COVID-19 Data Analysis and Insights using AWS Glue & Redshift Serverless

Objective:

To build a scalable ETL (Extract, Transform, Load) pipeline using AWS services for analyzing the global COVID-19 dataset. The project involved cleaning the dataset, storing it in Amazon S3, transforming it via AWS Glue, and querying insights using Redshift Serverless.

To design and implement an ETL pipeline on AWS that:

- Ingests raw COVID-19 data from S3.
- Cleans and transforms it using AWS Glue.
- Analyses the cleaned dataset using SQL queries in Amazon Redshift Serverless.
- Derives actionable insights such as cases, fatalities, vaccination rates, and more.

Architecture Overview:

Component	Role
Amazon S3	• Data storage (CSV source & Parquet output) • Storage for raw and transformed datasets
AWS Glue	ETL job: extract from S3, transform, and load to Redshift
AWS Glue Data Crawler	Auto-detect schema and create tables in Data Catalog
Amazon Redshift Serverless	SQL querying engine for analysis
Redshift Query Editor v2	UI for executing queries & visualizing results

Steps Performed:

1. Raw Data Setup

- Uploaded owid-covid-data.csv (from Our World In Data) into S3 under a bucket named my-covid-etl-bucket.
- Source: OWID COVID-19 dataset

2. Glue Crawler Setup

- A Glue Crawler was created to crawl the CSV in S3.

- It created a table covid_data_raw in database covid-db inside the AWS Glue Data Catalog.

3. ETL Job in AWS Glue

- A Glue Spark job was written to clean and transform the data:
  - Removed rows with missing location or iso_code
  - Removed regional aggregate rows (iso_code starting with OWID_)
  - Converted date formats
  - Filtered unnecessary columns
  - Wrote output in **Parquet** format for better Redshift performance
- The final output was saved back to S3 as **Parquet**, and crawler updated the schema in covid_data_cleaned.

4. Redshift External Schema

- Set up Redshift Serverless with a workgroup covid-etl-wg.
- Connected it to AWS Glue Catalog via awsdatacatalog external schema.
- Loaded and tested data from the external schema and table covid_data_cleaned.

Issues Faced & Resolutions:

Issue	**Resolution**
Multiple similar database/schema names (e.g., covid-db, covid_db)	Used SVV_EXTERNAL_SCHEMAS to inspect actual mappings. Decided to retain only covid-db and clean up unnecessary ones.
Aggregates like "World", "Europe" appearing as countries	Applied filter iso_code NOT LIKE 'OWID_%' to eliminate non-country entries from all queries.
Querying SHOW TABLES IN ... gave errors because syntax was not supported for external schema in Redshift	Used metadata tables like SVV_EXTERNAL_SCHEMAS to verify schema presence
Parquet data not visible because Redshift Glue integration required crawler re-run after job	Triggered crawler after successful ETL job output

---

SQL Queries for Insights (Final Cleaned Dataset)

All queries run on: covid_db.covid_data_cleaned

Query 1: Top 10 Countries with Highest Total COVID-19 Cases

```
-- Top 10 countries with the highest total COVID-19 cases
-- Excludes regional aggregates like "World", "Asia", etc.
SELECT
    location AS country,
    MAX(total_cases) AS max_total_cases
FROM covid_db.covid_data_cleaned
WHERE location IS NOT NULL
  AND iso_code IS NOT NULL
  AND iso_code NOT LIKE 'OWID_%'
GROUP BY location
ORDER BY max_total_cases DESC
LIMIT 10;
```

Query 2: Countries with Highest Death-to-Case Ratio

```
-- Top 10 countries with the highest death-to-case ratio (fatality rate)
SELECT
    location AS country,
    MAX(total_deaths) AS total_deaths,
    MAX(total_cases) AS total_cases,
    ROUND(MAX(total_deaths)::numeric / NULLIF(MAX(total_cases), 0), 4) AS death_case_ratio
FROM covid_db.covid_data_cleaned
WHERE total_deaths IS NOT NULL
  AND total_cases IS NOT NULL
  AND iso_code IS NOT NULL
  AND iso_code NOT LIKE 'OWID_%'
GROUP BY location
HAVING MAX(total_cases) > 10000  -- avoid skew from very small countries
```

```
ORDER BY death_case_ratio DESC
LIMIT 10;
```

Query 3: Countries with Highest Single-Day Spike in New Cases

```
-- Top 10 countries with the highest single-day spike in new cases
SELECT
    location AS country,
    MAX(new_cases) AS peak_new_cases
FROM covid_db.covid_data_cleaned
WHERE new_cases IS NOT NULL
  AND iso_code IS NOT NULL
  AND iso_code NOT LIKE 'OWID_%'
GROUP BY location
ORDER BY peak_new_cases DESC
LIMIT 10;
```

Query 4: Highest 7-Day Rolling Average of New Cases

```
-- Top 10 countries with the highest 7-day rolling average of new cases
SELECT
    location AS country,
    MAX(rolling_avg_7d_cases) AS peak_7d_avg_cases
FROM (
    SELECT
        location,
        date,
        AVG(new_cases) OVER (PARTITION BY location ORDER BY date ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_7d_cases
    FROM covid_db.covid_data_cleaned
    WHERE location IS NOT NULL
    AND iso_code NOT LIKE 'OWID_%'
) subquery
GROUP BY location
ORDER BY peak_7d_avg_cases DESC
LIMIT 10;
```

Query 5: Highest Cases Per 100 People

```
-- Top 10 countries by total COVID-19 cases per 100 people
SELECT location AS country,
       MAX(total_cases) AS total_cases,
       MAX(population) AS total_population,
       ROUND(MAX(total_cases) * 100.0 / NULLIF(MAX(population), 0), 2) AS cases_per_100
FROM covid_db.covid_data_cleaned
WHERE location IS NOT NULL
  AND iso_code NOT LIKE 'OWID_%'
GROUP BY location
ORDER BY cases_per_100 DESC
LIMIT 10;
```

Outcome & Learnings

- Demonstrated full end-to-end ETL on AWS with serverless architecture.
- Tackled real data issues like missing values, wrong schema linkage, and non-standard records.
- Gained hands-on with:
  - S3 file management
  - Glue Crawler + Job configuration
  - Redshift External Schema setup
  - Analytical SQL for time-series and aggregates
- Produced real-world COVID-19 country-level insights for health analytics and reporting.

Future Scope

- Automate the pipeline using AWS **Step Functions** or **Glue Workflows**
- Add a **QuickSight Dashboard** for visual insights
- Set up **scheduled crawlers** and **incremental jobs** for daily COVID updates