Research Document

Objective

The primary objective is to demonstrate that structuring chatbot conversation logs using a star schema improves analytics and chatbot performance compared to using raw unstructured logs or a normalized snowflake schema. Key goals include:

- **Data Transformation:** Convert raw, unstructured chatbot logs into a structured format using a star schema (fact table with multiple dimension tables) for efficient querying and analysis

- **System Implementation:** Design and build a star schema-based data warehouse alongside setups for unstructured log store and snowflake schema model for side-by-side comparisons

- **Pipeline Automation:** Develop ETL pipelines to extract entities and metrics from chatbot conversations and load them into the star schema, ensuring scalability and data freshness

- **Performance Evaluation:** Measure and compare query performance, scalability, and analytics usability across the three data modeling approaches

- **Chatbot Enhancement:** Integrate structured data models into a chatbot system to assess improvements in response quality and context awareness

Hypothesis

We hypothesize the following outcomes for using a star schema in chatbot data analytics:

- **Improved Query Performance:** Star schema implementation will significantly reduce query latency and accelerate analytical queries, even as data volume grows, due to fewer joins and simplified queries
- **Enhanced Analytics & Insights**: Transforming unstructured chatbot conversations into a structured star schema format will make it easier to extract actionable insights with higher accuracy and less effort than mining raw log text
- **Chatbot Performance Gains:** Feeding structured data from the star schema into the chatbot system will enable more context-aware and relevant responses by allowing quick retrieval of aggregated information
- **Efficient Scaling and ETL:** The automation of entity extraction and ETL processes for the star schema will reduce manual data preparation effort and scale efficiently to large volumes
- **Security and Data Integrity**: The structured approach will strengthen data governance and simplify implementation of access controls with fewer vulnerabilities compared to loosely managed log files.

DATASET

Data Model	Dataset Description
Unstructured Logs	We are using **synthetic dataset** of chatbot conversation logs stored as raw JSON/text format in NoSQL database (Firestore) without any structured schema or indexing.
Star Schema	Conversion of synthetic dataset into structured star schema format with central fact table (conversations) linked to dimension tables (User, Time, Intent, Channel) in BigQuery data warehouse.
Snowflake Schema	Conversion of synthetic dataset into normalized snowflake schema where dimension tables are further split into sub-dimensions (e.g., User dimension normalized into User and Location tables) requiring more joins for queries.

System Setup Overview

The system architecture involves several integrated components:

- **Chatbot Log Ingestion**: Raw conversation logs stored as unstructured text/JSON in NoSQL database (Firestore)

- **Entity Extraction:** NLP process parsing raw conversations to extract key entities and attributes (user intent, sentiment, keywords)

- **Star Schema Data Warehouse:** Central fact table recording conversations with metrics linked to multiple dimension tables (User, Time, Topic/Intent, Context) implemented in BigQuery Snowflake Schema Warehouse: Normalized version with sub-dimensions for comparison testing

- **ETL Pipeline:** Scheduled process using Apache Airflow to extract, transform, and load data into both structured schemas

- **Analytics Dashboard:** BI tool (Tableau/Power BI) connecting to data warehouses for interactive queries and visualizations

- **Chatbot Integration:** Structured data feeding back into chatbot system for enhanced responses

Evaluation Goals

The effectiveness and efficiency evaluation will focus on:

- **Query Latency**: Measure response time of analytical queries on each data model, expecting consistently low latency for star schema
- **Scalability:** Evaluate system performance and resource utilization as chatbot sessions grow from 10k to 1M+
- **Accuracy of Data Extraction:** Assess precision and recall of automated NLP entity extraction against ground truth
- **Usability & Insights**: Gauge user satisfaction and ease-of-use of analytics interface using System Usability Scale (SUS)
- **ETL and Automation Efficiency:** Track effort and time required to maintain each data model, quantifying reduction in manual work

Testing Plan

OLAP Query Performance	Benchmark query execution using representative analytical queries across varying dataset sizes (10k, 100k, 1M records)
Dashboard User Study	Conduct user study where participants use analytics dashboard connected to each data model to answer specific questions about chatbot data
Entity Extraction Accuracy Test	Validate NLP extraction by comparing output against manually curated sample, measuring precision and recall
Real-World Simulation	Stream new conversations to system as if chatbot is live, testing end-to-end stability and real-time update capabilities
Chatbot Response Impact	Evaluate chatbot's ability to answer meta-questions by querying its logs, measuring correctness and speed when backed by each data model

Threat Simulations to be Tested

Threat Simulation	Test Description
Data Poisoning Attack	Inject malformed or malicious data into input logs or ETL process to test schema resilience and validation.
Schema Manipulation	Attempt unauthorized modifications to star schema tables to verify access controls and **data integrity** safeguards.
Query Injection	Test system handling of harmful or non-optimal queries, including SQL injection attempts.
Privacy Leakage	Attempt to extract sensitive user information to ensure proper anonymization and access rules
ETL Pipeline Attacks	Introduce failures or delays in ETL workflow to test recovery mechanisms and data consistency
Dashboard Access Control	Test unauthorized access attempts to verify credential requirements and user role configurations

Metrics for Comparison

Metric	Description	Measurement Approach
Query Latency	Speed of queries across models	Average execution time (ms/seconds)
Storage Usage	Space used by each model	MB/GB per dataset
Entity Extraction Accuracy	Correctness of NLP entity detection	Precision, Recall, F1-score
Usability Score	Ease of analytics dashboards	SUS Score (0–100)
Chatbot Response Speed	Time taken by chatbot for meta-queries	Response time in ms
Error/Anomaly Detection	Ability to detect failures during queries or ETL	% errors identified and corrected

Steps to Perform:

- Collect raw chatbot logs (synthetic dataset).
- Store logs in NoSQL (Firestore).
- Run NLP-based entity extraction.
- Build ETL pipelines (Airflow) for transformation.
- Load data into Star Schema and Snowflake Schema in BigQuery.
- Connect schemas to BI dashboard (Tableau/Power BI).
- Run evaluation tests (latency, scalability, usability, security).
- Compare chatbot response quality using each model.
- Perform threat simulations.
- Record metrics and analyze results.

Expected Outcomes

- Star schema will outperform raw logs and snowflake schema in query performance, usability, and scalability.
- Snowflake schema will reduce redundancy but add design/query complexity.
- Raw logs will prove unsuitable for large-scale analytics.
- ETL automation will significantly reduce manual data preparation.
- Chatbot performance will improve when backed by star schema (faster, more context-aware responses).
- Security robustness will be stronger under structured schema (constraint validation, access controls).
- Final results will recommend star schema as the optimal model for chatbot analytics while noting conditions where snowflake or raw logs may be us