Used cars platform

Data Engineering project

Abduallah Hussien 2024 Ready Talent

Abstract

This data engineering project aims to create a robust and scalable data pipeline that extracts raw data from a Google Cloud Storage (GCS) bucket, transforms it into a star schema in BigQuery, and then visualizes the data in a Power BI dashboard. The project utilizes Apache Airflow for orchestrating the data ingestion process, DBT for implementing the dimensional modeling, and Power BI for creating the interactive dashboard. This documentation provides a comprehensive overview of the project's architecture, implementation details, and user manual, ensuring seamless understanding and adoption of the developed solution.

Table of Contents

1. Introduction
    - Project Objectives
    - Data Sources
2. Architecture
    - Overall System Design
    - Data Pipeline
3. Implementation
    - Data Ingestion with Airflow
        - GCS to BigQuery
    - Data Transformation with DBT
        - Dimensional Modeling
    - Data Visualization with Power BI
4. User Manual
    - Prerequisites
    - Setup and Configuration
    - Running the Data Pipeline
    - Viewing the Power BI Dashboard
5. Future Enhancements
6. Conclusion

Introduction

Project Objectives

The primary objectives of this data engineering project are:

- **Data Ingestion**: Automatically transfer raw data from a GCS bucket to a BigQuery dataset using Apache Airflow.
- **Data Transformation**: Convert the raw data into a star schema dimensional model using DBT to improve data querying and analysis.
- **Data Modeling:** The transformed data is loaded into the BigQuery star schema tables.
- **Data Visualization**: Leverage the star schema model to create an interactive and informative Power BI dashboard for data analysis and reporting.

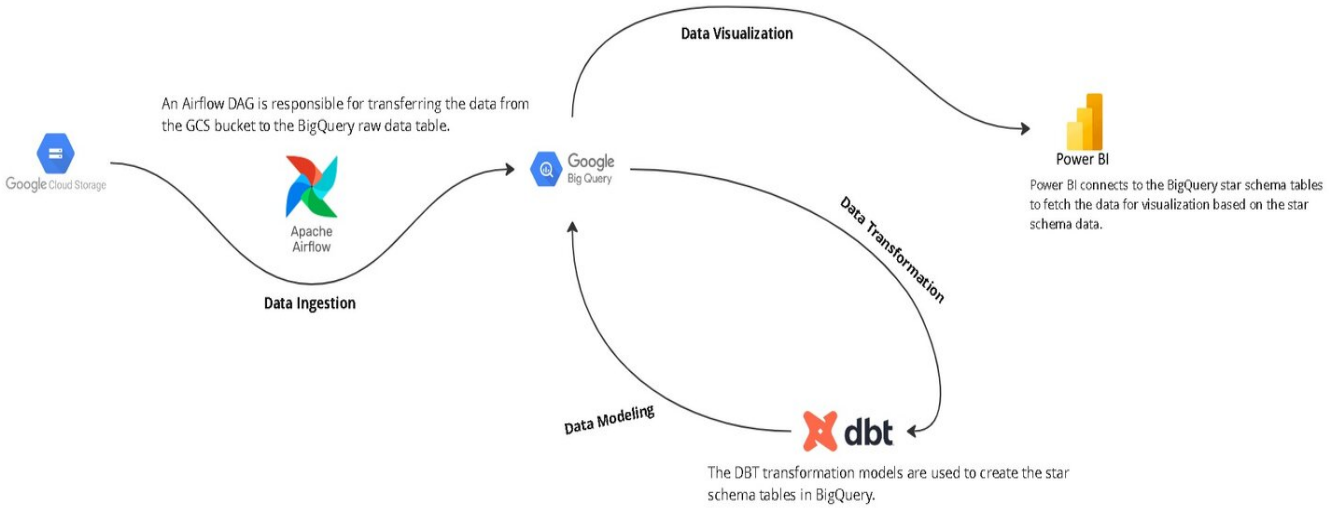
Data Sources

The data for this project is stored in a GCS bucket and consists of CSV file containing used cars information. The specific data sources and their schema will be detailed in the Architecture section.

Architecture

Overall System Design

The overall system design for this data engineering project follows a typical data pipeline architecture, as shown in the diagram below:



The key components of the system are:

1. **GCS Bucket**: Stores the raw data files in CSV format.
2. **Apache Airflow**: Orchestrates the data ingestion process from the GCS bucket to BigQuery.
3. **BigQuery**: Serves as the data warehouse, storing both the raw data and the transformed star schema model.
4. **DBT**: Implements the dimensional modeling and data transformation logic to convert the raw data into a star schema.
5. **Power BI**: Consumes the star schema model to create the interactive data visualization dashboard.

Data Pipeline

The data pipeline for this project consists of the following steps:

1. **Data Ingestion**: Apache Airflow is used to transfer the raw data files from the GCS bucket to a landing zone dataset in BigQuery.
2. **Data Transformation**: DBT is employed to transform the raw data into a star schema model, which is stored in a new BigQuery dataset.
3. **Data Visualization**: The star schema model in BigQuery is connected to Power BI, where an interactive dashboard is created to visualize the data.

The detailed implementation of each step is covered in the Implementation section.

Implementation

Data Ingestion with Airflow

GCS to BigQuery

The data ingestion process using Apache Airflow involves the following steps:

1. **Define the DAG**: Create an Airflow DAG (Directed Acyclic Graph) that represents the data transfer workflow.
2. **Specify the Tasks**: Define the individual tasks within the DAG, such as extracting data from GCS, loading it into BigQuery, and validating the data.
3. **Implement the Task Logic**: Write the Python code to execute the tasks, including the GCS to BigQuery data transfer.

Here's a sample of the Airflow DAG code:

```python
from datetime import datetime
from airflow import DAG
from airflow.operators.empty import EmptyOperator
from airflow.providers.google.cloud.transfers.gcs_to_bigquery import GCSToBigQueryOperator

# Dags initialization
dag = DAG(
    dag_id="Abduallah_from_GCStoBQ",
    description="Transfering_from_GCStoBQ",
    schedule_interval=None,
    start_date=datetime(2021, 1, 1),
    catchup=False,
)

start_task = EmptyOperator(task_id="start_task", dag=dag)

load_csv = GCSToBigQueryOperator(
    task_id="gcs_to_bigquery",
    bucket="ready-project-dataset",
    source_format='CSV',
    skip_leading_rows=1,
    field_delimiter=',',
    max_bad_records=100,
    source_objects=[source_objects_path],
    destination_project_dataset_table=f"{destination_project}.{destination_dataset}.{destination_table}",
    schema_fields = schema_fields,
    write_disposition="WRITE_TRUNCATE",
    create_disposition= "CREATE_IF_NEEDED"

end_task = EmptyOperator(task_id="end_task", dag=dag)

start_task >> load_csv >> end_task
```

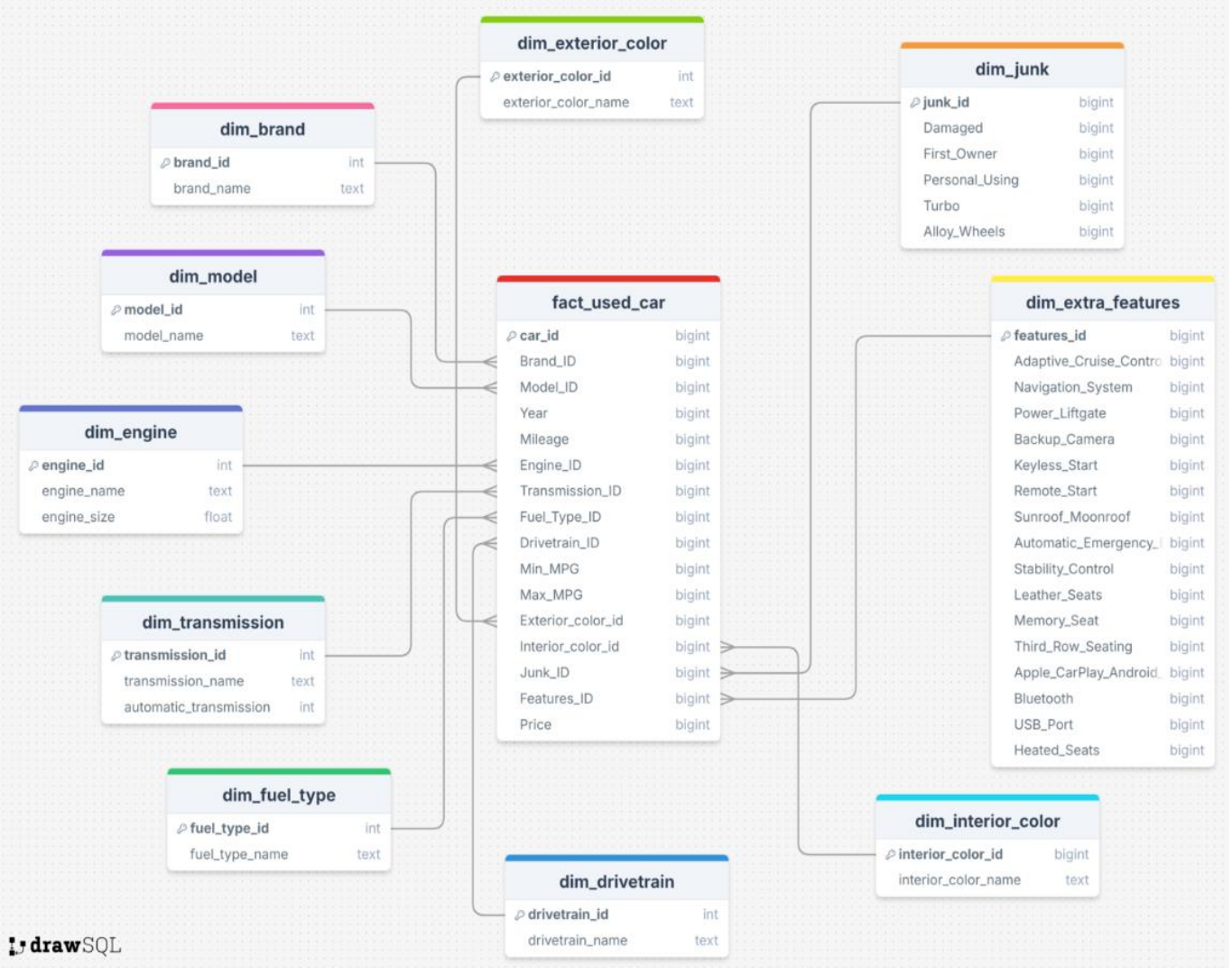
Data Transformation with DBT

Dimensional Modeling

To transform the raw data into a star schema model, the following steps are performed using DBT:

1. **Define the Data Model**: Identify the fact and dimension tables required for the star schema, based on the analysis of the raw data.
2. **Implement the Transformations**: Write SQL scripts to perform the necessary data transformations and load the data into the fact and dimension tables.
3. **Test and Validate**: Implement data quality checks and unit tests to ensure the correctness of the transformed data.
4. **Deploy the Model**: Deploy the DBT project to the BigQuery dataset, creating the star schema tables.

Here's the entity relationship diagram **(**ERD):



Data Visualization with Power BI

After the star schema model is available in BigQuery, the final step is to create the Power BI dashboard:

1. **Connect to BigQuery**: In Power BI, establish a connection to the BigQuery dataset containing the star schema tables.
2. **Design the Dashboard**: Create various visualizations (e.g., charts, tables, KPIs) to effectively present the data and provide insights.
3. **Publish and Share**: Publish the Power BI report and share it with the relevant stakeholders.

Here's a screenshot of a Power BI dashboard:

User Manual

Prerequisites

To use this data engineering solution, you will need the following:

- Access to a Google Cloud Platform (GCP) project with the necessary permissions for Airflow, BigQuery, and GCS.
- A GCS bucket to store the raw data files.
- Download Power BI desktop to view the dashboard.

Setup and Configuration

1. **Configure Airflow**: Set up an Apache Airflow instance and configure the necessary connections to GCS and BigQuery.
2. **Prepare the Data**: Upload the raw data files in CSV format to the designated GCS bucket.
3. **Set up DBT**: Create a DBT project and connect it to the BigQuery dataset.
4. **Connect Power BI**: Establish a connection between Power BI and the BigQuery dataset containing the star schema model.

Running the Data Pipeline

1. **Initiate the Airflow DAG:** Start the Airflow DAG to trigger the data ingestion process from GCS to BigQuery.
2. **Monitor the Dataform (DBT) Deployment:** Ensure the DBT transformations are successfully deployed, creating the star schema model in BigQuery.
3. **Refresh the Power BI Dashboard:** In Power BI, refresh the dataset connection to fetch the latest data from BigQuery and update the visualizations.

Viewing the Power BI Dashboard

1. **Open the Power BI Report:** Access the published Power BI report and navigate to the dashboard.
2. **Explore the Visualizations:** Interact with the various charts, tables, and KPIs to analyze the data and gain insights**.**
3. **Filter and Drill-down:** Use the available filters and drill-down capabilities to explore the data at different levels of granularity.

Future Enhancements

As the project evolves, the following future enhancements may be considered:

- Configure the DAG to run on a scheduled interval (e.g., daily, weekly) to keep the data up-to-date.
- Adding the staging stage (3nf schema) before transferring directly to star schema.
- Integration with additional data sources (e.g., external APIs, on-premises databases)
- Exploring advanced analytics and machine learning capabilities
- Improved monitoring and alerting with custom dashboards and notifications
- Publish the dashboard and schedule refreshment of the dashboard to keep the data up-to-date.

Conclusion

This data engineering project demonstrates a robust and scalable solution for ingesting, transforming, and visualizing data from a GCS bucket. By leveraging Apache Airflow, DBT and Power BI, the project delivers a comprehensive data pipeline that efficiently processes raw data, transforms it into a star schema model, and presents insightful visualizations for data analysis and reporting. The detailed documentation provided ensures seamless understanding and implementation of the project, allowing for easy maintenance and future enhancements.

****Please let me know if you have any further questions or require additional information.****