Jedha

Overview

Company: Stripe
Industry: Financial Technology (FinTech)
Founded: 2010
Headquarters: San Francisco, California
Main Product: Payment processing platform for businesses

Background: Stripe is a leading financial technology company that offers a comprehensive suite of products to manage payments and financial operations for businesses of all sizes. The company processes billions of transactions annually, supporting millions of merchants worldwide. As Stripe continues to grow, it faces increasing complexity in managing its data across various platforms, including transactional systems (OLTP), analytical systems (OLAP), and non-relational databases (NoSQL).

Stripe is looking to build a cutting-edge data infrastructure that seamlessly integrates these different types of data systems to support both operational excellence and advanced analytics. The goal is to create a unified, scalable data architecture that can handle high volumes of transactional data, support complex analytical queries, and manage unstructured data for advanced use cases like fraud detection, customer behavior analysis, and real-time recommendations.

Business Scenario

As Stripe expands globally, the company is encountering several challenges:

1. **Complex Transactional Workloads:**
   Stripe's OLTP systems handle millions of transactions daily, with high expectations for uptime, consistency, and speed. These transactions include payments, refunds, chargebacks, and subscription management. The company needs to ensure that these

systems are highly available, scalable, and capable of supporting new features like real-time fraud detection.

2. **Advanced Analytics Requirements:**
   Stripe needs to provide robust analytics to its clients and internal stakeholders. This includes revenue analysis, customer segmentation, compliance reporting, and performance tracking of various products. The company requires an OLAP system that can process complex queries efficiently, support ad-hoc analysis, and provide near real-time insights.

3. **Handling Unstructured and Semi-Structured Data:**
   Stripe collects vast amounts of unstructured and semi-structured data, such as logs, customer feedback, and interaction data from its website and mobile apps. This data is critical for building machine learning models for fraud detection, customer personalization, and predictive analytics. The company needs a NoSQL database that can efficiently store and process this data, integrate it with relational data, and provide flexible querying capabilities.

4. **Data Integration Across Systems:**
   Integrating data across OLTP, OLAP, and NoSQL systems is a significant challenge. Stripe needs a unified data architecture that allows seamless data flow between these systems, ensuring that data is consistent, up-to-date, and accessible for various use cases.

5. **Compliance and Security:**
   Given the sensitive nature of financial data, Stripe must ensure that its data architecture complies with regulations like GDPR, PCI-DSS, and CCPA. The company also needs to implement robust security measures to protect data at rest and in transit, manage access controls, and monitor data usage.

Business Requirements

1. **Transactional Integrity (OLTP):**
   Stripe's OLTP systems must handle high volumes of transactions with minimal latency. The system should support ACID properties to ensure data consistency and integrity across distributed environments. It must also support features like multi-document transactions, real-time data synchronization, and disaster recovery.

2. **Scalable and Efficient Analytics (OLAP):**
   The OLAP system must support complex queries across large datasets. It should be able

to aggregate data across multiple dimensions, support time-series analysis, and handle large-scale joins and subqueries. The system should provide low-latency responses for critical business queries and support both historical and near real-time analytics.

3. **Flexible Data Management (NoSQL):**
   Stripe needs a NoSQL system that can handle diverse data types, including JSON, XML, and binary data. The system should support flexible schema design, allow for efficient querying of nested and unstructured data, and integrate with other data systems to provide a holistic view of the data.

4. **Data Integration and Consistency:**
   The data architecture must ensure consistent data flow between OLTP, OLAP, and NoSQL systems. Stripe needs to implement data pipelines that can handle batch processing, streaming data, and real-time synchronization. The architecture should support ETL/ELT processes and allow for data transformation and enrichment.

5. **Security and Compliance:**
   The data architecture must include robust security measures, such as encryption, access controls, and audit logging. Stripe must ensure that its data storage and processing meet regulatory requirements across different jurisdictions. The system should support automated compliance reporting and real-time monitoring for security breaches.

Data Sources

1. **Transactional Data (OLTP):**
   Captures all financial transactions processed by Stripe. Fields include:

   - Transaction ID
   - Merchant ID
   - Customer ID
   - Transaction Amount
   - Currency
   - Payment Method
   - Transaction Date and Time
   - Location (IP-based geolocation)
   - Device Type (mobile, desktop, etc.)

- Status (successful, failed, refunded)
- Fraud Indicators (e.g., anomaly scores)

2. **Analytical Data (OLAP):**
   Aggregated and historical data used for reporting and analysis. Fields include:

   - Revenue Metrics (daily, weekly, monthly)
   - Customer Segmentation Data
   - Product Performance Metrics
   - Fraud Analysis Data
   - Compliance and Audit Logs

3. **Unstructured and Semi-Structured Data (NoSQL):**
   Data collected from various sources, such as logs, customer interactions, and machine-generated data. Fields include:

   - Log Data (error logs, access logs)
   - User Interaction Data (clickstream data, session data)
   - Machine Learning Features (input data for models)
   - Customer Feedback (reviews, survey responses)

4. **Reference Data:**
   Static or slowly changing data used across systems:

   - Country and Region Codes
   - Currency Exchange Rates
   - Merchant Information
   - Product Catalogs

Technical Requirements

1. **Data Model Design (OLTP, OLAP, NoSQL):**
   You are required to design data models for each type of data system:

   - **OLTP:** Focus on normalized schemas, transactional integrity, and performance optimization.

- **OLAP:** Design star or snowflake schemas for efficient querying and aggregation. Include pre-aggregated tables for common queries.
- **NoSQL:** Design flexible schemas that support unstructured data and allow for efficient querying and data retrieval.

2. **Data Pipeline Architecture:**
   Develop a data pipeline that integrates data across OLTP, OLAP, and NoSQL systems. The pipeline should support batch and real-time data processing, data transformation, and enrichment. Consider using tools like Apache Kafka for real-time data streaming and Apache Airflow for orchestrating ETL processes.

3. **Scalability and Performance Optimization:**
   Ensure that the data architecture can scale horizontally to handle increasing data volumes. Implement indexing, partitioning, and caching strategies to optimize performance across all systems. Consider the use of distributed databases and data sharding for high availability.

4. **Data Consistency and Synchronization:**
   Implement mechanisms to ensure data consistency across OLTP, OLAP, and NoSQL systems. Consider eventual consistency models where appropriate, and implement conflict resolution strategies for distributed data. Use CDC (Change Data Capture) techniques to keep data synchronized in real-time.

5. **Security and Compliance:**
   Design a security framework that includes data encryption, role-based access control, and audit logging. Ensure that the architecture complies with relevant regulations (e.g., GDPR, PCI-DSS). Implement automated compliance checks and reporting features within the data systems.

6. **Advanced Analytics and Machine Learning:**
   Integrate machine learning models within the NoSQL system for real-time fraud detection and customer personalization. Ensure that the data pipeline supports feature extraction and model deployment. Provide mechanisms for monitoring model performance and updating models as needed.

Task

As part of the data engineering team at Stripe, your task is to design a comprehensive data architecture that integrates OLTP, OLAP, and NoSQL systems. Your design should meet the company's needs for transactional integrity, advanced analytics, flexible data management, and regulatory compliance.

1. **OLTP Data Model:**

    - Design a normalized schema that supports high-volume transactional processing.
    - Ensure the model supports ACID properties and provides mechanisms for real-time data replication and failover.

2. **OLAP Data Model:**

    - Design a star or snowflake schema that supports complex queries and aggregations.
    - Propose a strategy for handling large-scale joins, subqueries, and time-series analysis.
    - Include a plan for pre-aggregations, materialized views, or summary tables to optimize query performance.

3. **NoSQL Data Model:**

    - Design a schema that can handle unstructured and semi-structured data.
    - Propose strategies for managing relationships between documents, including embedding, referencing, and indexing.
    - Ensure that the model supports integration with OLTP and OLAP systems.

4. **Data Integration Architecture:**

    - Develop a data pipeline that ensures consistent and real-time data flow between OLTP, OLAP, and NoSQL systems.
    - Include both batch processing and streaming components.
    - Propose tools and technologies for orchestrating and managing the data pipeline.

5. **Security and Compliance:**

    - Design a security framework that includes data encryption, access control, and audit logging.
    - Propose strategies for ensuring compliance with regulations like GDPR and PCI-DSS.
    - Develop a plan for automated compliance reporting and monitoring.

6. **Machine Learning Integration:**

    - Design a process for integrating machine learning models within the NoSQL system.
    - Propose strategies for real-time fraud detection, customer personalization, and predictive analytics.
    - Include mechanisms for monitoring and updating machine learning models.

Deliverables

- **Comprehensive Data Architecture Diagram:**
  A detailed diagram showing the integration of OLTP, OLAP

, and NoSQL systems, including data flows, pipelines, and data models.

- **ERD for OLTP System:**
  An Entity-Relationship Diagram showing the normalized schema for the transactional system.

- **Schema Design for OLAP System:**
  A detailed schema design for the OLAP system, including star or snowflake schemas, aggregation strategies, and query optimization techniques.

- **NoSQL Data Model:**
  A detailed schema design for the NoSQL system, including strategies for managing unstructured data, relationships, and indexing.

- **Data Pipeline Architecture:**
  A technical document describing the data pipeline architecture, including tools, technologies, and processes for data integration and synchronization.

- **Security and Compliance Plan:**
  A detailed plan outlining the security measures, compliance strategies, and monitoring tools to be implemented across the data architecture.

- **Machine Learning Integration Strategy:**
  A document describing the integration of machine learning models within the NoSQL system, including feature extraction, model deployment, and performance monitoring.

- **SQL and NoSQL Queries:**
  A set of SQL and NoSQL queries demonstrating how key business questions (e.g., revenue

analysis, fraud detection, customer segmentation) can be answered using the proposed data models.