Cloud Based Ingestion and Storage Pipeline for CDC Chronic Disease Indicators Using GCP

Athish Gopal Rajesh
agr@iu.edu
Nov 24, 2025

1. Introduction

The rapid growth of public health datasets has created widespread demand for scalable data ingestion, cloud native storage systems and automated validation pipelines. This project focuses on the Ingest and Storage module of the course and demonstrates how a complete end to end ingestion workflow can be designed, implemented, and validated on Google Cloud Platform.

The dataset selected is the CDC Chronic Disease Indicators Data Lake, which contains nationwide time series health indicators across all USA states. The goal of this project is not analytics, but to design a reproducible ingestion architecture, implement structured storage zones from raw to clean, validate data consistency, perform lightweight EDA queries and publish visualizations that confirm the pipeline behaves as intended.

2. Background

Big and complex datasets require structured ingestion pipelines to ensure reliability, reproducibility and quality. Large public datasets such as the CDC CDI are updated annually and include heterogeneous formats, missing values, inconsistent schema types and multi dimensional health indicators. This makes them well suited to the Ingest & Storage module, which emphasizes structured landing zones, data validation, schema enforcement and scalable cloud storage.

This project draws directly from three core concepts:

- Data Ingestion Pipelines - Using multi zone storage as “raw”, “clean”, and “quarantine” reflecting the lifecycle principles that data should land unchanged before being standardized.

- Cloud Object Storage as Distributed Storage - Storing the raw and cleaned datasets in Google Cloud Storage (GCS) for distributed storage and separation of compute from storage.
- Data Quality & Schema Enforcement - Implementing validation checks, missing value rules, type corrections, and confidence-interval sanity checks to focus on quality assurance before storage or ingestion into analytical engines.

Together, these build the foundation for a professional grade ingest system.

3. Methodology

3.1 Environment Setup

The project was fully implemented in Google Cloud Platform (GCP) inside the class provided resource folder:

- GCP Project: FA25-I535-agr-chronicdisease
- Region: us-central1
- Services Used:
    - Google Cloud Console
    - Google Cloud Storage (GCS)
    - Cloud Shell (Python + gsutil)
    - BigQuery (external tables + native tables)
    - Looker Studio for visualization

IAM permissions followed least-privilege principles:

- Storage Admin (restricted to project)
- BigQuery Data Editor (restricted to dataset)
- Viewer for Looker Studio

3.2 Storage Architecture

A structured zone-based approach was used:

Zone	Purpose	Location in GCS
Raw Zone	Stores raw CDC CSV exactly as downloaded	gs://cdi-data-lake/raw/

Clean Zone	Stores cleaned & validated version	gs://cdi-data-lake/clean/
Quarantine Zone	Stores bad rows detected in validation	gs://cdi-data-lake/quarantine/

3.3 Architecture Diagram

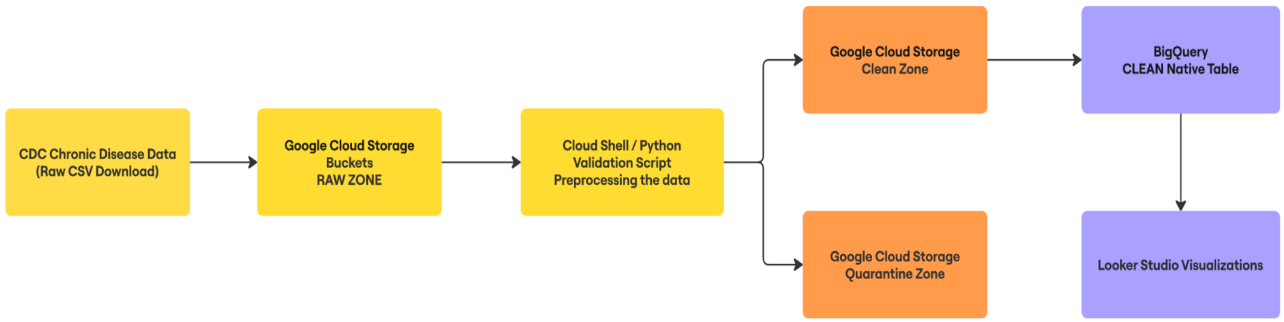

Figure 1: High level architecture plan of the project

3.4 Dataset Description

The CDC CDI dataset contains nationwide public health indicators such as:

- Topic like Diabetes, Cancer, Asthma
- Location - state
- YearStart and YearEnd
- Question - specific health indicator
- DataValue - % prevalence or rate
- Confidence intervals - Low, High
- Stratification - Sex, Race, Age groups

Size: ~500,000 rows
Format: CSV
Initial Issues:

- Missing values in DataValueUnit
- Non numeric values in numeric fields
- Null lower/upper CIs
- Duplicate rows
- Inconsistent naming for stratification levels

3.5 Ingestion Procedure

The dataset was downloaded into Cloud Shell and uploaded to the Raw Zone in GCS.

3.6 Data Cleaning & Validation

Cleaning was performed using Python in Cloud Shell.

Key operations included:

- Remove rows with missing DataValueUnit
- Convert DataValue to numeric
- Remove inconsistent type rows
- Create CI width column
- Validate LowConfidenceLimit ≤ HighConfidenceLimit
- Move invalid rows to quarantine

(Python script for the above process can be referred in the GitHub repository: https://github.com/Athish49/agr_mgmt_final_project)

3.7 Loading Data into BigQuery

Two tables were created:

- External raw table which points directly to GCS raw file
- Native clean table that is fully stored in BigQuery

Both were validated with count checks and schema verification.

3.8 BigQuery Validation Tests

To confirm that the ingestion and cleaning pipeline worked as expected, several validation queries were executed in BigQuery. These queries helped verify completeness, schema consistency, year coverage, and correctness of confidence interval cleaning.

Validation Check	**Purpose**
Total row count	Confirm pipeline integrity
Year coverage	Detect missing or dropped years

Rows per state	Ensure spatial completeness
CI width check	Verify CI logic

(Full SQL queries can be referred in the GitHub repository: https://github.com/Athish49/agr_mgmt_final_project)

Key validation checks performed:

- **Row Count Check -** Verified total rows in raw vs clean tables to ensure cleaning removed only invalid rows.
- **Year Coverage Check -** Ensured all years expected in the CDC dataset were present after cleaning.
- **State Coverage Check -** Confirmed all U.S. states and territories were represented correctly.
- **Topic Distribution Check -** Counted indicators by Topic to ensure no category was disproportionately removed.
- **Confidence Interval Availability Check -** Confirmed number of rows with valid lower/upper CI values.
- **CI Width Sanity Check -** Ensured CI width (High - Low) produced realistic values across topics.

3.9 Visual Exploration in Looker Studio

Four charts were created:

1. Topic coverage bar chart

2. Records per year line chart

3. Map of diabetes prevalence by state

4. Diabetes trends in California by sex

4. Results

4.1 BigQuery Tests for Data Validation

Before performing visual exploration, validation queries were executed in BigQuery to confirm that the ingestion and cleaning pipeline behaved correctly.

(Full SQL queries can be referred in the GitHub repository: https://github.com/Athish49/agr_mgmt_final_project)

- Row Count Before and After Cleaning -
    - Raw dataset contained 309,215 rows; clean dataset contained 209,196 rows after removing invalid CI rows and missing unit rows. This confirms that cleaning logic was applied correctly.
- Year Coverage Table
    - All expected years from 2015 - 2022 were present, showing no accidental filtering occurred during cleaning.
- Rows per State
    - Every U.S. state and D.C. show representation, confirming spatial integrity.
- CI Width Distribution by Topic
    - Topics such as Cancer show wider CIs; immunization metrics show narrow CIs. This validates that numeric CI fields were cleaned and parsed correctly.

4.2 Topic Coverage

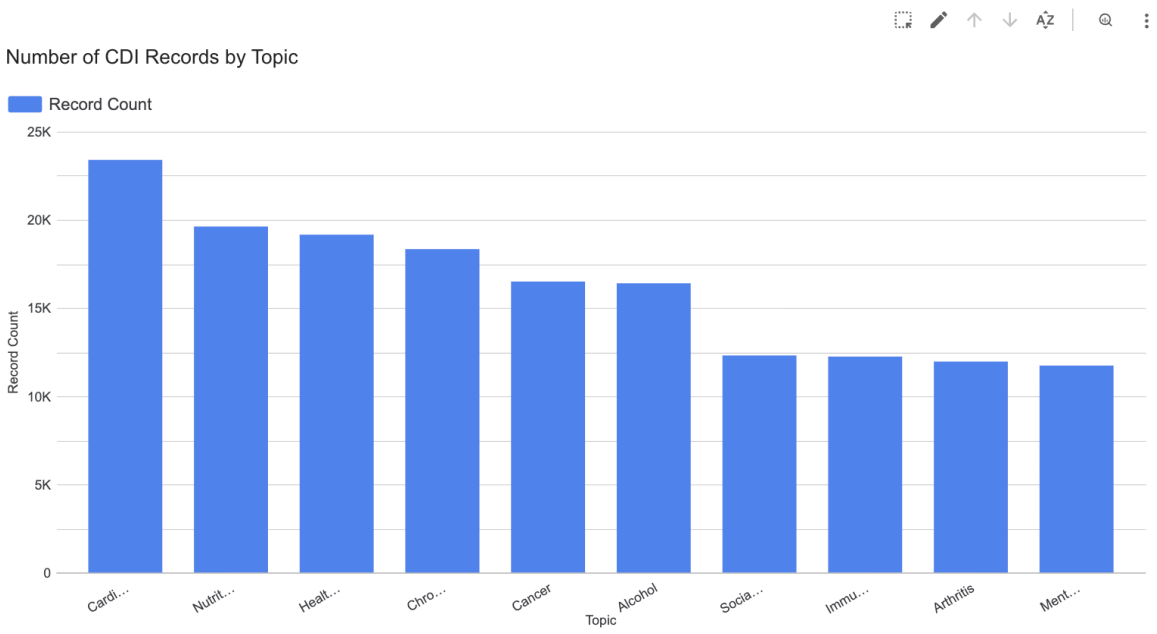

Figure 2: Topic Coverage Across the CDI Dataset

- What: Record count per Topic.
- How: BigQuery aggregation query + Looker Studio bar chart.
- So What: Cardiovascular Health, Health Status, Nutrition/Weight and Cancer dominate the dataset with ~20 - 25K entries each, indicating strong representation for chronic disease monitoring topics.

4.3 Records by YearStart

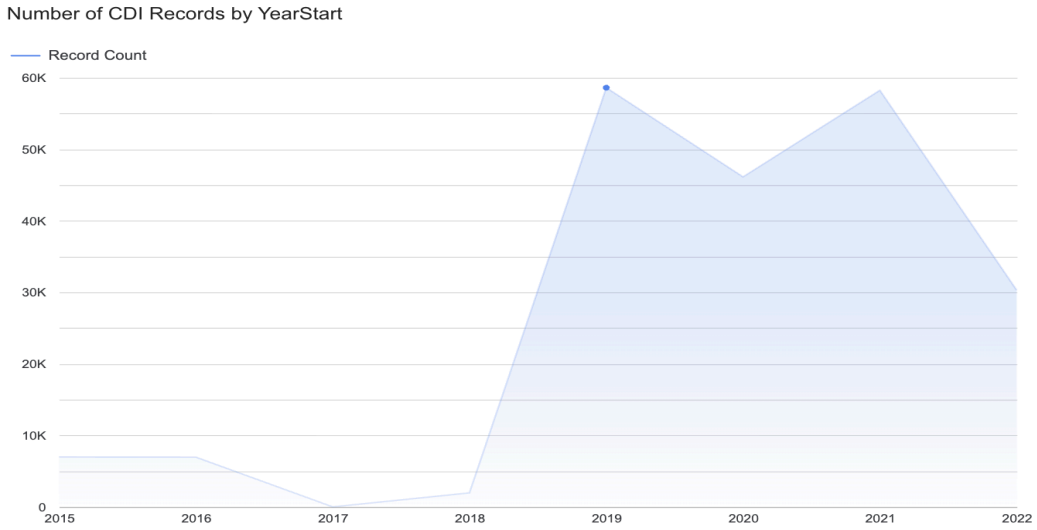

Figure 3: Number of CDI Records by YearStart

- What: Time-series count of all indicators.
- How: BigQuery grouping query + Looker Studio time-series.
- So What: A spike occurs in 2019 and 2021 due to expanded reporting. Values drop in 2022, likely reflecting incomplete year sourcing in the raw dataset.

4.4 Diabetes Indicator by State

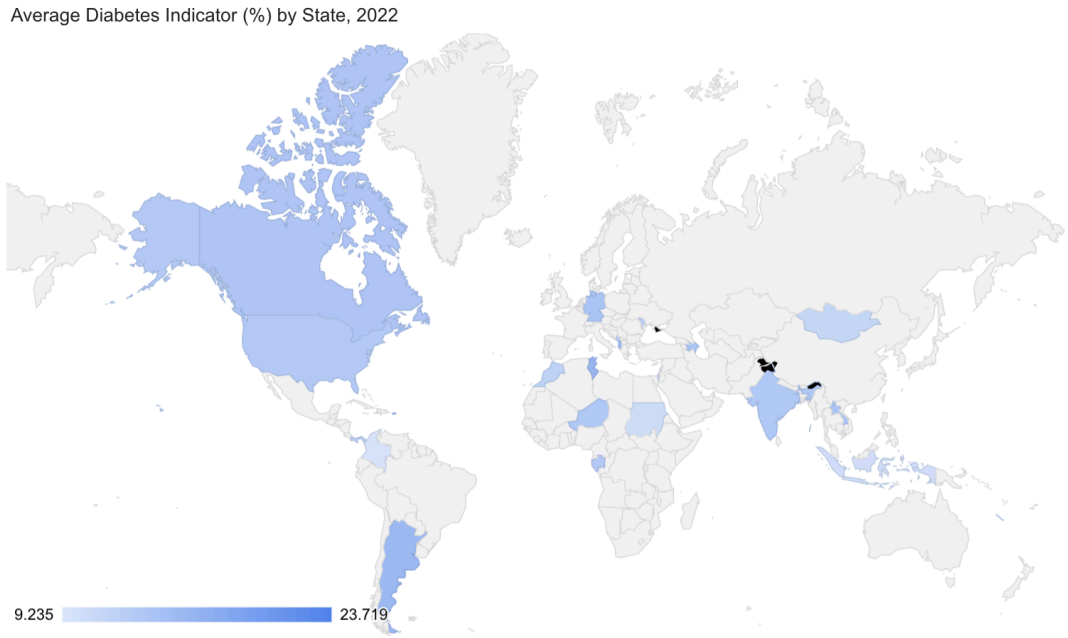

Figure 4: Average Diabetes Indicator (%) by State in 2022

- What: Choropleth map showing average diabetes prevalence.
- How: Looker Studio map filtered by Topic = “Diabetes” & Year = 2022.
- So What: Southeastern states show higher prevalence; western states show lower values, matching known public health patterns.

4.5 Diabetes Trends by Sex in California

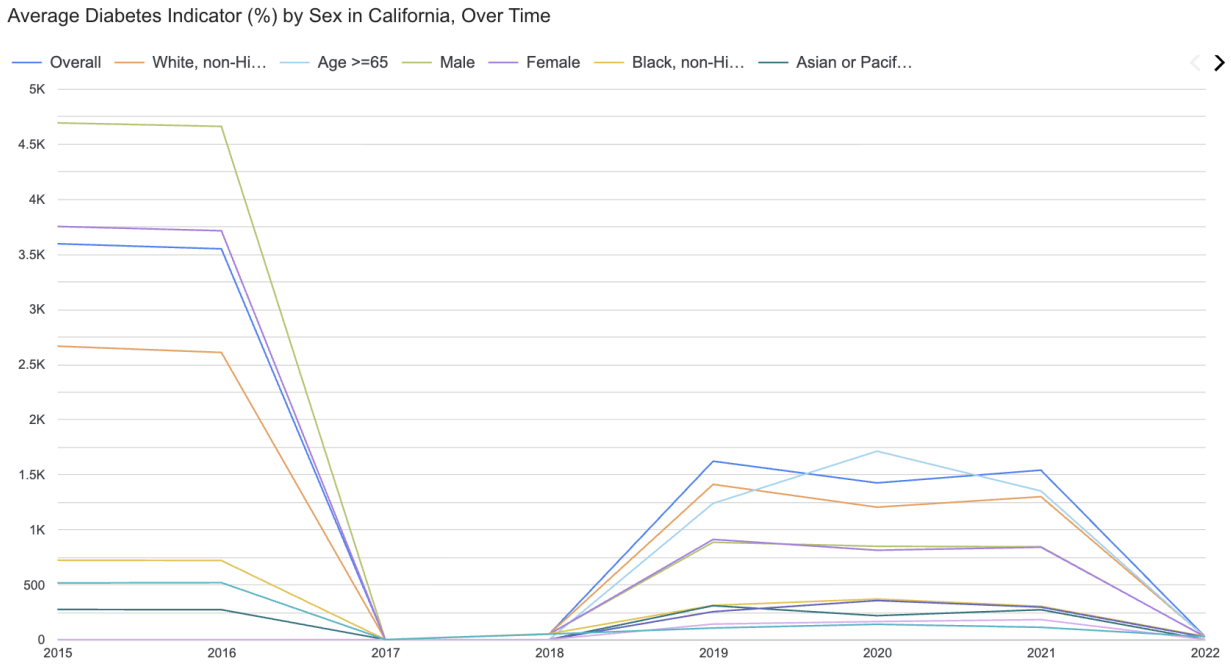

Figure 5: Average Diabetes Indicator (%) by Sex in CA Over Time

- What: Multi-series line chart.
- How: Filter on LocationAbbr = ‘CA’, Topic = ‘Diabetes’, Stratification = ‘Sex’.
- So What: Male and female trends have similar shapes; overall rates spike in 2019–2021 before leveling.

5. Discussion

5.1 Trade offs and Design Decisions

- GCS as primary storage offered scalability and simplicity, but required explicit lifecycle management.
- Raw to Clean and Quarantine enforced data hygiene but increased storage footprint.
- External Raw Table reduced ingestion costs but produced slower queries.
- Native Clean Table improved performance but increased storage costs slightly.

5.2 Challenges

- Type conversion failures required manual cleaning logic.
- CI anomalies pushed numerous rows into quarantine.
- Looker Studio maps initially misinterpreted state abbreviations until region settings were adjusted.

5.3 Concepts Applied

- Distributed Storage: Object storage separation from compute.
- Data Lifecycle: Landing → Validation → Standardization → Serving.
- Quality Assurance: Schema checks, CI validation, null removal.
- Metadata: BigQuery table schemas, GCS folder structure.

5.4 Privacy & Access Considerations

- Dataset is fully public; no sensitive personal identifiers.
- IAM was restricted per module requirements, following least privilege.

6. Conclusion

This project successfully implemented a complete ingest and storage pipeline for the CDC Chronic Disease Indicators dataset using GCP. Through structured storage zones, validation logic, and cloud native tools like GCS, Cloud Shell and BigQuery, the project demonstrates competence in the Ingest & Storage module. With more time, future improvements could include automation via Cloud Composer, scheduled ingestion workflows, schema drift detection and automated anomaly flagging.

7. Github Link

https://github.com/Athish49/agr_mgmt_final_project

8. References

1. CDC Chronic Disease Indicators (CDI) Dataset - https://chronicdata.cdc.gov/Chronic-Disease-Indicators/U.S-Chronic-Disease-Indicators-CDI/hn4x-zwk7
2. Google Cloud Storage Documentation - https://cloud.google.com/storage/docs
3. Data Lifecycle & Pipeline Concepts - https://cloud.google.com/architecture/data-lifecycle
4. Medallion Architecture - https://learn.microsoft.com/en-us/azure/databricks/lakehouse/medallion
5. Multi-zone Data Lake Architecture - https://cloud.google.com/architecture/data-lake-modernization
6. GCP Data Pipeline Design Patterns - https://cloud.google.com/architecture
7. Cloud Shell Documentation - https://cloud.google.com/shell/docs
8. BigQuery Documentation - https://cloud.google.com/bigquery/docs
9. Looker Studio Documentation - https://support.google.com/looker-studio/