BART and VTA Data Model: Exploring BART, and connecting SJSU

Sowmya Kuruba, Madhura Bhatsoori, Nikhil Thota, Swathi Ramesh, Reddysaketh Reddy Chappidi

***Abstract*—The primary objective of this database project is to conduct an in-depth analysis of the Bay Area Rapid Transit (BART) and VTA datasets to extract valuable insights. To achieve this goal, we will first obtain and clean the datasets, ensuring that the data is accurate and free from errors. We will then model them separately to obtain insights into the historical ridership data, present BART structure, and VTA operations. The insights obtained from the data analysis will enable us to create useful visualizations, that provide valuable information for BART business decisions. Additionally, we implemented an interactive web app that provides answers to interesting questions about travel time and cost for different BART destinations. These tools will be useful for commuters seeking information on the most cost-effective and efficient routes to their destinations and also promote the BART business. In addition to analyzing the existing transportation model, we also aim to compare it with Phase 2 implementation of the BART project. This comparison will identify any potential areas of improvement and provide valuable insights for future transportation planning and development.**

***Index Terms*—BART, VTA, Polyglot Persistence, SQL, Data Modelling, Cloud data warehouse, BigQuery, ETL, DBT, Neo4j, Graph Databases, NoSQL, MongoDB, Tableau, Python, Cyhper, Aggregation pipeline, Flask, Client Server Architecture.**

I. Introduction

THIS project showcases the concept of Polyglot Persistence, which involves using multiple databases to analyze and visualize datasets. To demonstrate this concept, we will utilize several databases, including MySQL, BigQuery, MongoDB, and Neo4j, and analyze ridership information using General Transit Feed Specification (GTFS) datasets from the VTA and BART transportation systems. To ensure that each dataset is modeled effectively, we employed separate data stores for each database type, as different datasets fit into different data stores based on the problem we aimed to solve. We used Google BigQuery relational databases to analyze the BART and VTA structures, conduct historical fact analysis and trend comparisons across different years using MongoDB, and used Neo4j Graph databases to answer time and cost-related questions within the BART system.

II. Problem Statement

The Bay Area Rapid Transit (BART) and Santa Clara Valley Transportation Authority (VTA) transportation systems serve millions of commuters daily, yet there is a significant lack of comprehensive insights into their operations. The current datasets are often siloed and it is very challenging to analyze them, resulting in a limited understanding of the structures and ridership patterns. This project aims to address these challenges by utilizing Polyglot Persistence to analyze ridership patterns from BART and VTA systems. The model will provide comprehensive insights that can inform future transportation planning and development efforts. Additionally, by creating an interactive tool that provides time and cost-related information for different BART destinations, we enable commuters to make more informed travel decisions, ultimately improving the overall efficiency of the public transportation systems.

III. Motivation

The transportation systems of the Bay Area, including BART and VTA, play a critical role in the daily lives of millions of commuters, making it essential to understand their operations comprehensively. However, the transport system information is complex and it is challenging to derive meaningful insights for both commuters and the transportation company. The motivation of this project is to understand different methodologies to store data efficiently and use advanced data analytics techniques to provide a deeper understanding of transportation systems. By doing so, this project will not only benefit the transportation industry but also the commuters who rely on these systems every day. The insights gained from this project can inform future transportation planning and development efforts, ultimately leading to more efficient and reliable transportation systems that meet the needs of the Bay Area's growing population.

IV. Goal

The goal of this project is to utilize Polyglot persistence and advanced data analytics techniques to gain comprehensive insights into BART. By doing so, the project aims to address the limitations and inform future transportation planning and development efforts, ultimately improving the overall efficiency and reliability of the transportation systems. The project also aims to develop an interactive tool that provides commuters with valuable insights into BART travel time and cost to different destinations, aiding in making more informed travel decisions. Additionally, the project aims to analyze the impact of the COVID- 19 pandemic on transportation systems and explore potential solutions to mitigate its effects. To further enhance our skills and understanding of the subject matter, we plan to apply the concepts learned in the course wherever applicable throughout the project.

V. Datasets

1) **BART GTFS Data** [6] The dataset contains information about BART stations, ridership, fares, trips, services, etc.

This dataset is used both in could data warehousing and graph databases.

2) **VTA GTFS Data** [7] The dataset contains information about VTA stations, ridership, fares, trips, services, etc., and is used in cloud data warehousing.
3) **BART Historical Ridership Data** [8] was used to conduct our primary analysis. We have aggregated this data on different timelines and tried to answer some interesting questions using MongoDB, which we will cover in future sections.

VI. Literature Survey

Literature Survey was conducted to get all the information related to our project, below are some of the important research references which we studied in detail:

- **NoSQL vs RDBMS - Why There is Room for Both** [1] This paper can offer valuable insights for our project by presenting a thorough analysis of the advantages and disadvantages of both NoSQL and RDBMS technologies, and how they can be optimally integrated to meet specific data management requirements.
- **Intelligent Transportation Systems Data Warehouses and Their Applications** [2] This paper provides valuable insight into the benefits and challenges of using Intelligent Transport Systems data warehouses in transportation systems.
- **An Overview of Data Warehousing and OLAP Technology** [3] describes tools for extracting, cleaning, and loading data into a data warehouse; multidimensional data models typical of OLAP.
- **Forecasting San Francisco Bay Area Rapid Transit (BART) Ridership** [4] provides valuable insights into the factors that influence BART ridership, such as population growth, employment rates, and gas prices. Additionally, the forecasting methods discussed in the paper can help in predicting future ridership trends and developing strategies to address potential challenges.
- **Relational Database and NoSQL Inspections using MongoDB and Neo4j on a big data application** [5] This paper discusses polyglot persistence, and what kind of queries are run efficiently on what databases.

VII. System Requirements

A. Functional Requirements

1) Data Pre-processing.
2) Data Modelling.
3) Aggregation pipelines using MongoDB
4) ETL using DBT and BigQuery
5) Analysis using SQL in BigQuery
6) Cloud Data warehouse
7) Visualizations using Tableau,
8) App development using Flask
9) Implementation of Graphdb using Neo4j
10) Calculator functionality using Cypher
11) Integrating UI to database server

B. Software Requirements

1) OS: Windows, MAC
2) Language: SQL, Python, CQL, HTML, CSS
3) Web Framework: Flask
4) IDE/tool: Jupyter Notebook, SQL Work-bench, Mongo Compass, Neo4j desktop, Visual Studio, Google BigQuery, DBT
5) Database: BigQuery, MongoDB, Neo4j
6) Data Lake: BigQuery
7) Data Warehouse: Google BigQuery, Mongo Atlas
8) Data Visualization: Tableau, Python, Mongo Atlas
9) ETL Tool: DBT

VIII. System Design

A. System Architecture

Fig 1. presents an overview of the project and its different subsections in the system architecture.

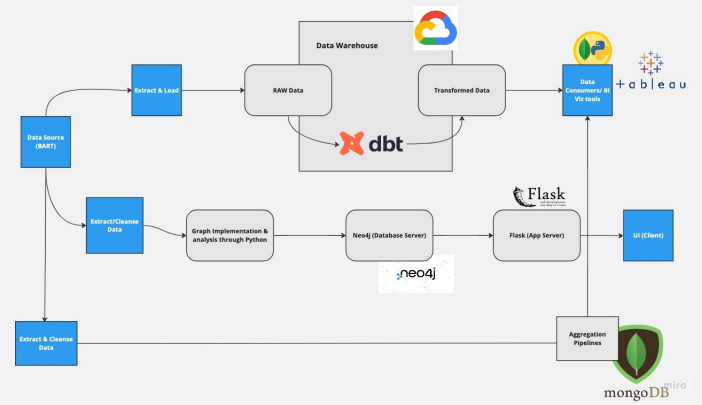


Fig. 1. System Architecture

B. Data Modeling

BART and VTA ERDs were implemented using MySQL workbench and the schemas were normalized to 3NF.

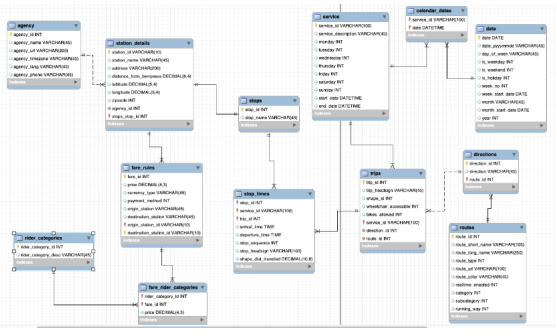


Fig. 2. BART ERD

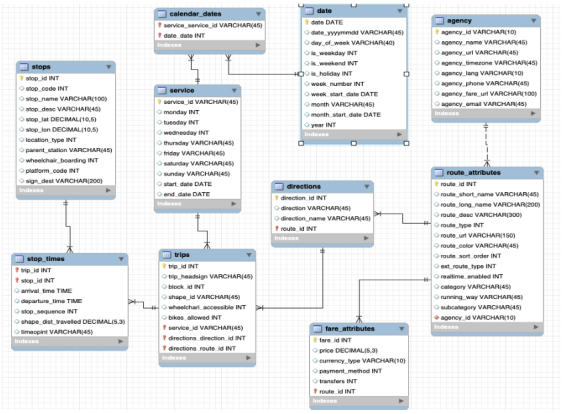


Fig. 3. VTA ERD

C. Graph Data Model

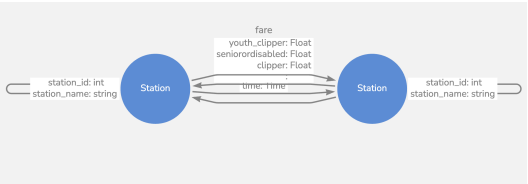


Fig. 4. Graph representing nodes(stations) and time-fare relationships

D. Star Schema

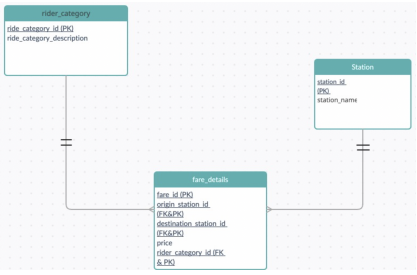


Fig. 5. Start schema for data warehousing

To analyze the fare section of our data using a data warehouse, we opted to use a Star Schema. This allowed us to easily map out the relationships between our various data sets and efficiently organize the information we needed to analyze.

E. Sequence Diagram

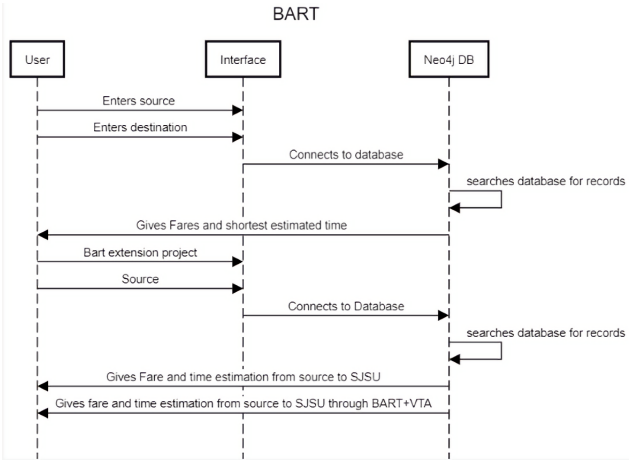


Fig. 6. The sequence diagram wrt our BART Calculator application.

IX. Project Modules

A. Data Preprocessing

1) Data Cleaning for BART datasets:

- Converted .txt files into CSV using Microsoft Excel's Text to Columns feature with comma as a delimiter.
- Fixed errors caused by BART's 26-hour daily schedule in the time field during loading into BigQuery by using an Excel formula.
- Deleted columns that were completely null in multiple CSV files and removed extra spaces in columns using the strip() function.
- Populated the station_master for BART datasets by scraping data from the internet and cleaning it for duplicates, extra spaces, and unwanted characters.
- Scraped BART time as an average between connected stations using Google Maps for analysis.

2) Data Cleaning for VTA datasets:

- Converted .txt files into CSV using Microsoft Excel's Text to Columns feature with commas as delimiter.
- Fixed errors caused by BART's 26-hour daily schedule in the time field during loading into BigQuery by using an Excel formula to convert the time format.
- Deleted columns that were completely null in multiple CSV files and handled other blank values using fillna() in Python.
- Merged multiple route-level tables (using VLOOKUP on route_id) to create route_attributes of VTA.

B. BART Calculator application using Neo4j

1) Graph Modelling and implementation using Neo4j: The graph was created using a Python connection to Neo4j using Pyneo. The data was loaded into the nodes and relationships were appropriately created between the station for both times (travel_time) and fare(fare)

C. BART Graph Model

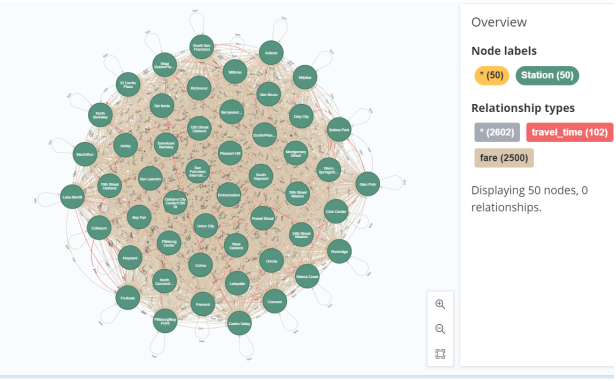


Fig. 7. BART graph model using neo4j

The graph was created with 50 station nodes, 2500 fare relationships, and 102 travel_time relationships.

Implementation Logic: We have identified four distinct rider categories with different fares in our system. These categories include.

1) Clipper
2) Clipper Start
3) Young Clipper
4) Senior/Disabled Clipper

Each of these categories has a unique fare structure, with Clipper Start offering lower fares for the first few trips, Young Clipper offers discounts for riders under a certain age, and Senior/Disabled Clipper offering reduced fares for eligible riders. By categorizing riders based on their fare structures, we can more accurately calculate and display the appropriate fares for each trip within our system.

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
```

Fig. 8. The shortest path between any two stations w.r.t time

```
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
```

Fig. 9. The fare between any two stations

```
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
```

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

Fig. 10. Logic for Calculating Fare when BART Phase 2 is in Effect

```
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
```

```
VTA fare from Berryessa/North San Jose Station to
Downtown San Jose Station = $2.5
```

Fig. 11. Logic for calculating the time when BART Phase 2 is in effect

1) Application development using Flask: 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.

2) UI and Backend Integration: 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. Once the driver is created, we create a session object, which enables us to execute Cypher queries on the database.

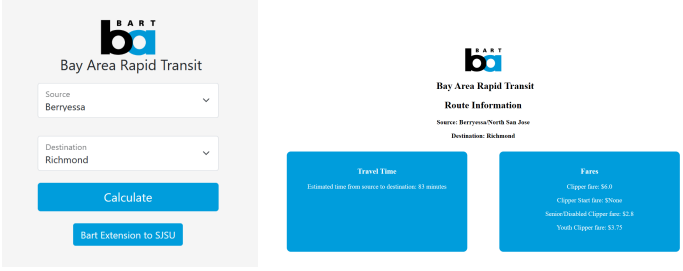


Fig. 12. User interface to show fare and time.

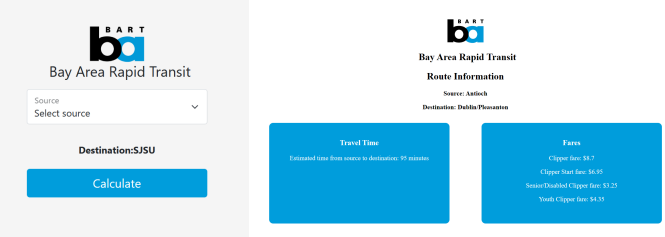


Fig. 13. Fare and time to SJSU

Using this setup, we can effectively query and retrieve data from the Neo4j graph database within our Flask application, allowing us to build powerful and dynamic applications with complex data requirements.
The BART calculator app provides commuters with information on travel time and cost based on their rider category, resulting in an improved customer experience. This, in turn, can positively impact BART's business by increasing ridership and customer satisfaction.

D. Analysis using BigQuery

1) Normalization: To transform the BART datasets into 3NF, the following steps were taken:

1) The cardinalities between service, trips, routes, fares, and stations were analyzed and understood.
2) 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.
3) Two new columns named origin_station and destination_station were created in the fare_rules table to fulfill the 3NF requirements.
4) A duplicate table called stop was created (derived from station_details) to connect the left and right sections of the ERD.
5) The station_details table had two new columns agency_id and stop_id which was added to manage foreign key relationships.

With the above steps, the BART ERD was transformed into 3NF, which will make future modifications more manageable.

2) Database Creation: All the tables with respect to BART, and VTA datasets(mentioned above in the ERD) have been created in BigQuery, and the code is present in BART_creation.sql and VTA_creation.sql

3) Database Population: Once we created the above 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.

E. ETL Operations for BigQuery Data

We have used DBT cloud to perform all our ETL operations. For example, to create the basic transformations like joining some tables with the generic Date table, which we created as part of the warehouse, and other transformation queries where multiple tables were involved to build the new table. We have also scheduled a job, which refreshes every day according to a cron schedule, and updates all the tables which we have transformed/created using DBT.

Setting up the job for the daily pipeline.

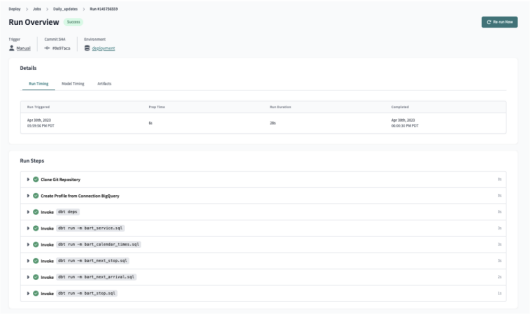


Fig. 14. Creating ETL job using DBT

This pipeline helps us to keep the data updated when sources are getting updated frequently.

F. Data Warehousing for Analytics

As mentioned, Google BigQuery is a cloud Data warehouse, with the help of DBT, we were able to set up two environments, which results in two datasets in the project data lake. One dataset for the development, and the other dataset for daily pipeline and production environment. We had to extract the credentials.keys file from the service account which was linked to the bigquery project I was working on,once I had the key, I had to configure it with dbt, to set the target of the queries right w.r.t environment used.

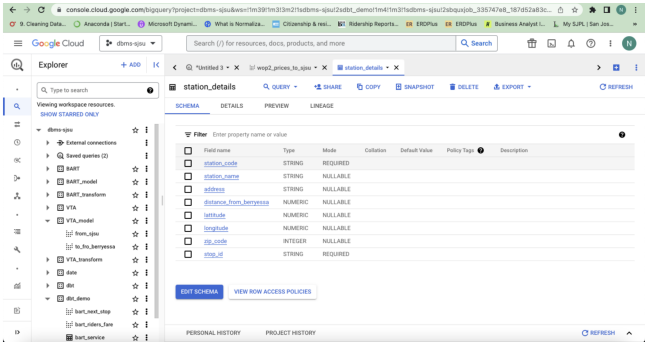


Fig. 15. Big Query Warehouse Dashboard

We implemented the below concepts of Data warehousing with big query.

1) Creating data lakes as per the kind of data, BART, and VTA.
2) Separate datasets for transformed schemas for both BART and VTA, similar structure for dataset which considers the final views of data which are used in the visualization.

3) We created a date schema and table separately so that we can use this table to join, wherever necessary, this will help in reduced data redundancy.
4) The tables/views from dbt also get into the respective dataset whether dbt or dbt_demo based on the environment we ran the jon from, Deployment environment has the target path in dbt, and the development environment in dbt_demo.
5) We also explored the IAM, Credentials, and service accounts to connect with the DBT cloud.

Our analysis source code is present here

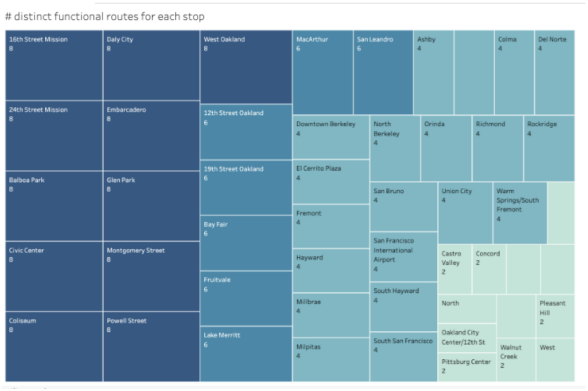


Fig. 18. Distinct routes from each stop

G. Data Visualization for Analytics through Datawarehouse

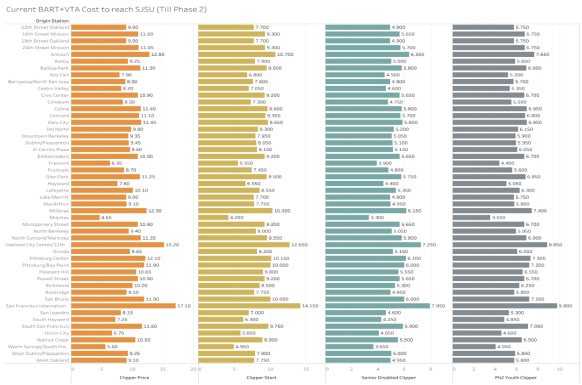


Fig. 16. BART and VTA cost to reach SJSU

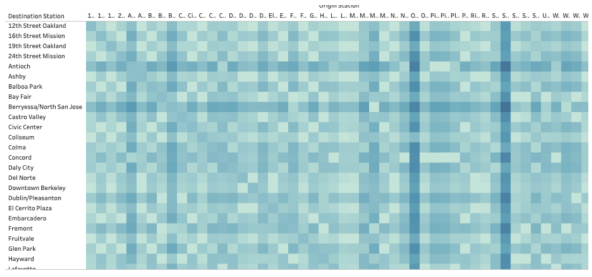


Fig. 17. Big Query Warehouse Dashboard

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

H. Analysis of Ridership and trends using MongoDB

We used MongoDB as our database management system to store, manage, and process large transportation datasets from BART Ridership documents. The MongoDB Aggregation Framework enabled us to perform complex data analysis queries and gain valuable insights. We established a connection between MongoDB and Python using PyMongo, enabling us to execute queries and manipulate data directly from Python. For visualization, we utilized the Matplotlib library in Python.

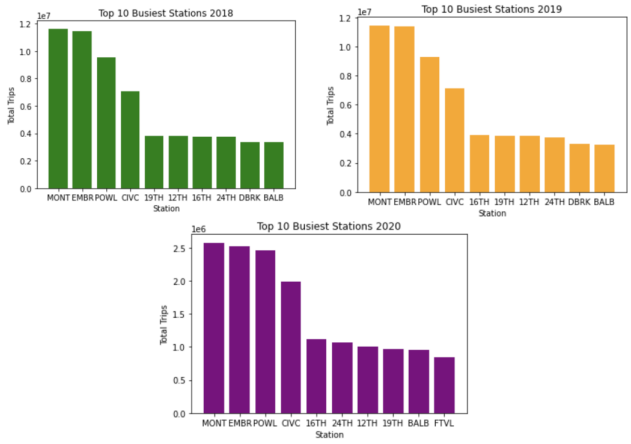


Fig. 19. Top 10 Busiest stations during years 2018, 2019 and 2020

By gaining insights into which stations are the busiest and which experienced the the largest drop in ridership, BART can allocate resources more efficiently, such as by adding additional trains or staff at the busiest stations.

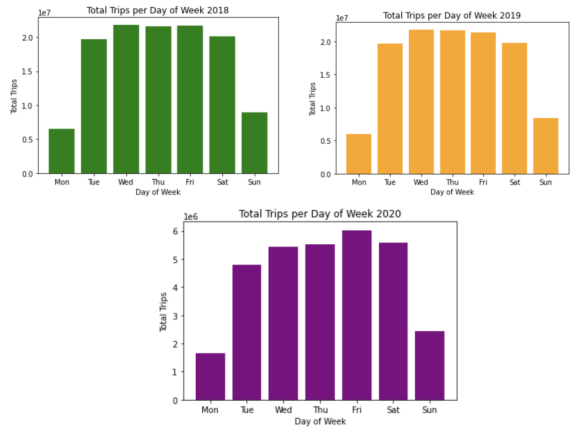


Fig. 20. Comparison of ridership by day of the week during 2018, 2019 and 2020

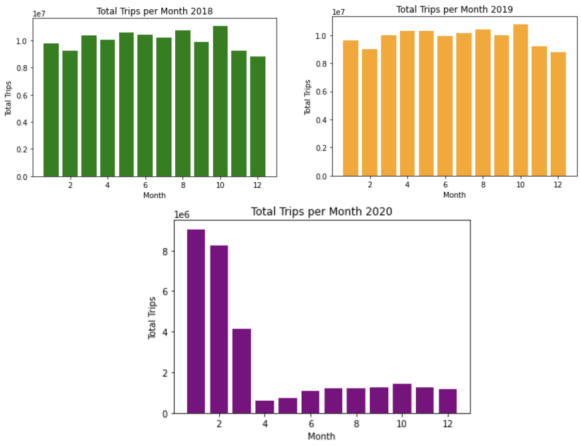


Fig. 22. Comparison of ridership by month in 2018, 2019 and 2020

By analyzing the comparison of ridership by day of the week and month, BART can identify trends and patterns in ridership that can inform future planning and decision-making. This data can be used to adjust services during peak periods and identify areas for growth and improvement.

Understanding the average number of trips per day can also allow BART to adjust schedules and staffing to accommodate ridership patterns and improve overall efficiency.

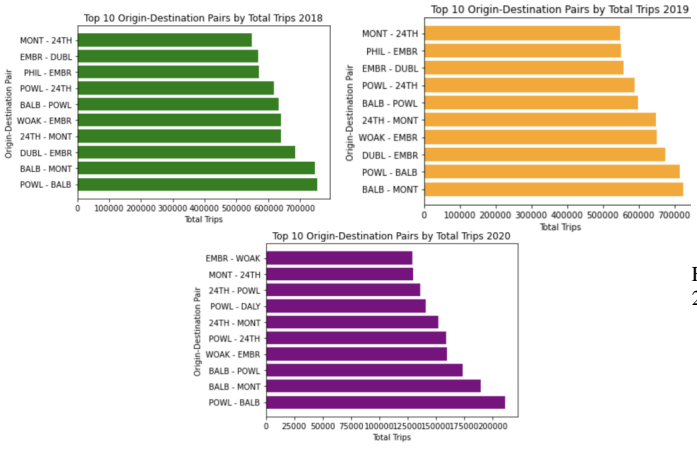


Fig. 21. Most popular origin-destination pairs during 2018, 2019 and 2020

Most popular origin-destination pairs can provide valuable information for businesses looking to tailor their offerings to meet the unique needs of commuters along those routes.

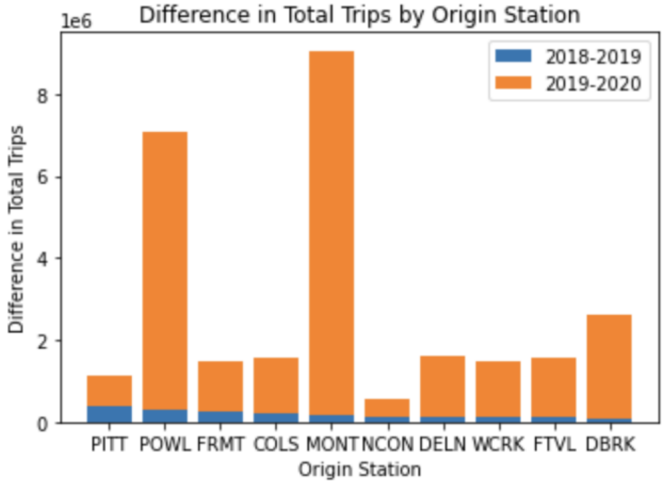


Fig. 23. Top 10 stations that experienced the largest drop in ridership from 2018 to 2020 due to the pandemic

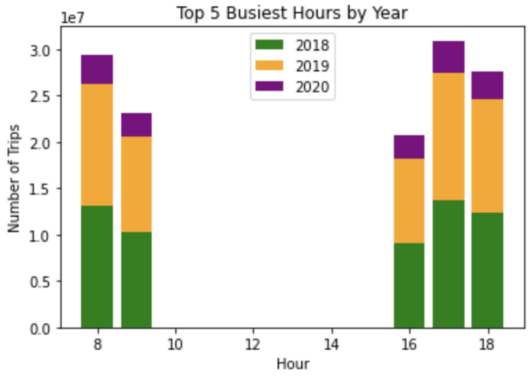


Fig. 24. Comparison of the busiest hour in 2018, 2019 and 2020

By identifying the busiest stations and times of day, businesses can target promotions and advertisements more strategically, maximizing exposure to their target audience.

I. Permormance Analysis

MySQL	Neo4j
423ms	10ms

Fig. 25. 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.

1) Sanity Testing: We applied sanity testing to ensure the correctness and completeness of the data in the BART system. We conducted various test cases to check for consistency and accuracy of the data, including checking for any missing or inconsistent data in the system and verifying the time and fare results with google maps and Bart calculator. In case of any bugs or failures, we addressed and fixed them to improve the reliability of our system. The testing also helped us to identify and incorporate any additional data required for the efficient functioning of our BART system.

X. Conclusion

Our project analyzed historical ridership data, presented BART structure, and Phase2 implementation, providing valuable insights for commuters and business decision-makers. We used big data technologies, graph databases, SQL, and NoSQL to build a polyglot persistence data model and visualized findings with Python and Tableau. Our project unlocked key historical ridership data, presented BART structures, and provided insights for future development, empowering commuters with travel time and cost information for different BART destinations. Using Polyglot Persistence, SQL, Data Modelling, and other technologies, we created a formidable database and transformative tool for better decision-making and improved transportation experience in the Bay Area.

Blogs representing our project work:

1) Bart Using Neo4j
2) BART Ridership Analysis using MongoDB
3) BART and VTA data models using RDBMS

XI. Future Scope

1) Real-time data integration: Currently, the project uses static data from BART. However, integrating real-time data could provide more accurate and up-to-date information for users.
2) Integration with other transit systems: The project currently only considers BART and VTA. However, integrating other transit systems could provide users with more comprehensive travel options.
3) Integration with ride-sharing services: Integrating ride-sharing services such as Uber and Lyft could provide users with additional transportation options and improve their overall travel experience.

XII. Appendix

A. Presentation Skills

We will conduct an interactive session for each team member to deliver a presentation. The entire presentation, including QA, will last for 20 minutes.

B. Code Walkthrough

Please refer to the project modules for a detailed explanation.

C. Discussion

We will be conducting an interactive presentation of our project, including a case study and slides. The session will also involve a QA throughout the presentation.

D. Demo

We will present an interactive demo lasting for 5 minutes, during which any volunteer from our class can test the app.

E. Version Control

We are utilizing a Git repository here to track the code developed by each individual.

F. Significance to the real world

The BART project is significant to the world because it provides a critical transportation service to the San Francisco Bay Area, connecting people to jobs, education, and other opportunities. It also serves as a model for sustainable public transportation, reducing carbon emissions and helping to combat climate change. By using modern data analysis techniques the BART project can optimize its operations, improve customer experience, and ensure the continued success of the system for generations to come.

G. Lessons Learned

- Learned how to use graph data models to solve real-world problems efficiently (Refer: Project Modules section)
- Learned how to use ETL tools for data warehousing(Refer: Project Modules section)
- Improved debugging capabilities and out-of-the-box thinking to solve problems to achieve proposed goals
- Polyglot persistence was utilized in our project by implementing MySQL and NoSQL(MongoDB and Neo4j) as data stores for specific modules. This optimized our application's performance, scalability, and data management flexibility(Refer: Project Modules section)

H. Innovation

1) Choosing unique ideas to solve real-world problems in BART.
2) Efficiently practicing polyglot persistence by selecting the right data store for different types of datasets.

I. Teamwork

Throughout the project, we ensured that the workload was distributed equally among all team members. We also made sure that all planning activities involved equal participation from all teammates.

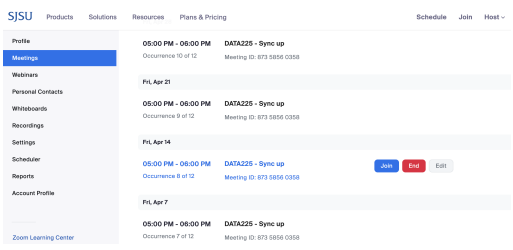
TASKS	ASSIGNED TEAM MEMBER
Data Gathering	Team
Data cleaning	Team
ERD modelling	Nikhil, Swati, Saketh
Graph modelling	Sowmya, Madhura
Database creation and population	Nikhil, Swati
Graph creation	Sowmya, Madhura
ETL	Nikhil, Saketh
Logic planning and Integration	Team
Mango DB analysis and aggregation pipeline	Swati, Nikhil
UI creation	Madhura, Sowmya
UI Integration with DB	Madhura, Sowmya
Data warehouse	Nikhil, Saketh
Visualization	Team
Reporting	Team

J. Technical difficulty

Cleaning the data, the BART data was not well structured Using new tools had some amount of a learning curve

K. Practiced pair programming?

We practiced pair programming using Slack, JIRA, zoom calls, and screen-sharing.



L. Practice agile/scrum (1-week sprints)?

We made use of JIRA to track each team-member task.

M. Used Grammarly or other tools for language

Used Grammarly for proofreading.

N. Slides

We created an interactive presentation for our term report using Microsoft PowerPoint.

O. Report

We ensured that our project report followed the IEEE LaTeXformat, and we paid careful attention to completeness, language, and plagiarism.

P. Used unique tools

Used LaTeX for writing reports, used Lucid Chart for creating DFD, Flowcharts, and visual paradigms for creating system architecture and use case diagrams (Refer: Project Module section), used Canva for slides.

Q. Performed substantial analysis using database techniques

We utilized Google BigQuery as our Data Warehouse Service to perform analytics on the BART, and we also used Mongo Atlas to perform analysis on BART yearly ridership data.

R. Used a new database or data warehouse tool

We used Google Bigquery and DTB for ETL

S. Used appropriate data modeling techniques

ER model for MySQL database and Graph data model for Neo4j

T. Used ETL tool

Used DBT to perform ETL operation from Google BigQuery and store the transformed data into Bigquery.

U. Demonstrated how Analytics support business decisions

We utilized MongoCharts and Tableau to create visuals that aid in analyzing the performance and other metrics of BART, enabling informed decision-making. Developed an app that calculates the fare and time for BART users, potentially impacting BART's business.

V. Used RDBMS

Used MySQL(Refer)

W. Used Data Warehouse

Used BigQuery(Refer)

X. Includes DB Connectivity / API calls

Used Python as our programming language to establish a connection between our application and the Neo4j database, utilizing the py2neo package. Implemented Flask to develop our user interface and make API calls to connect to the Neo4j backend.

REFERENCES

[1] Nance, Cory; Losser, Travis; Iype, Reenu; and Harmon, Gary, "NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH" (2013). *SAIS 2013 Proceedings. 27*

[2] Yunjie Zhao, Madhubabu Sandara, Shan Huang, Adel Sadek "Intelligent Transportation Systems Data Warehouses and Their Applications" *Conference: ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems, Volume 1, Beijing, China, 8-11 June, 2011*

[3] Chaudhuri, Surajit and Dayal, Umeshwar "An overview of data warehousing and OLAP technology" (1997) *ACM Sigmod record, ACM New York, NY, USA*

[4] Chew, Swee K.; Lepe, Alec; Tomkins, Aaron; and Scheirer, Peter (2020) "Forecasting San Francisco Bay Area Rapid Transit (BART) Ridership," *SMU Data Science Review: Vol. 3: No. 1, Article 11.*

[5] S. Uzunbayir, "Relational Database and NoSQL Inspections using MongoDB and Neo4j on a Big Data Application," *2022 7th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey, 2022, pp. 148-153*

[6] BART GTFS Data

[7] VTA GTFS Data

[8] BART Historical Ridership Data