## Step-by-Step Guide: From Raw Data to AI-Ready Platform on GCP Using Dataplex

### Scenario

You are starting with a fresh Google Cloud Platform project. You have **Editor rights** and access to the Google Cloud Console. Your task is to:

1. Upload simulated data to Cloud Storage
2. Organize it into zones using Dataplex
3. Load, transform, and analyze data in BigQuery
4. Use Generative AI tools to support analysis or dashboard creation

## 1. ✅ **Initial GCP Setup**

### a. Enable Required APIs

gcloud services enable \  
 storage.googleapis.com \  
 dataplex.googleapis.com \  
 bigquery.googleapis.com \  
 bigquerystorage.googleapis.com \  
 aiplatform.googleapis.com

### b. Create GCS Bucket

gsutil mb -l europe-west1 gs://your-bucket-name

Upload your files (CSV):

gsutil cp customers.csv transactions.csv support\_logs.csv gs://your-bucket-name/raw/

## 2. 🗂️ **Create Dataplex Lake and Zones**

### a. Create a Dataplex Lake

From the GCP Console:

* Go to **Dataplex > Lakes**
* Click **Create Lake**
* Name: banking-lake
* Region: europe-west1

### b. Create Dataplex Zones

* **Raw Zone**: raw-zone (for unprocessed CSVs)
* **Curated Zone**: curated-zone (transformed BQ tables)
* **Analytics Zone**: analytics-zone (cleaned BQ views or models)

Each zone can use the same GCS bucket or BigQuery dataset as a backing store.

## 3. 🧾 **Load Data into BigQuery**

### a. Create BigQuery Dataset

bq mk --location=europe-west1 your\_dataset\_name

### b. Load CSV Files

bq load --autodetect --skip\_leading\_rows=1 \  
 your\_dataset.customers \  
 gs://your-bucket-name/raw/customers.csv  
  
bq load --autodetect --skip\_leading\_rows=1 \  
 your\_dataset.transactions \  
 gs://your-bucket-name/raw/transactions.csv  
  
bq load --autodetect --skip\_leading\_rows=1 \  
 your\_dataset.support\_logs \  
 gs://your-bucket-name/raw/support\_logs.csv

### c. Register in Dataplex

From **Dataplex > Assets**, register each BigQuery table under the relevant zone (e.g. raw-zone).

## 4. 🧪 **Transform & Analyze**

Use BigQuery to:

* Join customer and transaction/support data
* Derive metrics (churn, fraud likelihood)
* Save output to curated zone

Example query:

CREATE OR REPLACE TABLE your\_dataset.curated\_churn\_features AS  
SELECT  
 c.customer\_id,  
 c.age,  
 c.credit\_score,  
 c.monthly\_balance,  
 COUNT(t.transaction\_id) AS transaction\_count,  
 AVG(t.amount) AS avg\_transaction\_amount,  
 COUNT(DISTINCT s.log\_id) AS support\_issues,  
 c.is\_churned  
FROM your\_dataset.customers c  
LEFT JOIN your\_dataset.transactions t ON c.customer\_id = t.customer\_id  
LEFT JOIN your\_dataset.support\_logs s ON c.customer\_id = s.customer\_id  
GROUP BY c.customer\_id, c.age, c.credit\_score, c.monthly\_balance, c.is\_churned;

## 5. 🔐 **Apply Governance in Dataplex**

* Add metadata tags (e.g. PII, Financial, Transactional)
* Assign IAM roles (e.g. roles/dataplex.metadataReader, roles/bigquery.dataViewer)
* Ensure access to curated and analytics zones is restricted to analysts and data scientists

## 6. 🤖 **Use Generative AI for Insights or Dashboards**

### a. BigQuery Data Insights with Gemini (Preview)

* Open BigQuery Console
* Click the **“Chat” or “Help me analyze”** option in the query editor
* Ask questions like:
  + “Summarise churn trends by age group”
  + “Generate a SQL query to detect high-risk customers”

### b. Use Looker Studio + Gemini

* Connect BigQuery tables
* Let Gemini suggest dashboard visuals or write calculated fields

### c. Vertex AI Notebooks + GenAI

* Launch a managed JupyterLab environment
* Use pandas and palm or gemini-pro APIs to auto-generate narratives, chart commentary, or decision summaries

## ✅ Outcome

By the end of this workflow, the project should have:

* A Dataplex-governed lake (3 zones)
* Tagged and registered assets
* Transformed and analyzable BigQuery data
* Optional ML model outputs or dashboards
* Generated narratives or insight summaries using GenAI

Would you like an accompanying .sh script and .ipynb notebook to go with this?