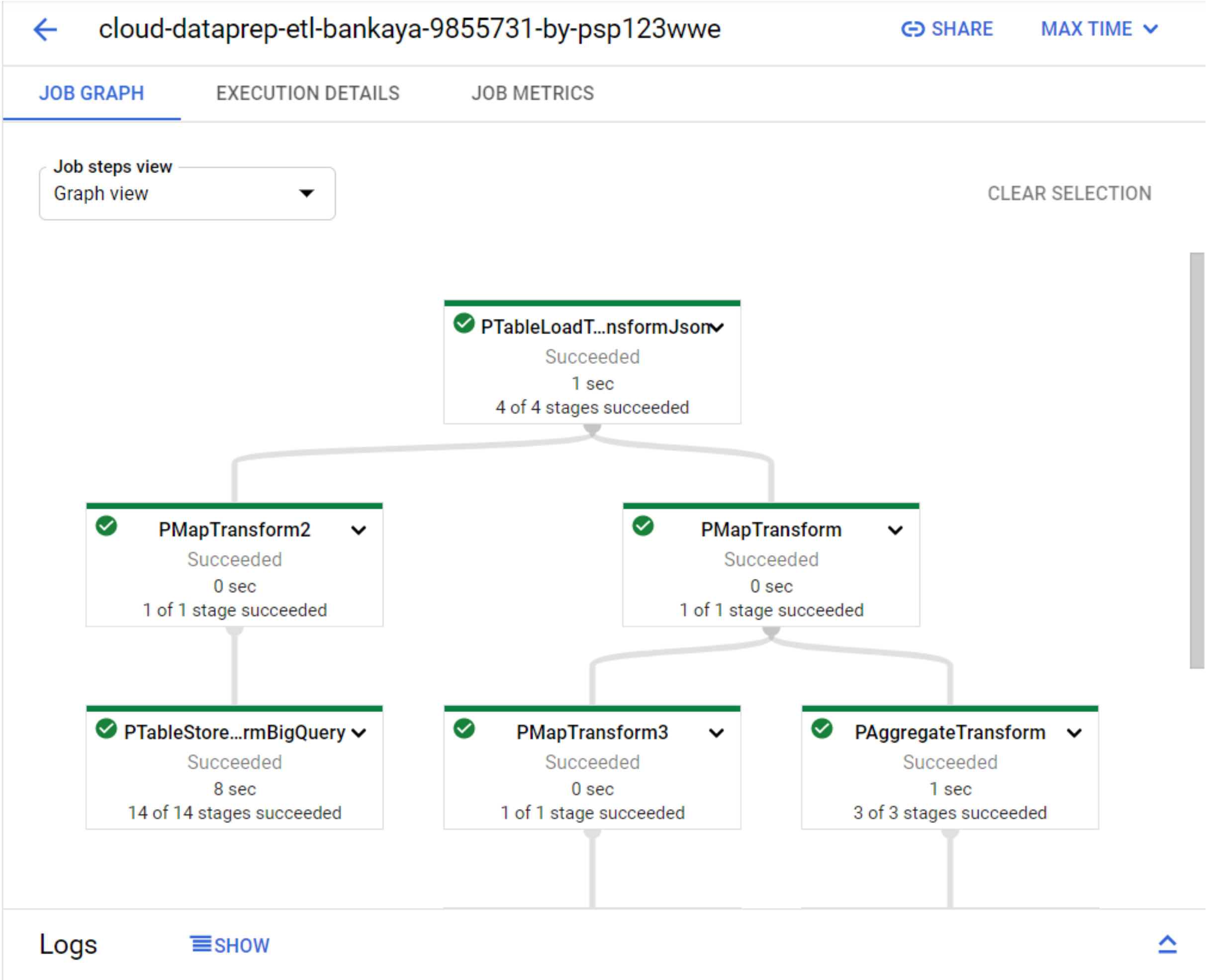


customer

Row	customer_id	first_name	last_name	phone_number	curp	rfc	address
1	1	EDGAR	CANSECO	9512288204	CAGEFA3F29HOCNRD08	CAQG9880328P9	Oaxaca de Juarez, Oax
2	2	OSCAR	CANSECO	9511438989	ADHRFA3F29HOCNRD04	CAQG98803ASCV	Guadalajara, Jalisco
3	3	MICHAEL	OWEN	9561411203	ADHRADE129HOCNRD05	ASCF9880328P9	Monterrey, Nuevo Leron
4	4	ROMAN	GARCIA	9514288204	RFGA3R3F29HOCNRD0|	DGAS3480328P9	Atizapan, Estado de México
5	5	LUCIA	RAMIREZ	9555588204	ADRFGAD23RHOCNRD02	CFAS988032FG4	Colonia del Valle, Ciudad de México
6	6	ALEJANDRA	CASTILLO	9541288294	FGADERFQ39HOCNRD06	CAQG98AD358P9	Ciudad Andina, Peru
7	7	SARA	NUÑEZ	null	CAGQ980329HCNRP08	AASFFA3F29HOC	Santader, Colombia
8	8	JHON	PETRUCCI	9242418442	FADEGA3F29HOCNRD08	ADCFVA23328P9	Paris, Francia

Figura 12- result

And this is what a workflow launched in dataflow looks like for data transformation.



cloud-dataprep-etl-bankaya-9855731-by-psp123wwe

Job steps view: Graph view

PTableLoadT...nsformJson
Succeeded
1 sec
4 of 4 stages succeeded

PMapTransform2
Succeeded
0 sec
1 of 1 stage succeeded

PMapTransform
Succeeded
0 sec
1 of 1 stage succeeded

PTableStore...rmBigQuery
Succeeded
8 sec
14 of 14 stages succeeded

PMapTransform3
Succeeded
0 sec
1 of 1 stage succeeded

PAggregateTransform
Succeeded
1 sec
3 of 3 stages succeeded

Figura 13- dataflow