UDACITY

< Return to Classroom

DISCUSS ON STUDENT HUB

Data Engineering Capstone

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Great job, you are ready to go! 💯Clearly, you have acquired all the important concepts from this project. Wish you all the best for the upcoming projects! 🤞

Tip: If you are interested in knowing more about data modeling in the real world, please read this series of posts and this experience sharing

Write Up

The write up includes an outline of the steps taken in the project.
The purpose of the final data model is made explicit.

Good job on explaining the goal for this data modeling.👍Using Spark to build an ETL is a good example of this project! Also, the logic behind this star schema is clearly explained.

Introduction

Hi, everyone. Welcome to my data engineering nanodegree capstone project. This project aims to allow me to utilize what I learned from the nanodegree. In short, we will build a data pipeline for the `I94 immigration` data and combine them with several useful other data sources such as `world temperature` and `US city demographic` data. This project aims to show you the end-to-end process of building a data pipeline from several data sources.

Benefits

We can answer many exciting business questions with this dataset. Here are examples

1. What is the current state of immigrant people in the US?
2. What are immigration patterns compared between different times (the 80s, 90s, ..., 2020)?
3. What factors correlate to the immigration pattern?
4. What is the number of immigration people in the next following month?

Those answers can help US immigration department to design a better policy to take care of the immigrant people. Also, not only from the data analytics and business side, you will find the data pipeline architecture and ETL code for running this project. This project will be a good start for anyone looking for where to start their data engineer journey.

The write up describes a logical approach to this project under the following scenarios:

- **The data was increased by 100x.**
- **The pipelines would be run on a daily basis by 7 am every day.**
- **The database needed to be accessed by 100+ people.**

Good answers for each question, well done! But I would encourage you to think a little more, for example, when the data was increased by 100x, do you store the data in the same way with Spark? What could be the better partition key in this case? If your project is heavy on reading over writing, how do you store the data in a way to meet this requirement? What if the requirement is heavy on writing instead?
Tip:

- If you are interested in knowing more about the heavy-read system design, I would suggest that you read this tutorial about Netflix system.
- I would suggest that you read this experience sharing about the lesson learned when implementing data pipeline

- The data was increased by 100x.
    - Because we leverage the power of spark. There is no need to worry about the scaling size of the underlying computation engine. In case we reach the limit, we can increase the cluster size that the spark is running on. Spark also works in a distributed way, so horizontal scaling is always an option to go for.
- The data populates a dashboard that must be updated on a daily basis by 7 am every day.
    - We can meet this requirement with the SLA option provided by Airflow. This feature will guarantee that the system should populate the data before 7 am every day. In case your task failed, you can fix the problem by shifting the start ETL time earlier or increasing the spark's computation power.
- The database needed to be accessed by 100+ people.

- We can store the data in any data warehouse options (e.g. redshift) and let them access our data. The underlying data format can still be a `delta` format.

The choice of tools, technologies, and data model are justified well.

Good job on providing your reasonings about each tool you use and justifying the data model well👍

Technology

There will be 2 main components in any data pipeline that we need to selectively choose for building the whole project. `storage format`, `computation engine`. There are many tools and technology out there, but here is what I decided to use in this capstone project.

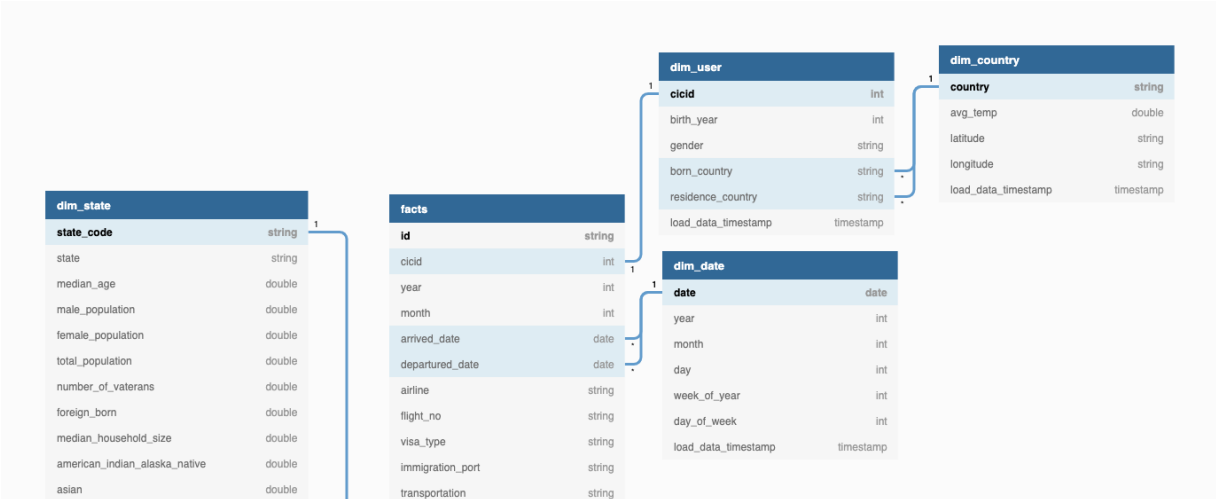
Delta Lake `storage format`

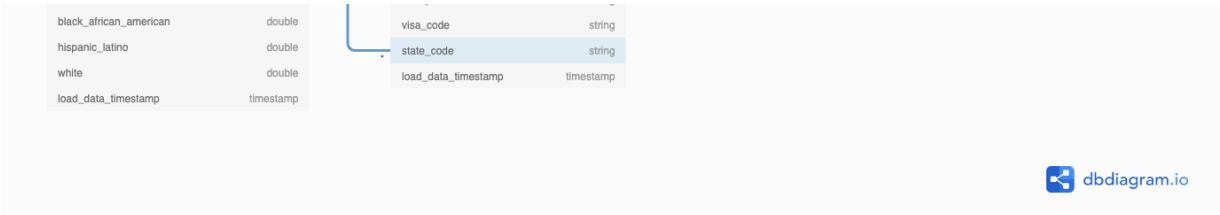
Delta Lake is an open source storage layer that brings reliability to data lakes. Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing. Delta Lake runs on top of your existing data lake and is fully compatible with Apache Spark APIs.

In short, delta lake is an updated version of parquet format. The development team brings many valuable features to fix the problem of storing data in NoSQL format. For example, I decided to use the delta lake format in this project because it provides the `UPSERT` ability compared to parquet that you have to code by yourself. It helps heavy-lifting unnecessary thing and help you focus on only the data. Also, there are other valuable features such as ACID transactions and metadata handling.

Apache Spark `computation engine`

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python, and R and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools, including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.





Execution

All coding scripts have an intuitive, easy-to-follow structure with code separated into logical functions. Naming for variables and functions follows the PEP8 style guidelines. The code should run without errors.

The code is clean and easy to follow👍All functions have their own docstrings for better understanding.

The project includes at least two data quality checks.

You correctly design the required data quality checks.💯

```
from pyspark.sql import SparkSession
spark = SparkSession.builder\
    .config("spark.jars.packages","saurfang:spark-sas7bdat:2.0.0-s_2.11,io.delta:delta-core_2.11:0.6.1")\
    .config("spark.sql.extensions", "io.delta.sql.DeltaSparkSessionExtension")\
    .config("spark.sql.catalog.spark_catalog", "org.apache.spark.sql.delta.catalog.DeltaCatalog")\
    .enableHiveSupport().getOrCreate()
```

```
def check_data(path):
    df = spark.read.format('delta').load(path)
    return df.count(), df.show(3)
```

```
check_data('./output/staging_fact')
```

```
+-------+-----+------+------+------+-------+-------+-------+-------+-------+------+-------+-----+--------
---+-------+-------+-------+-------+-------+--------+------+------+-------+--------------+-----+--------+
--------+
|  cicid|i94yr|i94mon|i94cit|i94res|i94port|arrdate|i94mode|i94addr|depdate|i94bir|i94visa|count|dtadfile
cup|entdepa|entdepd|entdepu|matflag|biryear| dtaddto|gender|insnum|airline|        admnum|fltno|visatype|
imestamp|
+-------+-----+------+------+------+-------+-------+-------+-------+-------+------+-------+-----+--------
---+-------+-------+-------+-------+-------+--------+------+------+-------+--------------+-----+--------+
--------+
|4664259| 2016|     5| 213.0| 213.0|    HOU|20597.0|    1.0|     TX|20621.0|  55.0|    2.0|  1.0|20160523
ull|      G|      O|   null|      M|   1961|11222016|     M|  null|     QR|9.718120793E10|00713|      B2|
7:18:...|
|4664260| 2016|     5| 213.0| 213.0|    HOU|20597.0|    1.0|     TX|20621.0|  51.0|    2.0|  1.0|20160523
ull|      G|      O|   null|      M|   1965|11222016|     M|  null|     QR|9.718129403E10|00713|      B2|
7:18:...|
|4664261| 2016|     5| 213.0| 213.0|    HOU|20597.0|    1.0|     TX|20641.0|  24.0|    2.0|  1.0|20160523
ull|      G|      O|   null|      M|   1992|11222016|     F|  null|     QR|9.717972563E10|00713|      B2|
7:18:...|
+-------+-----+------+------+------+-------+-------+-------+-------+-------+------+-------+-----+--------
---+-------+-------+-------+-------+-------+--------+------+------+-------+--------------+-----+--------+
--------+
only showing top 3 rows
```

- The ETL processes result in the data model outlined in the write-up.
- A data dictionary for the final data model is included.
- The data model is appropriate for the identified purpose.

✔ Good job on providing the data dictionary!

✔ Each table is appropriate for the identified purpose!

✔ Each table is appropriate for the identified purpose!
✔ The process results are provided in your submission

The project includes:

- At least 2 data sources
- More than 1 million lines of data.
- At least two data sources/formats (csv, api, json)

Your project meets all the requirements.👏

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START