DS-2002 – Data Project 2 (Course Capstone)

100 points
The goal of the second data project, building upon the first project, is to further demonstrate (1) your understanding of and (2) competence implementing the data science systems covered throughout this course (e.g., Relational & NoSQL databases, ETL process pipelines, data transformations, SQL and Python scripts, API's and cloud services). Please submit your project in your GitHub Repo or file drop on Collab.

Data Projects must be done individually.

Putting it All Together: Data Integration & Analysis

Deliverable: Design and populate a dimensional (star schema) data mart that represents a simple business process of your choosing. Examples might include retail sales, procurement, order management, transportation or hospitality bookings, medical appointments, student registration and/or attendance. You may select any business process that interests you, but remember that a dimensional data mart provides for the post hoc summarization and historic analysis of business transactions (represented by a Fact table) that reflect the interaction between various business entities (e.g., patients & doctors, retailers & customers, travelers & airlines/hotels) as represented by dimension tables.

Your project should demonstrate your ability to implement the Data Lakehouse architecture using Databricks Spark Structured Streaming & Delta Tables technologies. It should demonstrate your understanding of the differing types of data systems (e.g., Relational (aka SQL), NoSQL, File Systems, API's), and how data (structured, semi-structured, unstructured) can be **extracted** from those source systems, **transformed** (cleansed, integrated), and then **loaded** into (or exposed through) a destination system that's optimized for post hoc diagnostic analysis. Your project should also demonstrate your knowledge of data integration design patterns like ETL, ELT and ELTL, and architectures (e.g., lambda or kappa) for integrating batch and real-time (streaming) data sources.

Requirements:
Your solution (database schema) needn't be complex, but should meet the following requirements:

- Your solution must include a **Date dimension** to enable the analysis of the business process over various intervals of time *(the code for creating this in MySQL has already been provided for you).*
- Your solution must include at least 3 additional dimension tables (e.g., buyers, sellers, products)
- Your solution must populate its dimensions using data originating from the following sources:
    - A relational database like MySQL, Oracle or SQL Server
    - A NoSQL database like MongoDB, Redis, Cassandra or HBase
    - A Cloud file system like Azure Data Lake, AWS S3 hosting file-base data (e.g., JSON, CSV)
    - An API that returns a message payload (e.g., JSON, CSV, text) ***optional***
- Your solution must include at least 1 fact table that captures the business process transactions
- Your solution must integrate datum of differing granularity (static and near real-time); i.e., implement Databricks structured streaming to integrate hot-path and cold-path data.

- Your solution must include one or more visualizations that demonstrate the business value of your solution. For example, a "dashboard" developed using Excel, Power BI, Tableau or other data visualization tool capable of demonstrating the use of PivotTables and/or Pivot Charts

Benchmarks:

1. Your solution must demonstrate accumulating data that originates from a real-time (streaming) data source for a predetermined interval (mini-batch), integrating it with reference data, and then using the product as a source for populating some aspect of your dimensional data mart. (i.e., implement something like the Databricks bronze, silver, gold architecture).
    a. Your solution must demonstrate the integration of streaming data for at least 3 intervals. This behavior can be mimicked by exporting transaction (fact table) data into a collection of data files (e.g., JSON, CSV) that represent a sequence of rows from that table.
    b. Your data visualization(s) need NOT reflect the integration of data in real-time.
2. You must submit all reference data used to populate the source databases, JSON/CSV files, etc.
3. You must submit all SQL code, including any data definition and data manipulation statements.
4. You must submit all Python code needed to implement data integration, and any object creation.
5. You must submit either all data visualization source files (e.g., Excel, Power BI workbook).
   – OR –
6. Submit screen-grabs of your finished data visualization(s)

Note: *You may utilize any combination of Cloud service technologies. For example, using Databricks you can collect streaming data from a source API on the Internet, integrate it with reference data that's stored in other Cloud hosted data services (e.g., Mongo DB Atlas, Azure SQL Database, Azure Data Lake Storage), and then expose it using your dimensional Data Lakehouse data mart.*

Grading:

- Successful deployment – 40%.
- Functionality that meets all benchmarks – 50%.
- Documentation – Describe your process, code, deployment strategy – 10%.

Publicly-available sample databases:

- https://dataedo.com/kb/databases/mysql/sample-databases (Sample MySQL databases)
- https://docs.microsoft.com/en-us/sql/samples/sql-samples-where-are?view=sql-server-ver15 (Microsoft SQL samples)

Publicly-available datasets:

- https://www.kaggle.com/datasets
- https://data.world/
- https://www.data.gov/
- https://opendata.charlottesville.org/

Publicly-available APIs:

- https://docs.github.com/en/rest
- https://developer.twitter.com/en/docs/twitter-api
- HUGE LIST: https://github.com/public-apis/public-apis