DSC 202 - DATA MANAGEMENT FOR DATA SCIENCE

Health Management System

CHIRAG AGARWAL - A69034328
HEMANTH BODALA - A69037783
RAGHAV KACHROO - A69035155

CONTENT

INTRODUCTION

- The global healthcare data volume is projected to reach 10,000 exabytes by 2025, growing at a 36% annual rate. - Statista, 2024
- Health data is highly interconnected—activity levels, sleep patterns, and vital signs influence each other.
- Traditional relational databases struggle to model these complex relationships, resulting in fragmented insights.
- Graph databases like Neo4j are ideal for modeling complex, interrelated health data, enabling efficient querying and pattern detection.

OBJECTIVE

- Centralized Health Data Platform: Develop a system that consolidates and visualizes health data in a unified graph model.
- Anomaly Detection & Insights: Use graph-based querying to detect anomalies in health metrics.
- Health Insights: Generate personalized recommendations based on user-specific health patterns and thresholds.
- Scalable & Extensible: Design the system to be scalable—capable of integrating more health data sources or models in the future.

DATA SOURCES

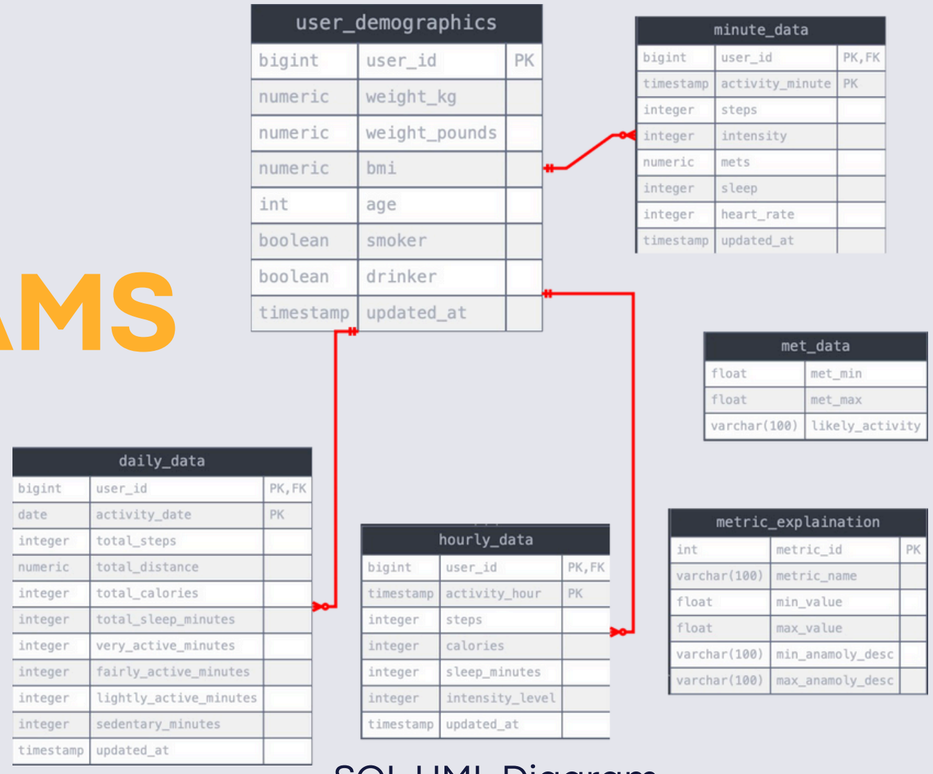
Kaggle – Fitbit Dataset - Link

- Content: Minute-level and hourly data (steps, intensity, METs, sleep, heart rate), daily aggregates (total steps, calories, sleep, active time), and user demographics (BMI, weight, age, lifestyle factors).
- Role: Used for real-world health metrics, provided baseline data for graph-based insights, and enabled realistic pattern modeling and anomaly detection.

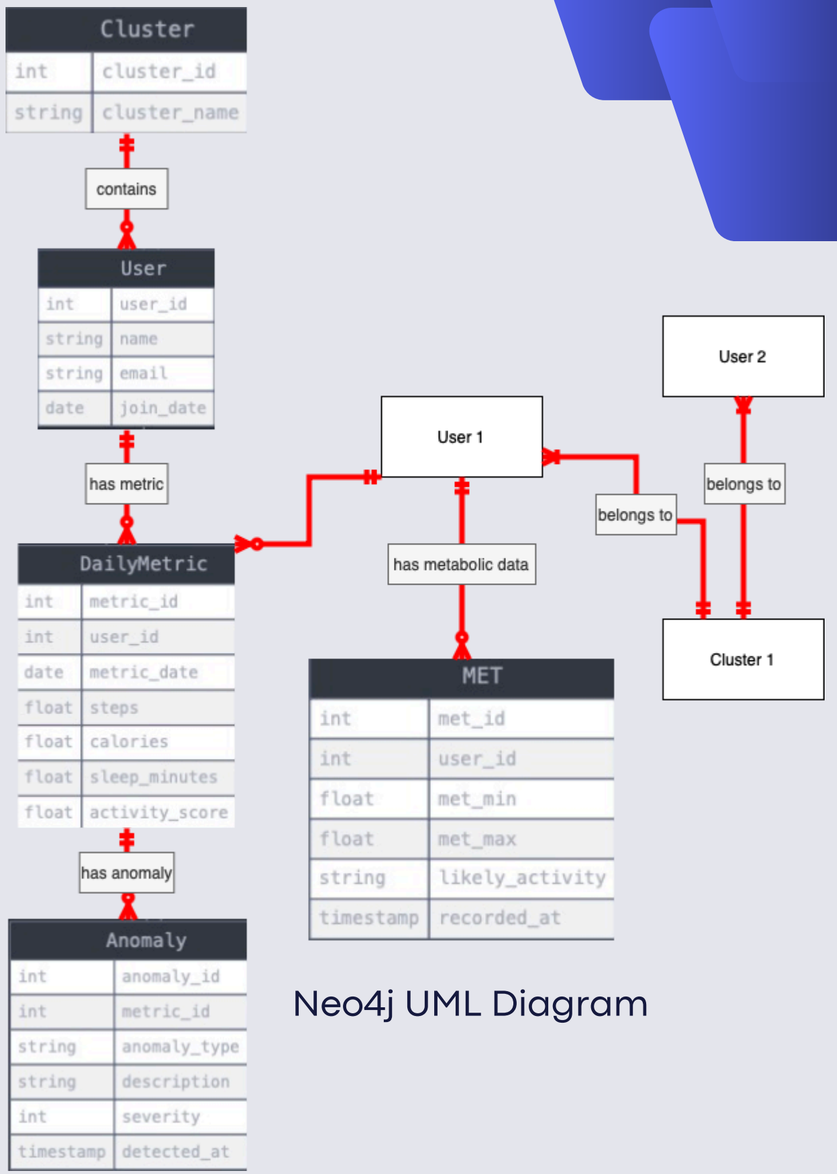
Enhanced with Synthetic Data

- Why: Expanded dataset to increase diversity and simulate rare health scenarios.
- How: Generated realistic variations in heart rate, steps, calories, BMI, and sleep using Python.
- Role: Boosted dataset size for robust testing, simulated edge cases, and improved anomaly detection.

UML DIAGRAMS



SQL UML Diagram



Neo4j UML Diagram

Database & User Relationship Models

- Our fitness tracking system uses six interconnected tables with user_demographics as the central entity linking to time-based activity data. Supporting tables provide metric explanations and metabolic equivalents for comprehensive analysis. Users are organized into behavioral clusters with a data flow from metrics to anomaly detection. For select users, detailed MET data enables precise activity recognition and personalized insights.

METHODOLOGY

Our methodology involved a multi-step process, starting with data collection and preprocessing, followed by integration into both relational and graph databases. We then applied complex querying and visualization techniques to extract meaningful insights, enabling advanced healthcare monitoring and social health interactions.

01

We collected Fitbit data and cleaned it by removing duplicates, invalid entries, and missing values. Data formats were standardized, converting text fields to integers and ensuring consistent timestamps. This produced a clean, structured dataset ready for database integration.

02

We stored the cleaned data in PostgreSQL with tables for user demographics, minute, hourly, and daily metrics. Relationships were established using primary and foreign keys, with indexing applied for faster querying. CRUD operations enabled dynamic data management. The result was a scalable, efficient relational database.

03

We imported the data into Neo4j, modeling users, metrics, and activities as nodes, with relationships like [:HAS_MINUTE_METRIC] and [:HAS_ANOMALY]. We added user clusters by age group, fostering social interactions where users could compare health metrics and share insights. This graph structure enabled dynamic relationship-based queries and pattern discovery.

04

Using Neo4j's visualization tools, we displayed real-time health metrics, detected anomalies, and visualized user connections. Cypher queries revealed at-risk users and community trends, turning raw data into actionable insights for better health monitoring.

TECHNOLOGIES USED

NEO4J

Neo4j is a graph database designed to efficiently store and query complex, interconnected data. In this project, it models health data relationships, enabling advanced insights into user activity patterns, anomalies, and social groupings. Cypher queries efficiently analyze health patterns, such as identifying users with similar activity.

PSQL

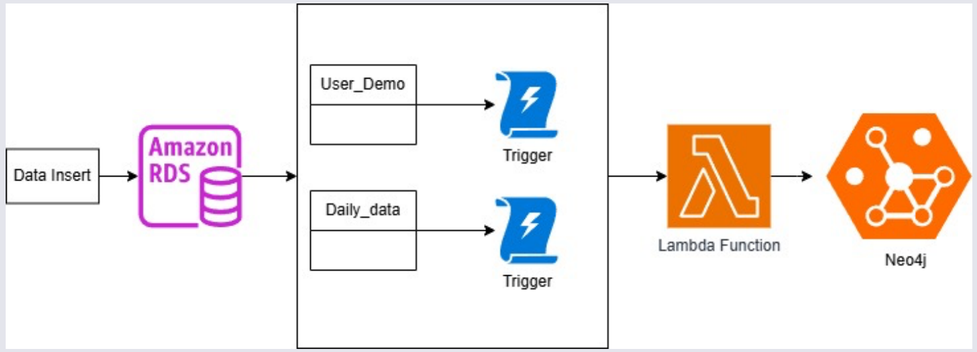
PostgreSQL (PSQL) is the relational database used to store structured health metrics. It handles data management, querying, and aggregation. The project uses PostgreSQL to store user demographics, minute, hourly, and daily metrics. SQL queries aggregate minute-level data into daily summaries for streamlined analysis.

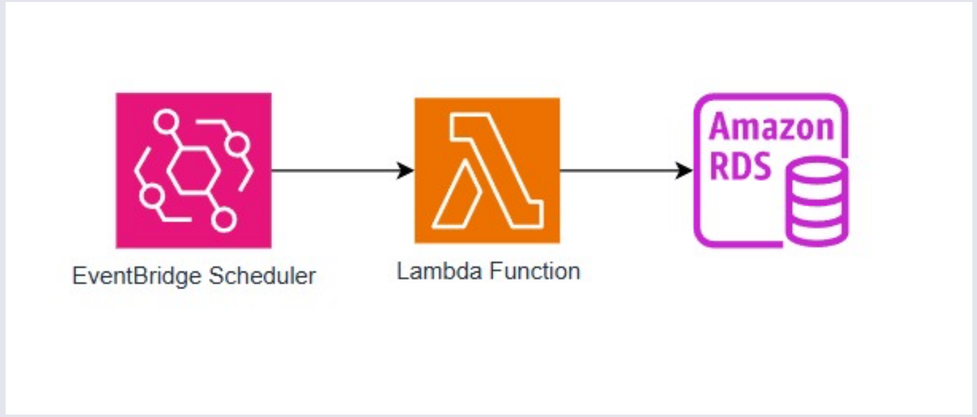
AWS (LAMBDA, CLOUDWATCH, RDS)

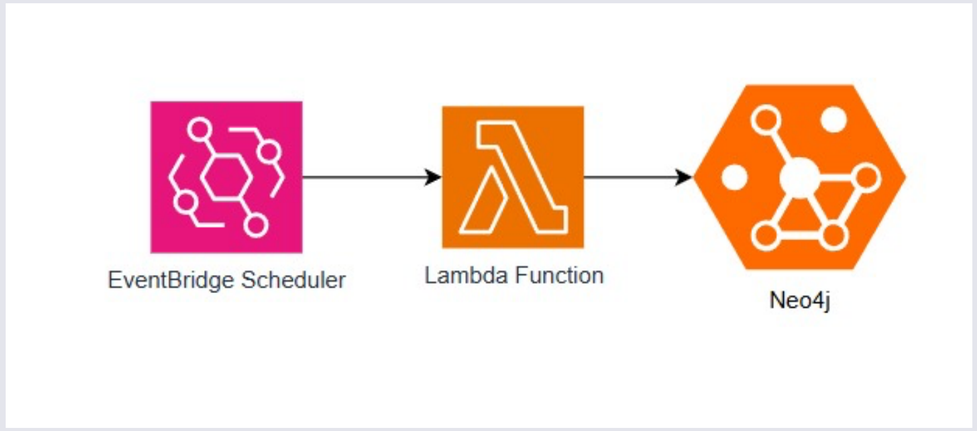
AWS provides cloud infrastructure for scalable deployment. Lambda executes serverless functions triggered by events, handling health data processing and scaling automatically. RDS hosts the PostgreSQL database securely, while CloudWatch monitors performance, tracks execution times, and logs errors.

PYTHON

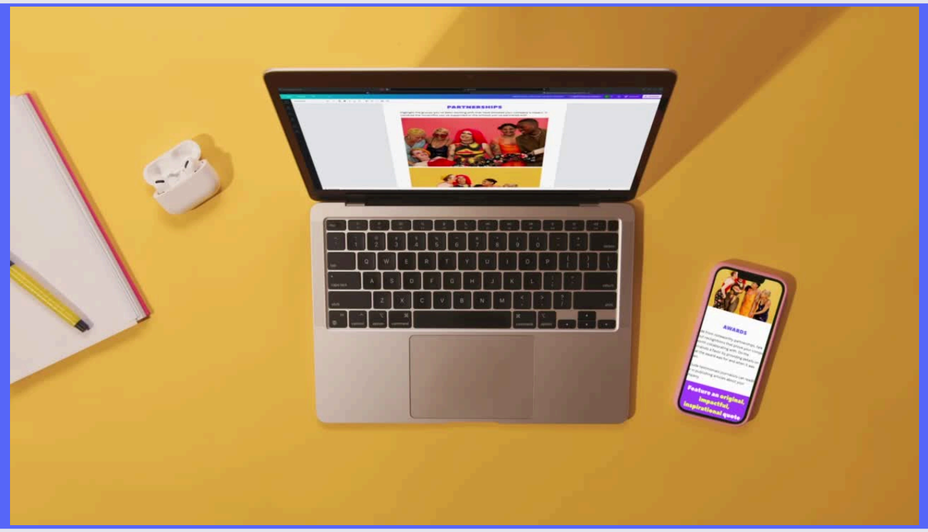
Python orchestrates data pipelines, automates queries, and integrates with PostgreSQL and Neo4j. It handles ETL (Extract, Transform, Load) processes, importing and cleaning data before storing it in the databases. Python scripts run SQL and Cypher queries, insert nodes and relationships into Neo4j, and automate data aggregation.







FUNCTIONAL DEMONSTRATION



- Case 1: Querying daily activity trends (steps, calories, sleep) to detect patterns.
- Case 2: Identifying health anomalies (abnormal BMI, heart rate spikes) with graph queries.
- Case 3: Generating personalized recommendations based on user metrics.
- Visualization: Real-time graph exploration of health data relationships.

WORK CONCLUSIONS

BEFORE OUR WORK

- Data Silos: Health data stored in separate tables made it difficult to detect patterns across metrics.
- Limited Anomaly Detection: Traditional relational models lacked the flexibility to track interconnected trends.
- Inefficient Analysis: Complex SQL joins made querying slow and cumbersome, limiting real-time insights.

AFTER OUR WORK

- Centralized Data Model: Neo4j's graph structure allows seamless exploration of interconnected health metrics.
- Enhanced Anomaly Detection: Efficiently detects outliers and trends by traversing relationships.
- Real-Time Insights: Faster, real-time querying for health patterns and personalized insights.

SCOPE FOR FUTURE WORK

While the current system effectively integrates relational and graph databases with user clustering functionality, there’s potential to expand functionality and improve insights further.

ENHANCED SOCIAL NETWORKING FEATURES

Introduce a chat system or message boards within clusters for better communication.
Develop a feature to recommend "Health Buddies" by analyzing shared metrics, goals, or activity patterns.
Introduce challenges or leaderboards within clusters to encourage group participation.

IMPROVED ANOMALY DETECTION

Implement real-time anomaly detection using machine learning models.
Develop alert systems that notify users about critical health conditions based on graph analysis.

PREDICTIVE ANALYTICS

Integrate predictive modeling to forecast potential health risks based on past data trends.
Introduce personalized recommendations for lifestyle improvements.

REAL-TIME DATA STREAMING

Enable streaming data integration for continuous updates of health metrics and anomaly detection.

THANK YOU

We hope this presentation provided valuable insights into how graph-based health management systems can transform wearable data into actionable health insights.