Course Project Presentation - BART & VTA Data Model: Exploring BART, and connecting SJSU

Team 1 - Nikhil Thota, Madhura,
Sowmya Kuruba, Swathi Ramesh,
Saketh Reddy

Introduction

- BART - Bay Area Rapid Transportation & VTA - Valley Transportation Authority
- We have tried to explore the BART & VTA data models using their respective GTFS datasets.
- This project also aims to answers some of the interesting questions w.r.t time and costs of BART between different stations. MongoDB was used to analyze the BART ridership data.
- Polyglot persistence implemented with BigQuery, MongoDB, and Neo4j to answer interesting questions.
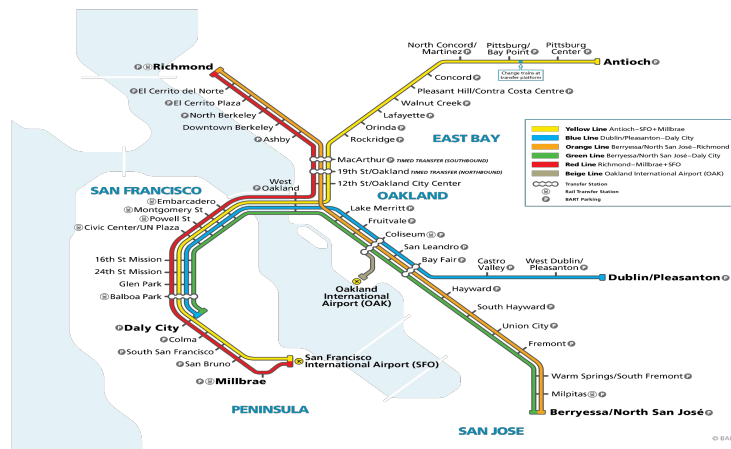
Problem Statement & Goals

To Explore the BART & VTA Data models, and find out interesting answers through questions on BART Ridership, travel time, and Cost.

To try and understand the BART phase 2 project implementation, and how it impacts SJSU commuters

Our project aims to use Polyglot persistence and data analytics to gain insights into BART's transportation system, address its limitations, and inform future planning efforts. We also plan to develop an interactive tool for commuters, analyze the impact of COVID-19 on transportation systems, and apply course concepts to enhance our skills. Ultimately, our goal is to improve the efficiency and reliability of BART while aiding commuters in making informed travel decisions.

BART GTFS, VTA GTFS, and BART Ridership Data will be our Primary Data Sources

System Architecture

OS: Windows, MAC
Language: SQL, Python, CQL, HTML, CSS
Web Framework: Flask
IDE/tool: Jupyter Notebook, SQL Workbench, Mongo Compass, Neo4j desktop, Visual Studio, Google BigQuery, DBT
Database: BigQuery, MongoDB, Neo4j
Data Lake: BigQuery
Data Warehouse: Google BigQuery, Mongo Atlas
Data Visualization: Tableau, Python, Mongo Atlas
**ETL Tool:** DBT

Project Modules

Dataset Finalization

1. Our Primary Dataset, the BART dataset is taken from the official ca.gov.in website, here.
2. We also modeled the VTA transportation system through the datasets available here
3. To conduct our primary analysis, we worked on the historical ridership data which is available here

Data Cleaning

1. **BART GTFS Dataset:** CSV file Conversion, Excel transformations to convert the 26 hour day format to 24 hour format, bringing in lot of redundant columns to a single table by using VLOOKUP(), populated the station details table through web scraping
2. **VTA GTFS Dataset**: CSV file Conversion, VLOOKUP() to normalize the data across tables, removed NULL columns, and converted the 26 hour day format to 24 hour format using Excel Formulas.
3. **BART Ridership Data:** Excel reports which are available on different timelines, removed the null columns, duplicates, and spaces

Graph Modelling:
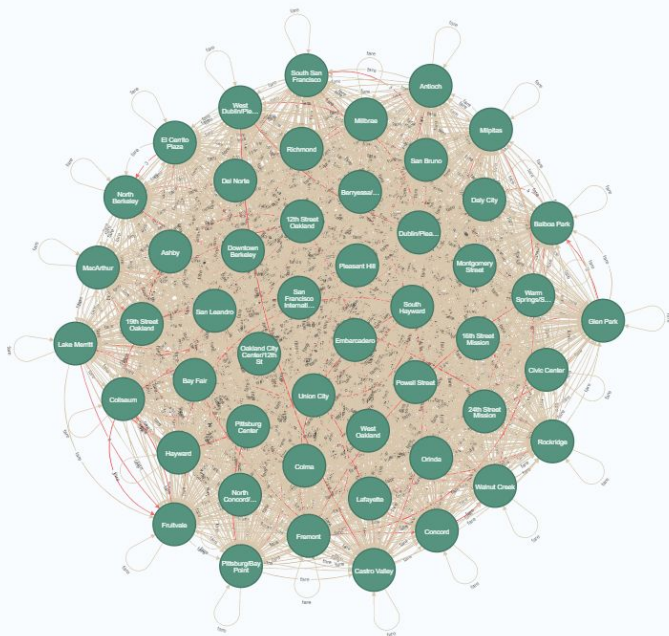
Overview

Node labels

* (50)   Station (50)

Relationship types

* (2602)   travel_time (102)

fare (2500)

Displaying 50 nodes, 0 relationships.

The graph is created by establishing a Python connection to Neo4j using Pyneo. We loaded the data into the nodes and accurately established relationships between the stations based on two factors: fares and travel time for both directions.

The graph was created with 51 station nodes and 2500 fares and 102 travel_time relationship
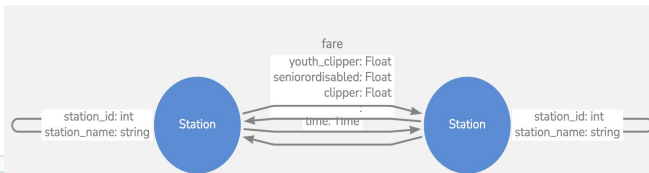
Neo4j Implementation Logic:

Fare calculation:

We have identified four distinct rider categories with different fares in our system. These categories include: Clipper, Clipper Start, Young Clipper, Senior/Disabled Clipper. We have stored these in fares relationship between nodes.

```
MATCH (origin:Station {station_id: 'BERY'})-[fare:fare]->(destination:Station {station_id: 'MLPT'})
RETURN origin.station_name as Source, destination.station_name as Destination,
fare.youth_fare as YouthClipper, fare.senior_fare as
SeniorDisabledClipper,fare.clipper_start_fare as ClipperStart, fare.clipper_fare as Clipper
```

Time calculation:

We have used Neo4j's allShortestPaths function which uses graph algorithm to find the shortest path between two stations w.r.t source and destination station_id.

```
MATCH (start:Station {station_id: "ANTC"}), (end:Station {station_id: "OAKL"})
MATCH path = allShortestPaths((start)-[:travel_time*]->(end))
WITH reduce(totalTime = 0, rel in relationships(path) | totalTime + rel.time) as totalTime, path
MATCH (origin:Station {station_id: 'ANTC'})-[fare:fare]->(destination:Station {station_id: 'OAKL'})
RETURN [node in nodes(path) | node.station_name] as path_stations, totalTime, fare.youth_fare
as YouthClipper, fare.senior_fare as SeniorDisabledClipper,fare.clipper_start_fare as
ClipperStart, fare.clipper_fare as Clipper
ORDER BY totalTime ASC
LIMIT 1
```

Neo4j Implementation Logic:

We have calculated estimated travel time from Berryessa to SJSU which is part of Bart Phase 2 project. Then we have given added this to source of traveller to get travel time from source to SJSU.

Track length form Berryessa/North San Jose to Station
to Downtown San Jose Station = 6.3 miles
Average Speed of BART = 35 mph
Time taken to travel from Berryessa/North San Jose to
Station to Downtown San Jose Station =
 6.3 miles / 35 mph = 10.8 minutes ~ 11 minutes

We have compared it with VTA bus time taken to travel from Berryessa to SJSU.

VTA time taken from Berryessa/North San Jose
Station to Downtown San Jose Station = 17 mins

We have calculated estimated price for Bart Phase 2 from Berryessa to SJSU.
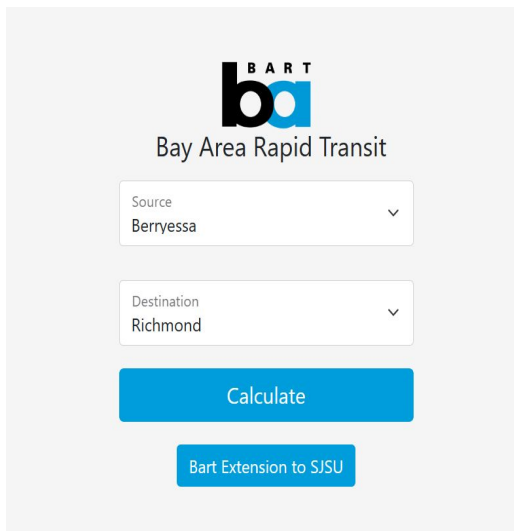
Average fare for Youth Clipper between any two stations = $2.5
Average fare for Senior/Disabled Clipper between any two stations = $2.06
Average fare for Clipper Start between any two stations = $4.42
Average fare for Clipper between any two stations = $5.5

We have compared it with VTA bus fare price to travel from Berryessa to SJSU.
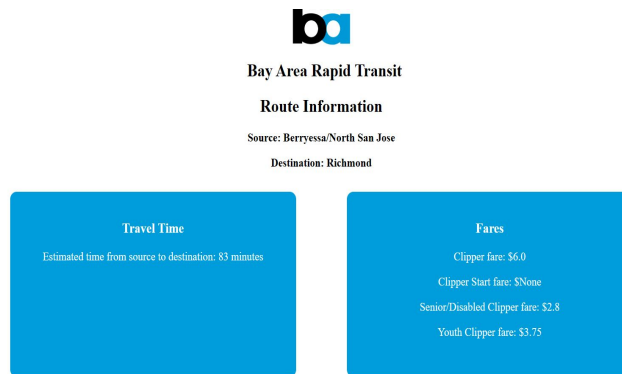
VTA fare from Berryessa/North San Jose Station to
Downtown San Jose Station = $2.5

Application Development and UI:

We have developed a user-friendly Flask interface that enables users to select their source and destination stations with ease. The application then displays the shortest estimated travel time between these stations and provides information on the available fare options. Additionally, our interface allows users to explore estimated travel times and fares for BART's upcoming extension project to SJSU, as well as for existing BART and VTA combination travel.

BART
ba
Bay Area Rapid Transit

Source
Berryessa

Destination
Richmond

Calculate

Bart Extension to SJSU

ba
Bay Area Rapid Transit

Route Information

Source: Berryessa/North San Jose

Destination: Richmond

Travel Time

Estimated time from source to destination: 83 minutes

Fares

Clipper fare: $6.0

Clipper Start fare: $None

Senior/Disabled Clipper fare: $2.8

Youth Clipper fare: $3.75

Backend Integration and analysis from Neo4j:

Our Flask application is connected to a Neo4j graph database via the GraphDatabase driver, which we established using the database's URI and authentication credentials.

```
driver = GraphDatabase.driver("bolt://localhost:7687", auth=basic_auth("Bart", "Bart@123"))
```

Using this setup, we can effectively query and retrieve data from the Neo4j graph database within our Flask application.

RDBMS : Data Operations

1. Normalization

- A new column called 'service_id' was added to the stops table, and the values were looked up from the trips table using VLOOKUP() to satisfy the 3NF condition.

- Two new columns named 'origin_station' and 'destination_station' were created in the fare_rules table to fulfill the 3NF requirements.

- A duplicate table called 'stop' was created (derived from station_details) to connect the left and right sections of the ERD.

- The station_details table had two new columns 'agency_id' and 'stop_id' added to manage foreign key relationships.

2. Data Creation

All the tables with respect to BART, and VTA datasets(mentioned above in the ERD) have been cleansed, and loaded into BigQuery.

The SQL scripts for this action can be found here

3. Data Population

Once we created the tables, we were able to load the data for each of these tables from the local directly, through one of the BigQuery features, which lets us do a data import like SQL workbench.

Data Modelling
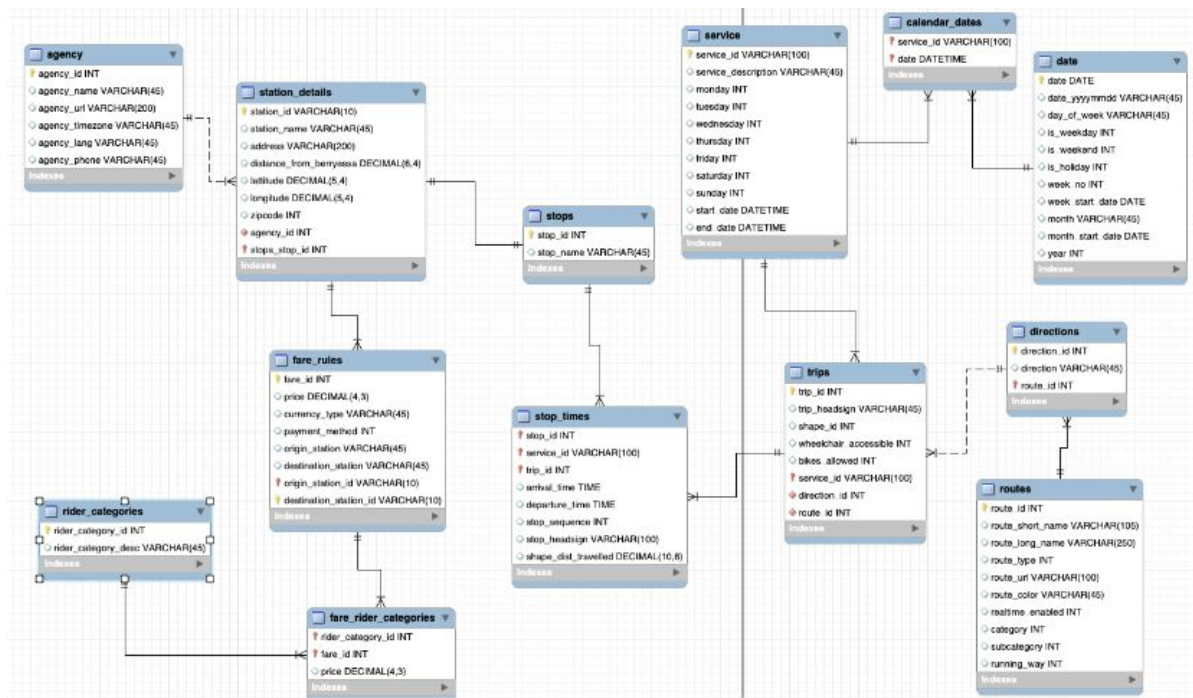
Using the GTFS Datasets, we were able to create ERDs, after some excel transformations, and understanding the cardinalities.

tables in BART: 13
tables in VTA: 10
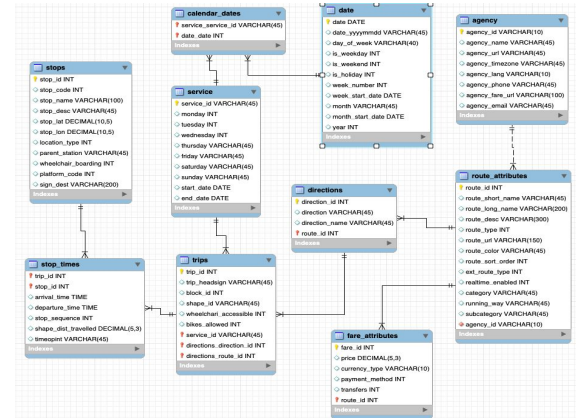
ETL using DBT

We have used DBT cloud as an ETL tool in our scenario, we tried to write SQL transformative queries in dbt on the raw tables from BQ.

These DBT transformed tables are going to get deposited in the target schema based on the environment we ran the job (Development & Production).

We have also scheduled a daily pipeline which runs daily using a cron schedule, and refreshes the data in the respective dataset of BQ.


The above job has been scheduled on Production environment, the tables which are scheduled for the daily runs get updated in a target dataset (dbt here, in BQ).

Data Warehousing
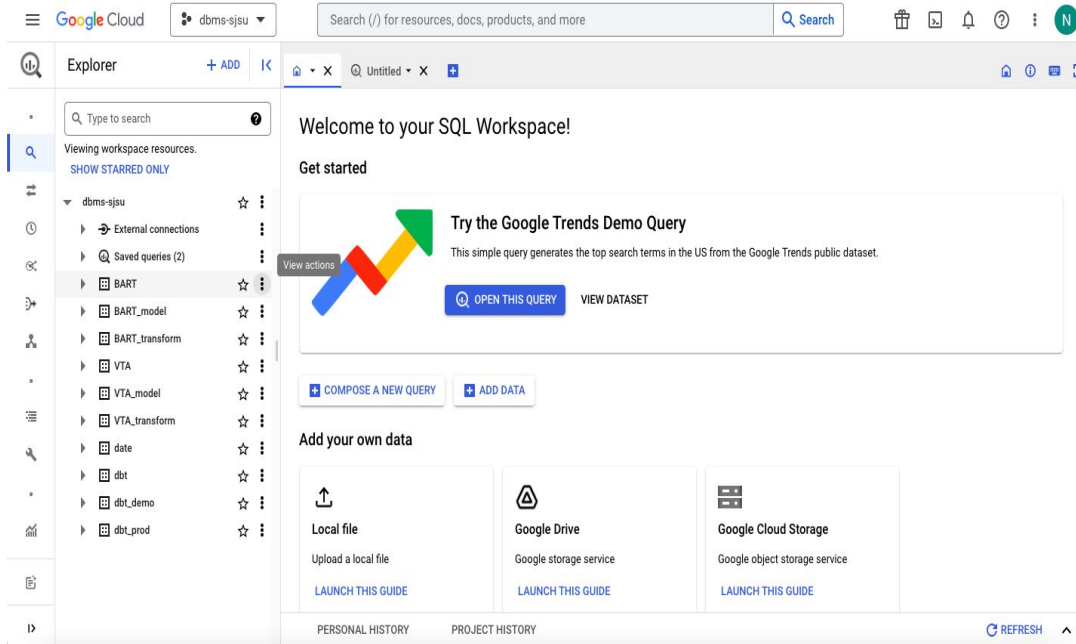
- We have implemented our relational database as a OLAP system in BigQuery, which is a Cloud Data warehouse.
- Project -> Datasets -> schemas
- Cloud warehouse advantages
- DBT connection through JSON credentials of the BQ service account
- IAM roles can be managed, and created according to the use case.
- Each dataset in the project has a purpose,.
  1. BART - Raw tables
  2. BART_Transformed - DBT transformed
  3. BART_model - Analytics views.
  4. Dbt - Daily pipeline from DBT.
- We have also created Date table in separate dataset because of the usage it has for other schemas, it helps us to maintain concept hierarchy
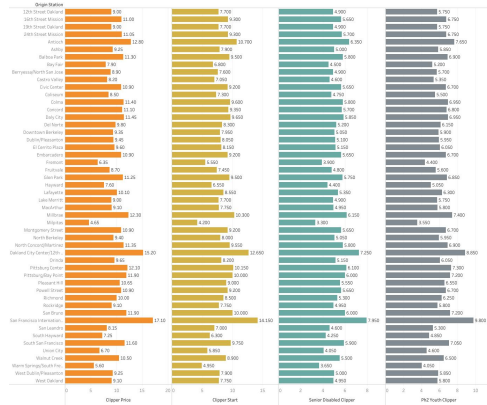
Dimensional Modelling
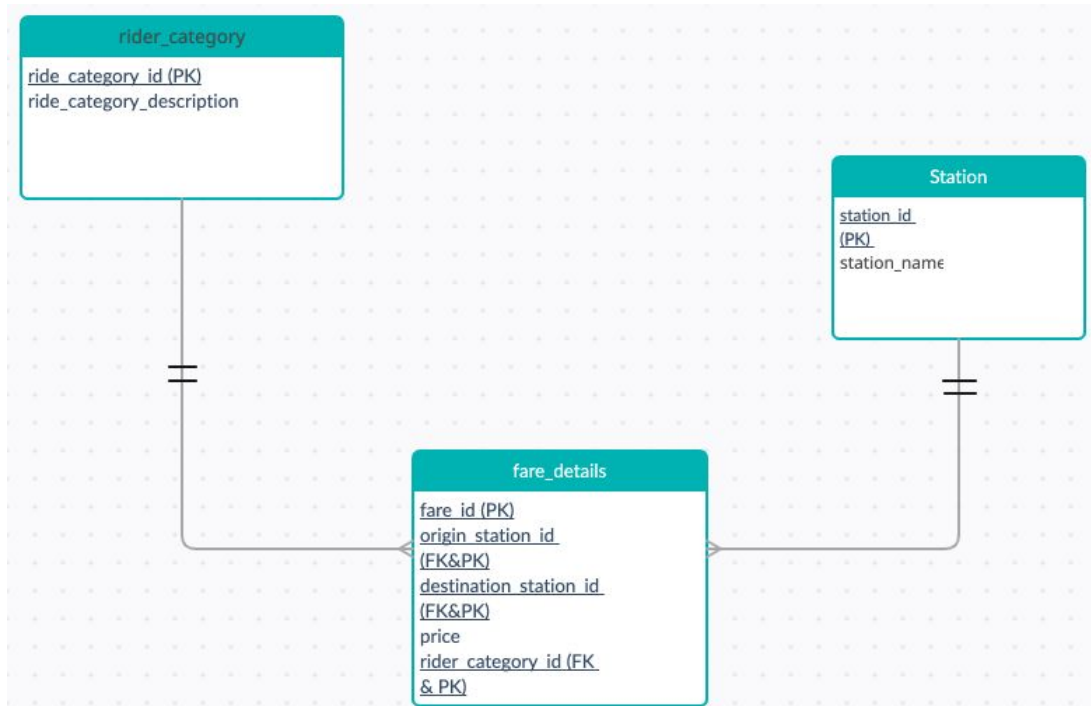
Our Database Design was in such a way that we couldn't create a star schema out of it when we look at the whole system.

So we tried to use dimensional modelling to Fare schema because rider categories, and station details are the dimensions to this Fact table
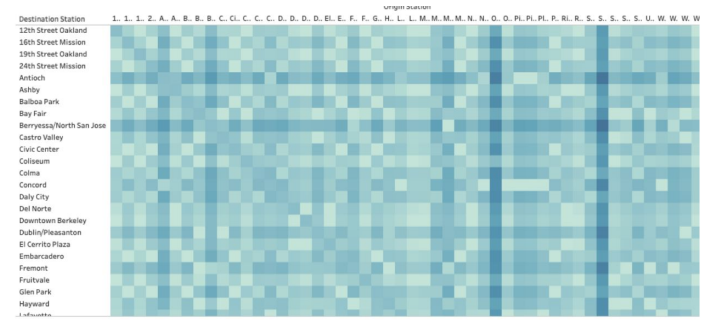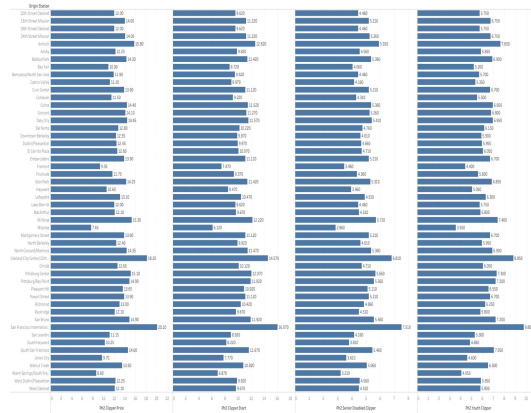
Results of the Exploratory Data Analysis - RDBMS

Analysis Fare trends when BART extension is in effect

Analyzing price trends in popular areas and airports can aid in determining pricing strategies for future stations based on past trends

Results of the Exploratory Data Analysis - MongoDB
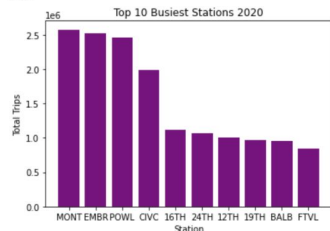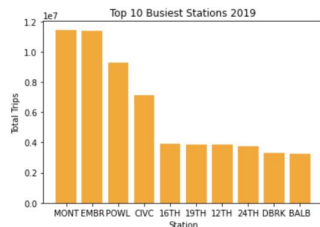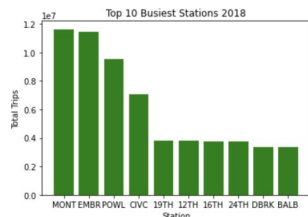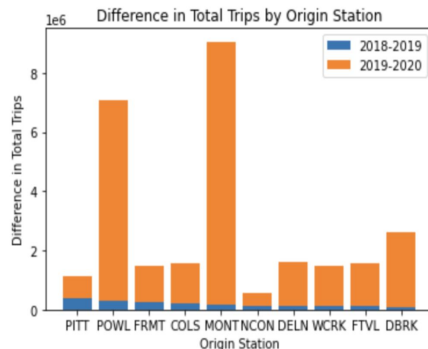
- It shows the difference in trip count between the years 2018, 2019, and 2020 for each station and returns the top 10 stations with the largest drop in ridership.

- The top 10 stations experienced a significant drop in ridership in 2019-2020, which can be attributed to the impact of COVID-19.

- Our other various analysis examined the impact of COVID-19 on BART ridership.

- We found a significant decrease in trips per day and hour from April to December 2020, with some stations seeing a larger decline.

- The busiest hour shifted from 4 PM to 9 AM, possibly due to changes in commuting patterns and remote work.

- Overall, reduced travel, remote work, and public health concerns were the main factors affecting BART ridership during the pandemic.

Performance Evaluation

- Performance analysis between MySQL and Neo4j for Fare calculation query
- We conducted a performance analysis between MySQL and Neo4j for the fare calculation query by taking the average query run time for 50 runs. Our case study compared RDBMS and NoSQL for our project and we observed significant performance differences between the two. Neo4j outperformed MySQL due to its direct relationship between source and destination stations, while MySQL used self-join to establish the relationship. This analysis provides insights into the strengths of NoSQL databases in handling complex relationships in data.

MySQL	Neo4j
423ms	10ms

Key Learnings

1. Modelling real world database systems by exploring VTA, BART GTFS datasets.
2. Polyglot persistence implementation with BigQuery, MongoDB, and Neo4j for different data use cases.
3. The concepts of Data warehousing, and all the rules followed while creating a data warehouse.
4. Integrating other cloud tools with cloud warehouses using the JSON keys and service accounts.
5. Pipeline setups for ETL, operations by scheduling jobs through DBT.
6. Working with Flask, for Web application,

Course Topics

This course project covered all the aspects of our course, all the CLO's mentioned in the description have been met.

Concepts of ER Model, Data wrangling, Database creation, and population, database creation using BigQuery, ETL operations through DBT, OLAP Data warehousing through BigQuery.

NoSQL concepts like MongoDB to do OLAP ridership data, while Neo4j to answer shortest path, least cost questions

Client Server Architecture in BART app
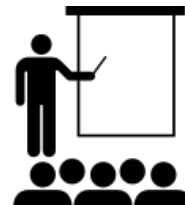
Rubrics

Innovation

The idea itself was a little unique, because we tried to implement a real world scenario of the broad usage of Databases.

The idea of combining the VTA, BART models to calculate the time, and cost for SJSU commuters, was completely new to the existing BART system.

Significance to Real World

Comparing the Phase 2 extension of the BART line, with the existing VTA in terms of Cost, and time for the SJSU commuters.

Also answering the shortest path, least cost questions through Neo4j directly helps the commuters to make their lives a little easier.

Lessons learnt

Polyglot persistence

Performance analysis between MySQL and Neo4j for Fare calculation query.

From the above experiment, we have concluded that Neo4j is much much faster than RDBMS for much complex queries

Future Scope

- Real time Data Integration - Creating a OLTP Databases for VTA, and BART to understand various customer trends.

- We can create multiple journeys combining VTA AND BART based on the origin station and destination station, and their scheduled routes.

- Integrating with Other transit systems.

- Optimizing the existing BART routes, and schedules, to decrease the revenue losses for the company.
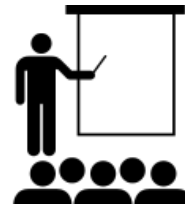
Technical Difficulties

Modelling the BART, and VTA databases through understanding the cardinalities between different entities.

All the tools which have been used in this project were our first time experiences working with them, so there was a huge learning curve.

MongoDB, and Neo4j technologies to implement Ridership analytics for BART.

Team Work

Task	Contributors
Data Gathering	Team
Data Cleaning	Team
ERD Modelling	Nikhil, Swathi, Saketh
Graph Modelling	Sowmya, Madhura
Database creation and population	Nikhil, Swathi
Graph Creation	Sowmya, Madhura
ETL	Nikhil,Swathi
Logic Planning, and Integration	Team
Mongo DB Analysis & Aggregation Pipeline	Swathi, Saketh
UI creation	Madhura, Sowmya
UI creation with DB	Madhura, Sowmya
Data Warehouse	Nikhil, Saketh
Visualization	Team
Project Artifacts	Team
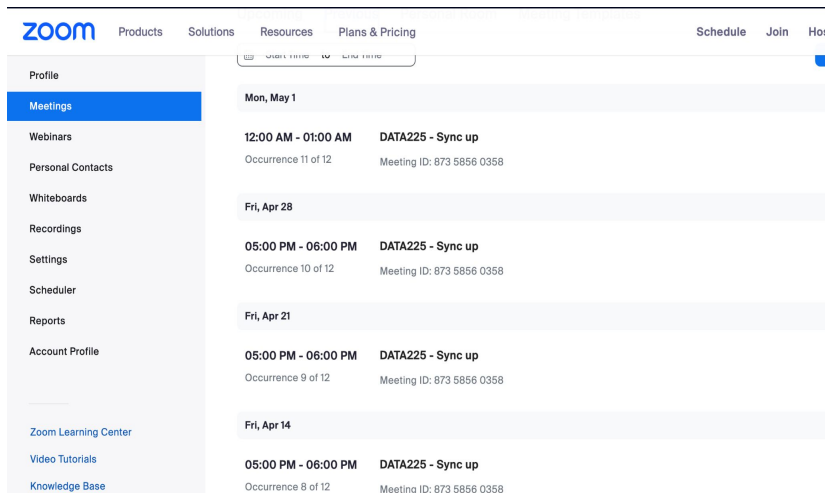
Practiced Pair programming for the whole project, which helped us in bringing the best possible version of our work, and results.

Task Management was done through Jira - A project management tool.

Scrum meetings were held weekly once throughout the course of this project to check the progress of the project, and to discuss the next milestones in detail.

Slack was used for all kinds of communication.

Project Management

Project Management tools like JIRA, Slack and Zoom (for weekly sprint meetings) were used to coordinate this project.

References

- https://www.mongodb.com/docs/manual/aggregation/
- https://stackoverflow.com/questions/17805304/how-can-i-load-data-from-mongodb-collection-into-pandas-dataframe
- https://cloud.google.com/docs/get-started
- https://cloud.google.com/bigquery/docs
- https://neo4j.com/docs/
- https://publications.waset.org/10011501/optimal-and-critical-path-analysis-of-state-transportation-network-using-neo4j
- https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1026&context=sais2013
- https://scholar.smu.edu/datasciencereview/vol3/iss1/11/
- https://www.scitepress.org/papers/2011/34315/34315.pdf
- https://dl.acm.org/doi/abs/10.1145/248603.248616
- Thank you image reference - https://www.dreamstime.com/photos-images/thank-you-end-presentation.html
- For Infographics - https://adioma.com/

THANK
YOU
FOR
YOUR
ATTENTION

?
?

Appendix

Presentation	All Slides	Story telling Flow
Key Learnings	# 18	From the project
Significance to the Real World, Innovation, Lessons learnt	# 19	Course work coverage & real world significance
Demo	#16-18	Visualizations and Results
Code walkthrough	# 8-12	Neo4j logics, SQL queries
Slides	All Slides	Slides for presentation
Technical Difficulty, Future Scope,	# 20	Future work possibilities

Appendix

Used Scrum model for discussions	Yes	Scheduled weekly calls in Zoon
Version Control through Git	Yes	Here
Report creation - Format, completeness using Latex	Yes	Very detailed work done in Project
Performed substantial analysis on databases in different databases (git link has all the files)	Yes	Here are the visualizations
Used new database tool (BigQuery), ETL tool (DBT), Used RDBMS, used Data warehouse (BigQuery), Includes DB calls (DBT and Neo4j)	Yes	Whole report and Slides
Jira Project Management tool, Slack for communications	Yes	Screenshots in the report