

Building an ETL Pipeline and Visualization Dashboard Using GCP

1. Introduction

In today's data-driven world, businesses rely on effective pipelines to turn raw data into actionable insights. For this project, I designed an end-to-end ETL (Extract, Transform, Load) pipeline to process an e-commerce dataset. The goal was to integrate cloud-based tools for data transformation, storage, and visualization, making the dataset more usable and insightful for decision-making.

This report walks you through the process of applying concepts learned in this course, such as data transformation and distributed storage, and combining them with new tools like Google Data Fusion and Looker Studio. By integrating both familiar and advanced technologies, the project showcases a comprehensive pipeline to transform raw data into insightful visualizations.

2. Background

I observed how e-commerce has become a central part of our lives. From ordering groceries and medicines to clothing and electronics, people rely on online platforms for almost everything. This shift to online shopping generates vast amounts of data, offering a unique opportunity to analyze trends and understand consumer behavior. This sparked my interest in choosing an e-commerce dataset for this project.

The dataset provides insights into product details, sales performance, and customer preferences, making it a perfect candidate for building an ETL pipeline. Analyzing this type of data can reveal patterns and trends that are not immediately obvious, and designing a pipeline ensures the data is structured and ready for exploration.

3. Methodology

The following steps outline the process I followed to build the project. These steps, along with the tools used, are described in detail to guide you in replicating this project:

1. Data Extraction
2. Data Transformation (renaming columns, combining columns, calculating revenue)
3. Data loading in Big Query
4. Data Visualization (lookerStudio)

Steps to recreate the project: -

Start by creating a new project in GCP. Once that is done you will land on the project dashboard page as shown below.

The screenshot shows the Google Cloud Platform dashboard for the project 'FA24-I535-Ayana-EcoPipeline'. The dashboard is divided into several sections:

- Project info:** Shows the project name (FA24-I535-Ayana-EcoPipeline), project number (380640162700), and project ID (fa24-i535-ayana-ecopipeline). It also has links to 'ADD PEOPLE TO THIS PROJECT' and 'Go to project settings'.
- API APIs:** A chart titled 'Requests (requests/sec)' showing data for the selected time frame from 2 PM to 2:45. The chart indicates 'No data is available for the selected time frame.' Below the chart is a link to 'Go to APIs overview'.
- Google Cloud Platform status:** Shows 'All services normal' and a link to 'Go to Cloud status dashboard'.
- Billing:** Shows 'Estimated charges' for the billing period Nov 1 – 24, 2024 (USD \$0.00) and a link to 'View detailed charges'.
- Monitoring:** Links to 'Create my dashboard', 'Set up alerting policies', 'Create uptime checks', 'View all dashboards', and 'Go to Monitoring'.
- Resources:** A list of services including BigQuery, SQL, Compute Engine, Storage, Cloud Run functions, and Cloud Run.
- Getting Started:** A section with a link to 'Error Reporting'.

Next step is to make a storage bucket. To do this navigate to google cloud storage->buckets> create a bucket as shown in the picture below name your bucket and let everything else be the default setting.

The screenshot shows the 'Create a bucket' wizard, step 1: Name your bucket. The steps are as follows:

- Name your bucket**:
 - Pick a globally unique, permanent name. [Naming guidelines](#)
 - Example: 'example', 'example_bucket-1', or 'example.com'
 - Tip: Don't include any sensitive information
- LABELS (OPTIONAL)**
- CONTINUE**

Subsequent steps are:

- Choose where to store your data**: Location: us (multiple regions in United States), Location type: Multi-region
- Choose a storage class for your data**: Default storage class: Standard
- Choose how to control access to objects**: Public access prevention: On, Access control: Uniform
- Choose how to protect object data**: Soft delete policy: Enabled, Object versioning: Disabled, Bucket retention policy: Disabled, Object retention: Disabled, Encryption type: Google-managed

At the bottom are 'CREATE' and 'CANCEL' buttons.

Initially the bucket will be empty as shown below.

The screenshot shows the Google Cloud Storage 'Bucket details' page for the bucket 'mgmtaccess-ecom-bucket'. The bucket was created in 'us-central1 (Iowa)' with a 'Standard' storage class, 'Not public' public access, and 'Soft Delete' protection. The 'OBJECTS' tab is selected, showing a 'Folder browser' with a single entry: 'Buckets > mgmtaccess-ecom-bucket'. Below this, there are buttons for 'CREATE FOLDER', 'UPLOAD', 'TRANSFER DATA', and 'OTHER SERVICES'. A filter bar allows filtering by name prefix, type, and other metadata. A message at the bottom indicates 'No rows to display'. On the left sidebar, there are links for 'Overview', 'Monitoring', 'Settings', 'Marketplace', and 'Release Notes'. A modal at the bottom center says 'Created bucket mgmtaccess-ecom-bucket' with a close button.

Now we will write the script for extracting the data in a google collab notebook and converting it to csv format you can take the code from below or you can access it from this link:
<https://colab.research.google.com/drive/1vxxP3VdCGRbtPHJaJnWrQrbRUuu8Ywo?usp=sharing>

You now have your bucket set up, so simply update the bucket name in the provided code. I've also included the authentication commands in the Colab file, so you don't need to worry about that.

The screenshot shows a Google Colab notebook titled 'Project.ipynb'. The code cell [1] contains the command '%pip install faker', which installs the Faker library. The output shows the package being downloaded and installed successfully. The code cell [3] imports the 'auth' module from 'google.colab' and calls 'authenticate_user()'. The code cell [5] runs '!pip install google-cloud-storage' to install the Google Cloud Storage package. The output for this cell shows that the requirement is already satisfied for both 'google-cloud-storage' and 'google-auth'. The notebook interface includes a toolbar with File, Edit, View, Insert, Runtime, Tools, Help, and a status bar showing RAM Disk and Gemini.

```
Project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Os
{x}
Os
import csv
from faker import Faker
import random

# Initialize Faker
fake = Faker()

# Number of records
num_records = 1000

# Product categories
categories = ['Electronics', 'Clothing', 'Home & Kitchen', 'Books', 'Sports']

# Generate e-commerce data
file_name = "ecommerce_data.csv"
with open(file_name, mode='w', newline='') as file:
    fieldnames = [
        "order_id", "product_name", "category", "price", "units_sold",
        "rating", "purchase_date", "customer_id"
    ]
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    for _ in range(num_records):
        writer.writerow({
            "order_id": fake.uuid4(),
            "product_name": fake.word().capitalize() + " " + fake.word().capitalize(),
            "category": random.choice(categories),
            "price": round(random.uniform(10, 500), 2),
            "units_sold": random.randint(1, 100),
            "rating": round(random.uniform(1, 5), 1),
            "purchase_date": fake.date_between(start_date='-1y', end_date='today'),
            "customer_id": fake.uuid4()
        })
print(f"Dataset created: {file_name}")

Dataset created: ecommerce_data.csv
```

```
{x}
{x}
Os
from google.cloud import storage

# Function to upload file to GCS
def upload_to_gcs(bucket_name, source_file_name, destination_blob_name):
    storage_client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(destination_blob_name)
    blob.upload_from_filename(source_file_name)
    print(f"File {source_file_name} uploaded to {destination_blob_name} in bucket {bucket_name}.")

# Configure your bucket and file details
bucket_name = "mgmtaccess-ecom-bucket"
source_file_name = "ecommerce_data.csv"
destination_blob_name = "ecommerce_data.csv"

# Upload the file
upload_to_gcs(bucket_name, source_file_name, destination_blob_name)

File ecommerce_data.csv uploaded to ecommerce_data.csv in bucket mgmtaccess-ecom-bucket.
```

Once you run the code, you should see a file named ecommerce_data.csv in the bucket.

Bucket details

mgmtaccess-ecom-bucket

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	Soft Delete

OBJECTS **CONFIGURATION** **PERMISSIONS** **PROTECTION** **LIFECYCLE** **OBSERVABILITY** **INVENTORY REPORTS** **OPERATIONS**

Folder browser

Buckets > mgmtaccess-ecom-bucket

CREATE FOLDER **UPLOAD** **TRANSFER DATA** **OTHER SERVICES**

Name	Size	Type	Created	Storage class	Last modified
ecommerce_data.csv	119.7 KB	text/csv	Nov 24, 2024, 2:55:10 PM	Standard	Nov 24, 2024, 2:55:10 PM

1. Data Extraction:

We will create a Data Fusion instance, you just must enter the name of the cluster and select the right region (in my case uswest1), and leave everything on default, it will take some time to run approximately 15 minutes.

Create Data Fusion instance

Instance name *

Alphanumeric characters, space and - only For eg: My Instance name-1024. Name must start with a letter, 30 character max

Instance ID

Description

Region
us-west1

Region in which the instance is created.

Version
6.9.2

Edition

Developer
This edition provides a full-feature edition for product exploration and development environments with zonal availability and limitation on execution environment.

Basic
This edition provides comprehensive data integration capabilities. Users can build batch data pipelines; connect to any data source; perform code-free transformations. Limitation on simultaneous pipeline runs. Recommended for non-critical environments.

Enterprise
This edition provides all the functionality provided in the Basic edition. In addition, includes support for real-time data pipelines; interactions with data lineage; higher scalability; and high availability. Recommended for critical environments.

ADD ACCELERATORS 0 accelerators added

Authorization

Dataproc Service Account
835516423674-compute@developer.gserviceaccount.com

A service account is used to authorize the instance to run a Dataproc job.

Advanced Options

The instance creation will take approximately 20 minutes.

CREATE **CANCEL**

The screenshot shows the Google Cloud Data Fusion Instances page. At the top, there are navigation links for 'Data Fusion' and 'Instances'. Below the header, a search bar contains 'data fusion'. A progress bar at the bottom indicates 'Creating instance ecommerce-data-pipeline'.

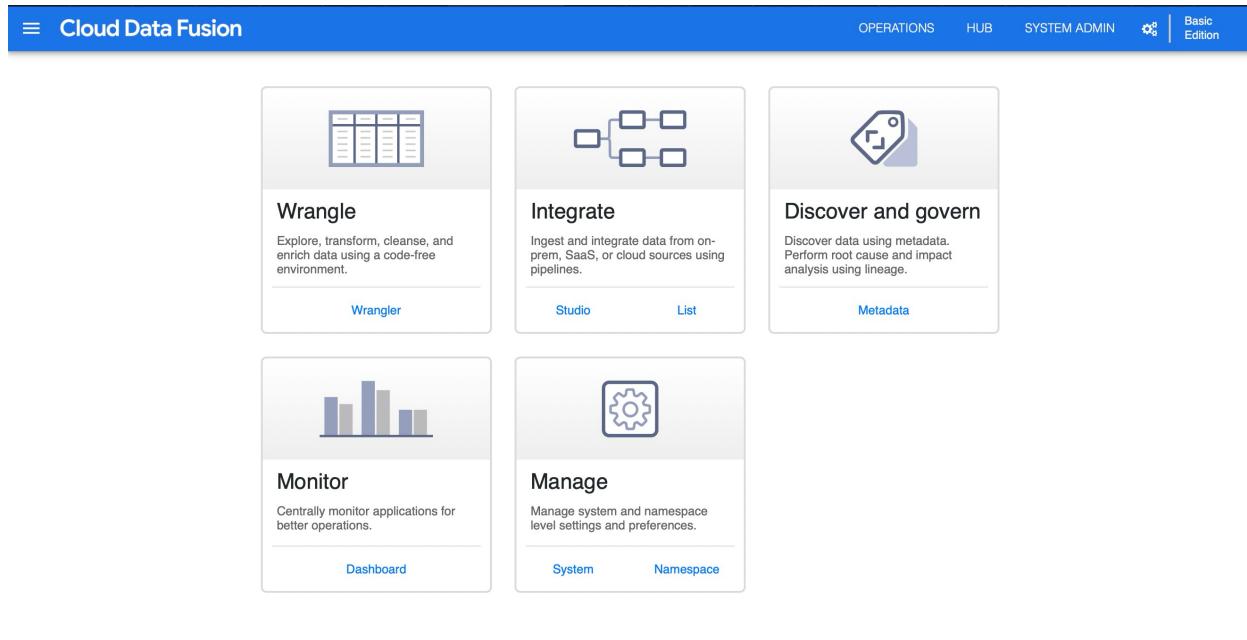
Instance Name	Action	Edition	Region	Zone	Version	Notifications	Encryption
ecommerce-data-pipeline	Creating...	Basic	us-west1	--	6.10.1(6.10.1.1)		Google-managed

Your Data Fusion instance, which is a Dataproc instance, should now be ready. Click on the "View Instance" button, and you will be prompted to authenticate yourself.

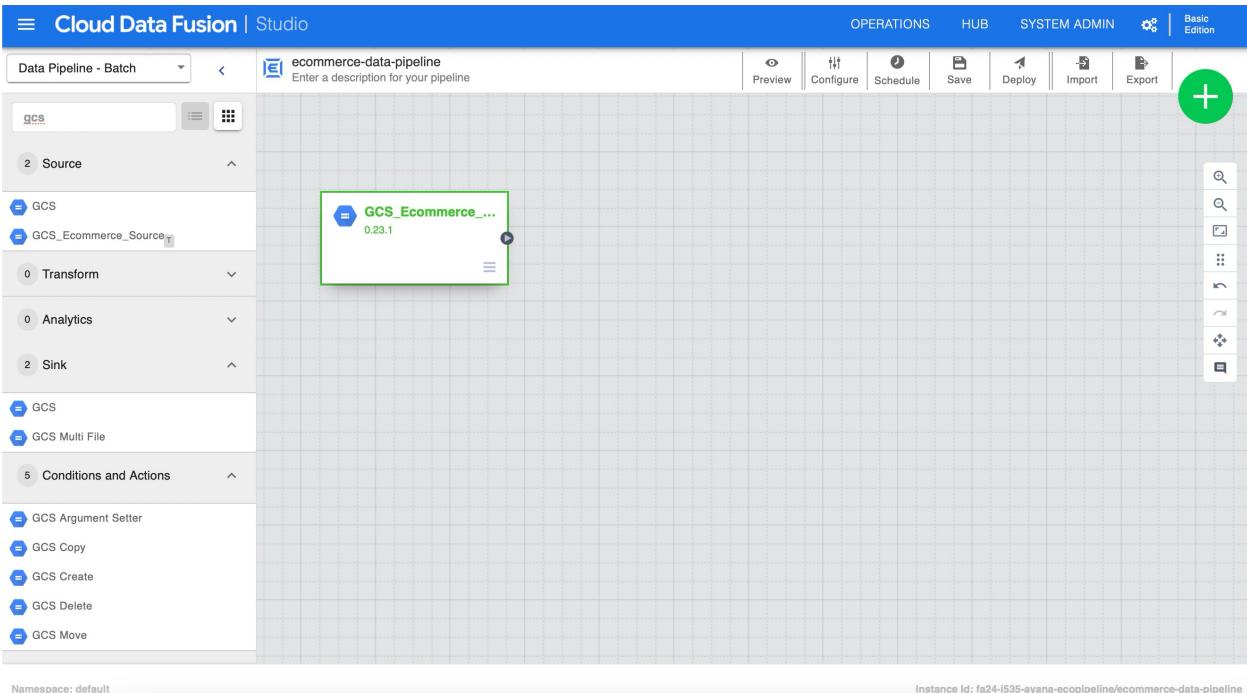
The screenshot shows the Google Cloud Data Fusion Instances page. The instance 'ecommerce-data-pipeline' is now listed with a status of 'Ready'. The 'View Instance' button is highlighted in blue.

Instance Name	Action	Edition	Region	Zone	Version	Notifications	Encryption
ecommerce-data-pipeline	View Instance	Basic	us-west1	--	6.10.1(6.10.1.1)		Google-managed

Once authenticated, you will land on the following page. Select "Studio" to create the pipeline.



- In Cloud Data Fusion Studio, click on "**Create Pipeline**".
- Select "**Batch**" as the pipeline type.
- Add the "**Cloud Storage**" source icon to the canvas and double click to open properties configure it to point to your e-commerce CSV file stored in Google Cloud Storage.
- Set default properties but make sure you name the path to your bucket and point to your project.
- Click validate to check for any errors.

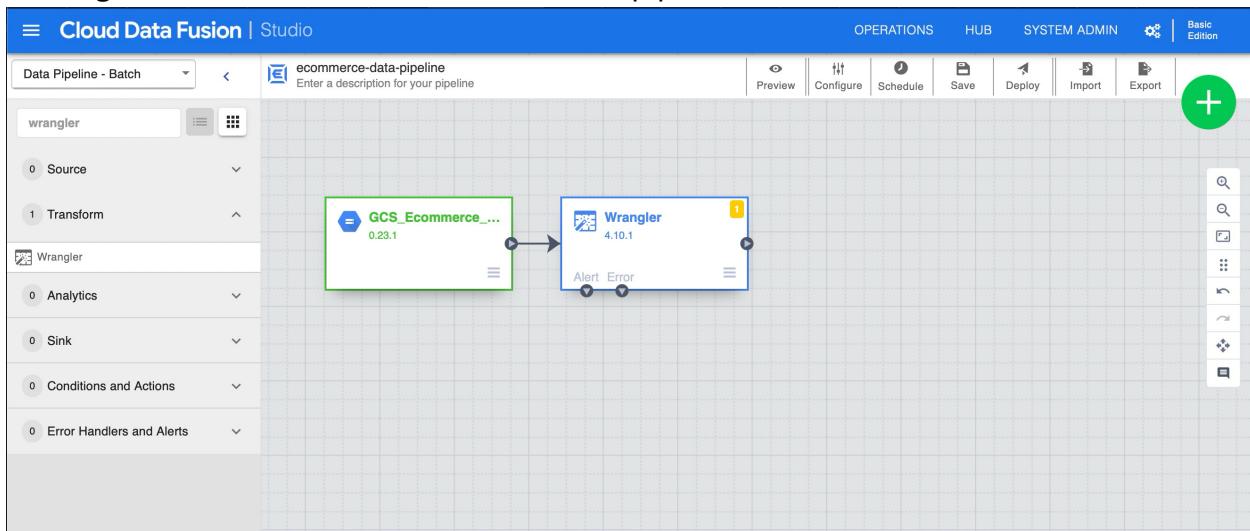


2. Data Transformation

Data Fusion's Wrangler tool was used to clean and enhance the dataset by applying several transformations.

- From the component's menu, drag and drop the "Wrangle" transformation icon onto the canvas.
- Connect it to the Cloud Storage source.
- Within the Wrangle transformation (double click to open), you can apply your transformations, like renaming columns, creating new calculated fields (e.g., product_details and revenue).

An image of when both icons are added to the pipeline.



For all the transformations I have mentioned below you just have to select the column you want to apply the transformation and click on the dropdown menu icon at the top of the column of any column (if you have selected more than one) and you will get the list of options from which you can choose.

Transformations I did:

Combining Columns:

- A new column, `product_details`, was created by merging the `product_name` and `category` columns. This provided a descriptive label for the products.
- **Formula Used:** `product_name + ' (' + category + ')'`
- **Purpose:** To make product analysis easier by creating a detailed product description.

☰ Cloud Data Fusion | Studio

Cloud Storage Default - mgmtaccess-ecom-bucket/commerce..
ecommerce_data.csv Columns: 8 | Rows: 1000

	String	String	String				
	<input type="checkbox"/> product_name	<input checked="" type="checkbox"/> category	<input checked="" type="checkbox"/> price				
c	Million Pattern						
ab9	High Place						
bed	Set Girl						
707	National Decade						
5c	Mr Analysis						
d9	Bill Stop						
6	Hope It						
ld	Coach Camera						
ea4	Research Quality						
5e1	Tree Single						
008	Painting Support						
36b	Season Certain						
6	Country Sign						
96c	What Part	Books	61.99	13	1.9	2024-01-08	cfc645a4-9081-453f-ba6f-e3cfe7ad6f4d
af	Employee Position	Home & Kitchen	308.94	52	2.7	2024-06-30	643f4f53-77da-40de-b8fd-7e40b00cb5eb

Set order

product_name ↗

category

Choose delimiter

Comma

Name new column

product_details

Join Cancel

Insights

String	String	String	
2024-08-26	4dcc151b-1d0c-49a7-99c0-e4a53476b1ec		
2024-05-15	fdffcc19-5b89-4ecc-a1e5-14056b7ede47		
2024-06-28	73aeab9b-bee8-470c-9182-d3faef89626		
2024-09-03	05e85ddd-3181-4226-8b2b-608243ebe6d2		
2024-01-11	972cf084-f298-45f5-90b4-a0dd099e0973		
85	3.6	2024-06-01	6758f193-e85f-4c0f-bf93-a3ad4229081b
37	3.2	2024-04-01	a40534e9-bf6f-406c-8a4b-2dc3af1781ed
57	3.0	2024-11-03	3a810ba7-2081-431b-9af0-218e7bdf8258
15	3.8	2024-08-16	8164e2ce-78b3-4479-9a4f-0158dc75a5ff
50	4.3	2024-06-25	99280c4e-d4bb-4e19-84cc-17e0a515a78c
45	3.2	2024-01-27	79b7ab8d-1678-4285-a8af-297f6e6840aa
60	4.0	2024-01-09	2f33b88b-1163-4273-a34d-0cf80cbcbb45
91	4.2	2024-04-07	8745b866-e2f5-49a0-a9e4-7a8d204118aa

Insights

`_id`

Parse

Set character encoding

Change data type

Format

Calculate

Custom transform

Filter

Send to error

Find and replace

Fill null or empty cells

Copy column

Delete column

Keep column

Join two columns

Swap two column names

Extract fields

Explode

5-4

1-453f-ba6f-e3cfe7ad6f4d

3-40de-b8fd-7e40b00cb5eb

Type the custom expression to transform "product_details"

`product_name + ' (' + category + ')'`

Transform

name :category

price * units_sold

venue

Apply Cancel

String

product_details

Million Pattern (Books)

High Place (Sports)

Set Girl (Sports)

National Decade (Clothing)

Mr Analysis (Home & Kitchen)

Bill Stop (Sports)

Hope It (Electronics)

Coach Camera (Electronics)

Research Quality (Books)

Calculating Revenue:

- To compute the total revenue for each transaction, a new column, `revenue`, was created by multiplying the `price` and `units_sold` columns.
- Formula Used:** $\text{price} * \text{units_sold}$
- Purpose:** To evaluate the monetary performance of products and categories.

Cloud Data Fusion | Studio

Type the custom expression to transform "price"

product_name category revenue units_sold rating purchase_date customer_id

Columns (9) Transformation steps (1)

Transformations

1 merge :product_name :category :product_details ,

Renaming Columns:

- Some columns were renamed for better readability and alignment with the analysis goals. Example: The column price was renamed to revenue after applying calculations.

Validating Data:

- Checked for null or inconsistent values across columns and ensured transformations were applied correctly.

This image shows the transformations that I have applied on this data:

Cloud Data Fusion | Studio

Data Insights

product_name category revenue units_sold rating purchase_date customer_id

Columns (9) Transformation steps (3)

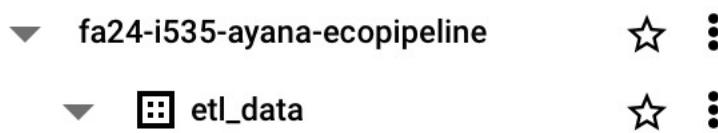
Transformations

1 merge :product_name :category :product_details ,

2 set-column :price price * units_sold

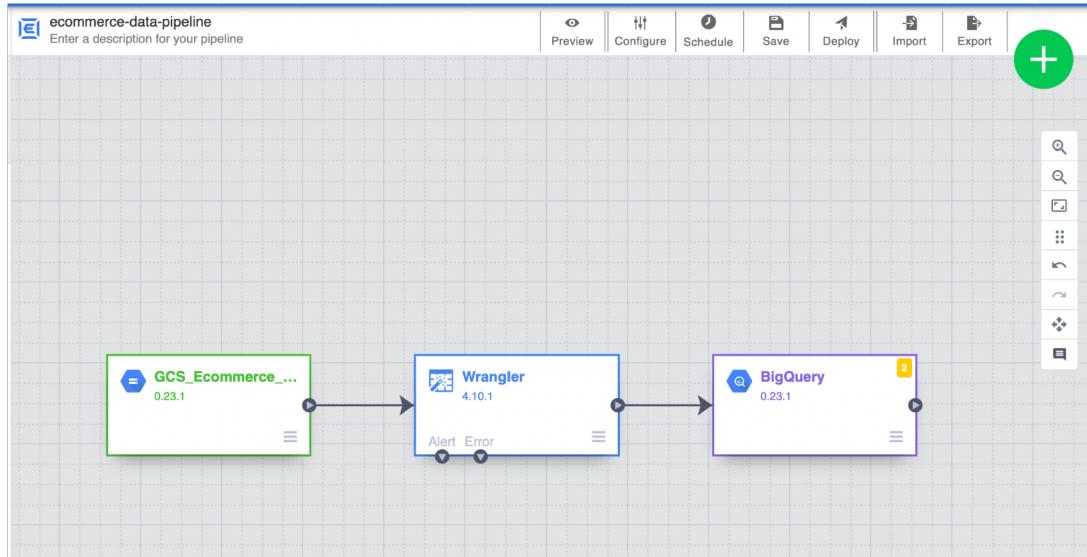
3 rename price revenue

Create a dataset in BigQuery like I have and use the same table name in the BigQuery properties below in the pipeline

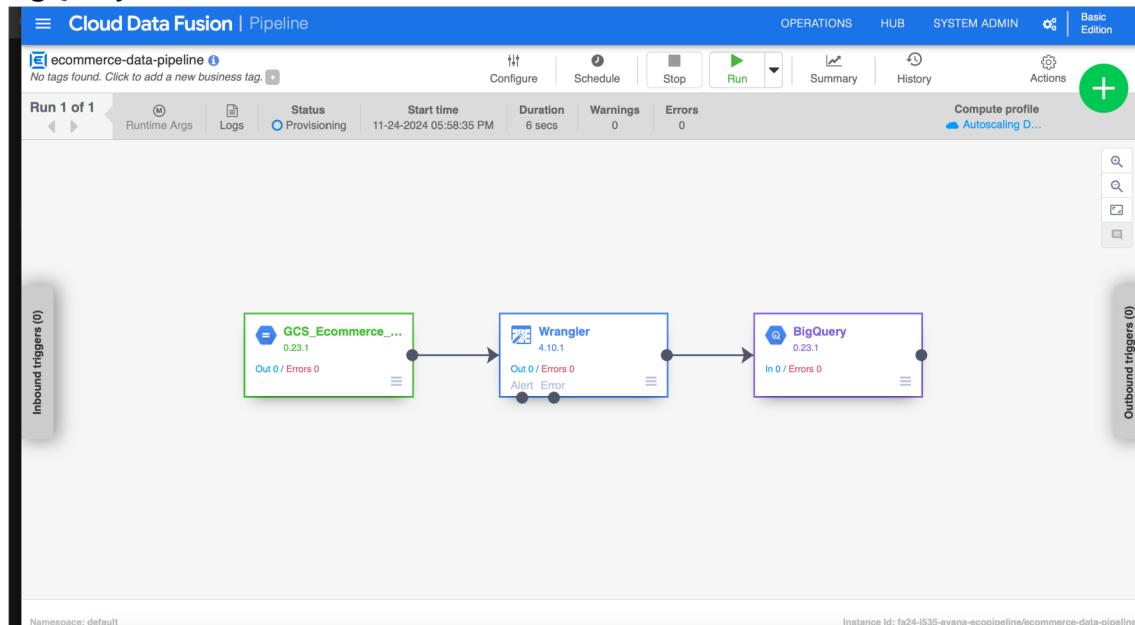


3. Load Data into BigQuery:

- Drag and drop the "BigQuery" sink (for structured and queryable format) icon to the canvas.
- Connect it to the "Wrangle" transformation.
- Configure the BigQuery sink to load the transformed data into a BigQuery table.
- Specify the name of the new transformed data, in my case I names it as transformed_data.
- Validate for errors



Hit the save button and then click on the Deploy button on the Cloud Data Fusion page. It will take you to the below page where you should click “Run” to execute the ETL process. It will take some time to run, and you can see your transformed_data in the dataset of BigQuery.



Once the data is loaded into BigQuery you can see it would look alike the image below.

transformed_data

SCHEMA

Field name	Type	Mode	Key	Collation	Defa
order_id	STRING	NULLABLE	-	-	-
product_name	STRING	NULLABLE	-	-	-
category	STRING	NULLABLE	-	-	-
price	FLOAT	NULLABLE	-	-	-
units_sold	INTEGER	NULLABLE	-	-	-
rating	FLOAT	NULLABLE	-	-	-
purchase_date	DATE	NULLABLE	-	-	-
customer_id	STRING	NULLABLE	-	-	-
product_details	STRING	NULLABLE	-	-	-
revenue	FLOAT	NULLABLE	-	-	-

After creating the dataset in BigQuery, I executed the following aggregation queries to analyze the data and extract meaningful insights. These queries provided valuable insights and served as the basis for creating visualizations that highlighted key trends and patterns in the dataset.

Query 1: Total Revenue per Category

Calculates the total revenue for each product category.

Untitled query

```

1 SELECT
2   category,
3   SUM(revenue) AS total_revenue
4 FROM
5   `fa24-i535-ayana-ecopipeline.etl_data.transformed_data`
6 GROUP BY
7   category
8 ORDER BY
9   total_revenue DESC;
  
```

Query results

JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	category ▾	total_revenue ▾			
1	Clothing	2759017.390000...			
2	Sports	2595116.719999...			
3	Electronics	2384527.550000...			
4	Books	2351950.660000...			
5	Home & Kitchen	2321869.299999...			

Query 2: Average Rating and Units Sold per Product Category

Calculates the average rating and total units sold for each product category.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the 'Explorer' section with resources under 'fa24-i535-ayana-ecopipeline'. The main area shows an 'Untitled query' with the following SQL code:

```
1 SELECT
2   category,
3   AVG(rating) AS average_rating,
4   SUM(units_sold) AS total_units_sold
5 FROM
6   `fa24-i535-ayana-ecopipeline.etl_data.transformed_data`
7 GROUP BY
8   category
9 ORDER BY
10  average_rating DESC;
```

The 'Query results' section displays the following data:

category	average_rating	total_units_sold
Clothing	3.150236966824...	10453
Electronics	3.085786802030...	9434
Home & Kitchen	3.052912621359...	9587
Books	2.909239130434...	9504
Sports	2.893069306930...	10395

Query 3: Monthly Revenue

Calculates the total revenue generated each month based on the purchase_date.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the 'Explorer' section with resources under 'fa24-i535-ayana-ecopipeline'. The main area shows an 'Untitled query' with the following SQL code:

```
1 SELECT
2   EXTRACT(YEAR FROM purchase_date) AS year,
3   EXTRACT(MONTH FROM purchase_date) AS month,
4   SUM(revenue) AS total_revenue
5 FROM
6   `fa24-i535-ayana-ecopipeline.etl_data.transformed_data`
7 GROUP BY
8   year, month
9 ORDER BY
10  year, month;
```

The 'Query results' section displays the following data:

year	month	total_revenue
2023	11	152926.900000...
2023	12	1230831.57
2024	1	1093405.109999...
2024	2	828973.960000...
2024	3	1234937.319999...
2024	4	1027644.340000...
2024	5	1000468.379999...
2024	6	1060845.980000...
2024	7	934520.809999...

4. Data Visualization

Visualizations were created using Looker Studio to derive insights from the dataset. Go to [lookerStudio](#) and click on blank report and now you will see multiple options to use the data of your choice.



Looker Studio



Search Looker Studio



Create

Recent

Reports

Data sc



Recent



Shared with me



Owned by me

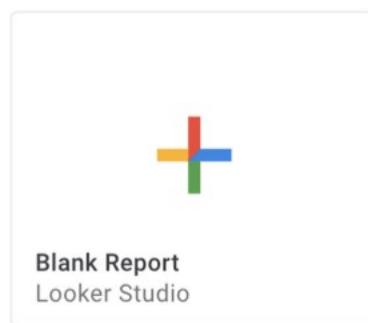


Trash



Templates

Start with a Template



Select bigQuery and you will see your dataset, select it and now select the table that we used till now as shown in the following screenshots:

Add data to report X

Connect to data My data sources

Search X

Google Connectors (24)
Connectors built and supported by Looker Studio [Learn more](#)

Looker By Google Connect to your Looker semantic models.	Google Analytics By Google Connect to Google Analytics.	Google Ads By Google Connect to Google Ads performance report data.	Google Sheets By Google Connect to Google Sheets.
BigQuery By Google Connect to BigQuery tables and custom queries.	AppSheet By Google Connect to AppSheet app data.	File Upload By Google Connect to CSV (comma-separated values) files.	Amazon Redshift By Google Connect to Amazon Redshift.
Apigee <small>PREVIEW</small> By Google Connect to Apigee API analytics and monetization data.	Campaign Manager 360 By Google Connect to Campaign Manager 360 data.	Cloud Spanner By Google Connect to Google Cloud Spanner databases.	Cloud SQL for MySQL By Google Connect to Google Cloud SQL for MySQL databases.
Display & Video 360 By Google Connect to Display & Video 360 report data.	Extract Data By Google Connect to Extract Data.	Google Ad Manager 360 By Google Connect to Google Ad Manager data.	Google Cloud Storage By Google See your files in Google Cloud Storage.

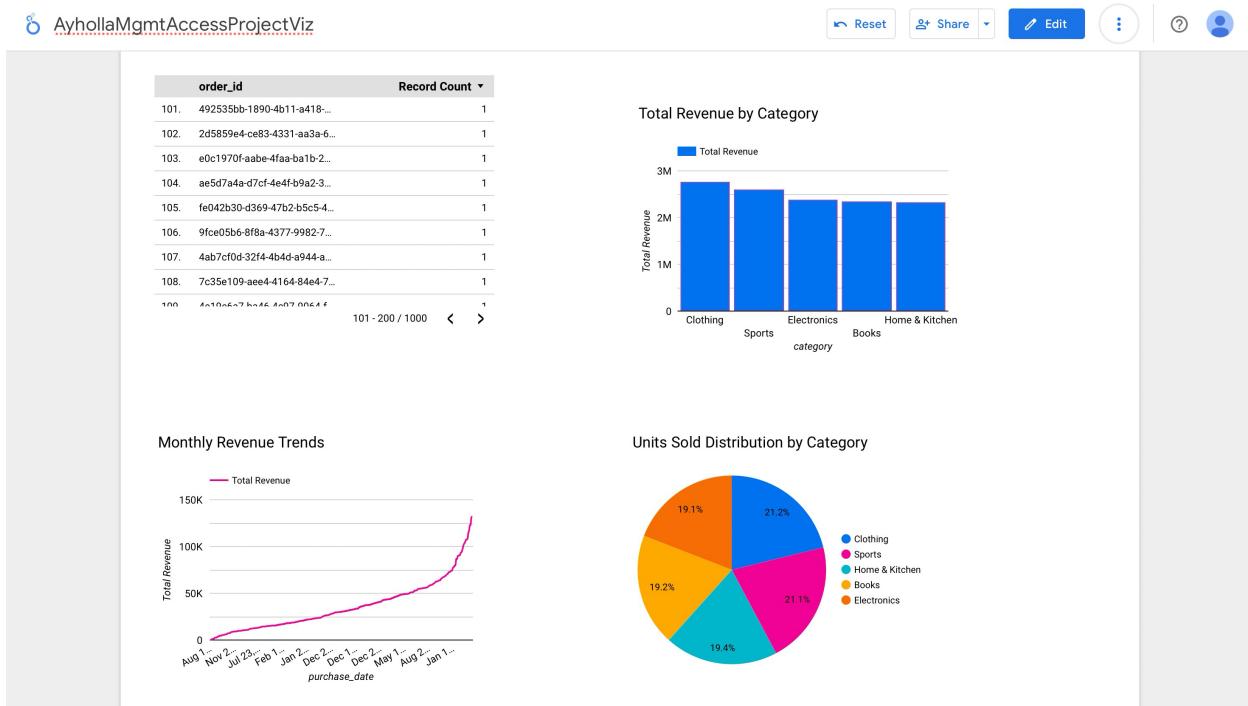
The screenshot shows the Looker Studio interface for a 'BigQuery' project. The top navigation bar includes 'File', 'View', 'Page', 'Help', 'Reset', 'Share', 'View', and user profile icons. Below the navigation is a toolbar with icons for 'Add page', 'Add data', 'Add a chart', 'Add a control', and 'Theme and layout'. A message banner at the top says 'Make your BigQuery reports load even faster with BigQuery BI Engine. Learn More'. The main area displays a 'RECENT PROJECTS' sidebar with sections for 'MY PROJECTS', 'SHARED PROJECTS', 'CUSTOM QUERY', and 'PUBLIC DATASETS'. Under 'CUSTOM QUERY', there is a search bar for 'Enter Project Id manually' and a selected project 'FA24-I535-Ayana-EcoPipeline'. To the right, there are three search bars for 'Project', 'Dataset', and 'Table', all set to 'Search Datasets' and 'transformed_data'. At the bottom right are 'Cancel' and 'Add' buttons.

The screenshot shows the 'transformed_data' table configuration screen. The top navigation bar includes 'Share', 'Help', and 'User Profile'. Below it are buttons for 'CREATE REPORT' and 'EXPLORE'. The main area has tabs for 'EDIT CONNECTION' and 'FILTER BY EMAIL'. A search bar for 'Search fields' is also present. The table structure is divided into 'DIMENSIONS (10)' and 'METRICS (1)'. The dimensions listed are: category (Text), customer_id (Text), order_id (Text), price (Number), product_details (Text), product_name (Text), purchase_date (Date), rating (Number), revenue (Number), and units_sold (Number). The metric listed is: Record Count (Number). At the bottom, there are buttons for 'REFRESH FIELDS' and '11 / 11 Fields'.

Now you have your table loaded in the lookerStudio you can use the menu for multiple options for visualisations. I have used tables, pie charts, line graph and bar chart for presentation of my data as shown below.

Three key visualizations were developed:

- Bar Chart - Revenue by Category:** Displayed the total revenue generated by each product category, highlighting the top-performing segments.
- Line Chart - Monthly Revenue Trends:** Captured the revenue trends over time, grouped by month, to provide an overview of seasonal patterns and growth.
- Pie Chart - Units Sold by Category:** Illustrated the proportion of units sold across different categories, offering a snapshot of sales distribution.



4. Results

The project successfully implemented an ETL data pipeline using Google Cloud Technologies, transforming and analyzing an e-commerce dataset to derive actionable insights. Each stage of the pipeline, from data generation to visualization, contributed to uncovering meaningful patterns and trends.

Data Generation

The dataset was generated using the Faker library, which created a synthetic e-commerce dataset containing fields such as product names, categories, prices, units sold, and purchase dates. This dataset simulated real-world e-commerce scenarios, making it an excellent foundation for the ETL pipeline. The generated data was saved as a CSV file and uploaded to Google Cloud Storage, ensuring it was accessible for further processing.

Data Extraction

The dataset stored in Google Cloud Storage was accessed using Cloud Data Fusion, which seamlessly integrated the data into the ETL pipeline for transformations.

Data Transformation

The transformations were conducted in Cloud Data Fusion using the Wrangler tool. Two key transformations were applied to enrich the dataset:

- A new column, product_details, was created by combining product_name and category, providing a descriptive label for each product.
- A revenue column was calculated by multiplying price and units_sold, enabling insights into product and category financial performance.

These transformations structured the data for analysis and ensured it was ready for loading into BigQuery.

Data Loading into BigQuery

The transformed dataset was loaded into BigQuery under the dataset ecommerce_dataset and table transformed_data. BigQuery's scalable infrastructure allowed for efficient querying and analysis of the enhanced dataset.

Visualizations and Insights

The final dataset was visualized using Looker Studio, producing the following charts:

- Bar Chart - Revenue by Category: This chart highlighted the total revenue generated by each category. Categories like Clothing and Sports were the top performers, driving the majority of sales revenue.
- Line Chart - Monthly Revenue Trends: Revenue trends over time showed steady growth, with occasional spikes suggesting seasonal demand or promotional impacts.
- Pie Chart - Units Sold Distribution by Category: This chart illustrated the proportion of units sold for each category, with Clothing leading slightly, reflecting its popularity.

Key insights from these visualizations include the identification of high-performing product categories, revenue seasonality, and balanced sales distribution across categories. These findings demonstrate how a well-designed pipeline can provide actionable insights for ecommerce businesses, helping them make informed decisions based on data.

5. Discussion

How the Project Aligns with Course Topics

This project incorporates several concepts and tools covered throughout the course. Key alignments include:

- **Big Data Lifecycle and Pipelines (Module 6):** The project exemplifies the lifecycle of data, starting from extraction, transformation, and loading (ETL) to final visualization. It also highlights the use of Cloud Data Fusion for pipeline automation, showcasing sequencing and scaling capabilities.

- **Cloud Computing (Module 3):** By leveraging Google Cloud Platform (GCP), the project demonstrates the practical use of cloud services for big data processing and storage.
- **Ingest and Storage (Module 7):** BigQuery, a data warehouse solution, was used to store the transformed data. This aligns with the module's focus on organizing and retrieving big data.
- **Processing and Analytics (Module 9):** Data transformations applied in Cloud Data Fusion ensured the quality and usability of the dataset, making it ready for further analysis.

These connections reinforce how theoretical concepts from the course translate into realworld applications, bridging the gap between learning and execution.

Challenges Faced

Throughout the project, I encountered several challenges:

1. **Quota Limitations in GCP:** The initial deployment of the pipeline failed due to insufficient disk space quotas. This required modifying configurations to reduce the size of the dataset and adjusting resource allocations in GCP.
2. **Complexity of Data Fusion Interface:** Configuring the ETL pipeline in Cloud Data Fusion involved a learning curve. Issues such as misconfigured transformations and validation errors required additional troubleshooting.
3. **Visualization Customization in Looker Studio:** Creating interactive and accurate visualizations, especially grouping dates for monthly trends, required multiple iterations and adjustments to ensure clarity.

How Challenges Were Addressed

- To resolve the **quota limitations**, I optimized the pipeline configuration and used a smaller dataset to fit within the GCP limits.
- For the **Cloud Data Fusion interface**, I referred to tutorials and documentation to understand the configuration steps and successfully implement the pipeline.
- In Looker Studio, I experimented with data fields and visualization settings to overcome grouping issues, resulting in cleaner and more meaningful charts.

Skills Gained

This project deepened my understanding of:

- Building scalable ETL pipelines with cloud-based tools.
- Managing and transforming big data for analysis.
- Creating insightful visualizations to communicate findings effectively. These skills are essential for handling real-world big data scenarios and align with my learning objectives for this course.

6. Conclusion

The project highlights the potential of cloud-based tools like Google Cloud Data Fusion and BigQuery in managing and analyzing large datasets. By processing e-commerce data, I was able to uncover valuable insights, such as identifying high-revenue categories, analyzing monthly trends, and understanding customer purchasing patterns.

For e-commerce businesses, the findings demonstrate how data pipelines can streamline operations and support data-driven decision-making. For example, insights into revenue trends and product performance can help optimize inventory, improve marketing strategies, and enhance customer satisfaction.

Overall, this project reinforced the importance of combining scalable infrastructure with robust data processing techniques to derive actionable insights. While the journey presented challenges, it was a rewarding experience that bridged the gap between theoretical concepts and practical applications.

You can also access all the code templates on my github-

<https://github.iu.edu/ayholla/I-535MgmtAccessProject>

References:

- GCP SDK installation: <https://cloud.google.com/sdk/docs/install-sdk> • Login on Colab with gcloud without service account • Google Cloud Platform: YouTube. Available online: <https://www.youtube.com/user/googlecloudplatform>
- Towards Data Science: Medium. Available online: <https://towardsdatascience.com/>
- Google Cloud Blog: Available online: <https://cloud.google.com/blog>
- VPC networks: <https://cloud.google.com/vpc/docs/vpc>
- Use uniform bucket-level access: <https://cloud.google.com/storage/docs/usinguniform-bucket-level-access>
- GCP SDK setup: <https://www.codingforentrepreneurs.com/blog/google-cloud-cliand-sdk-setup/>
- Questions tagged 'google-cloud-platform': Stack Overflow. Available online: <https://stackoverflow.com/questions/tagged/google-cloud-platform>
- ChatGPT: <https://chat.openai.com/c/051a2191-31a1-47b6-b1da-931f10581ce2>