ISOM 671: Managing Big Data Group Project

Alibaba E-Commerce Database System Design and Applications

Team 11

Roffy Shan
Junior Rutamu
Jiayue (Jaycee) Tang
Hsiao-chun (Audrey) Lai

Summary

This project centers on Alibaba, which is the largest e-commerce platform in China. It provides high-quality services to billions of customers and collects massive e-commerce transaction data. We focus on advertisement-related data derived from the "Alimama Search Ad Conversion Prediction Dataset" on Alibaba's Tianchi platform. The dataset includes information on advertised products, users, shops, and clicked samples, capturing whether Ad interactions led to transactions. To handle this data effectively, we adopted a comprehensive database design integrating MySQL RDBMS, NoSQL MongoDB, and a cloud-based HDFS system, ensuring scalability and efficiency across diverse data types and processing needs.

Our innovative feature, the Alibaba Shop Analytics Platform (ASAP), shifts the analytical focus from Ad effects on customers to empowering sellers with actionable insights. The platform enables sellers to monitor sales trends associated with periods, identify profitable product segments, and optimize inventory management. By querying the data warehouse, ASAP provides valuable metrics on campaign effectiveness, user engagement patterns, and risk indicators, equipping sellers with tools to make informed, data-driven decisions in the competitive e-commerce landscape.

Existing Systems

The database system is designed to meet Alibaba's data needs for daily operations, decision-making, and service enablement. First we implement the RDBMS using MySQL for structured data, where the clicked_samples table serves as the central table, connecting users, items, shops, and contexts by their primary keys for comprehensive analysis.

For daily operations, the clicked_samples table tracks user interactions, while the items table monitors inventory and sales trends, and the shops table evaluates shop performance through reviews and service quality. For decision-making, the users table combines demographic data with transactional data from clicked_samples to target ads effectively, while data from the shops and items tables informs strategies to improve services and promote high-performing products.

A complementary NoSQL MongoDB database is used to handle the semi-structured data present in our dataset. For instance, item categories and properties columns containing lists or dictionaries, are challenging to manage within a relational database. Therefore, MongoDB's document-based structure provides flexible schemas and allows for storing such data as JSON-like documents, preserving their hierarchical format.

Finally, our cloud-based Hadoop Distributed File System (HDFS) ensures scalability for processing large-scale clickstream and transactional data. By centralizing data from MySQL and MongoDB, HDFS provides a robust foundation for big data analytics, enabling efficient data exploration and modeling to support strategic decision-making and optimize further business applications.

Challenges during implementation included ensuring efficient JOINs in the RDBMS to handle high-volume queries, and managing data pipeline latency when integrating with the HDFS for real-time analytics.

New Business Application: Alibaba Shop Analytics Platform (ASAP)

Our application distinguishes itself by focusing on the seller's perspective, whereas the current recommendation system primarily targets buyers. ASAP provides sellers with valuable insights into key areas such as item sales trends, user targeting, and risk detection, enabling them to optimize their inventory, tailor marketing strategies, and proactively address potential advertising and sales risks. The functions of this platform are as follows:

1. Item Sale Trend: This feature helps sellers identify best-selling products and seasonal demand fluctuations. By leveraging these insights, sellers can optimize inventory and pricing strategies to maximize profits.
2. User Targeting: User targeting provides sellers with tools to identify and segment profitable customer groups based on demographics, purchase behavior, and preferences. Sellers can then design tailored marketing campaigns to reach the right audience effectively, enhancing conversion rates (CVR).
3. Risk Detecting: This feature evaluates potential risks in advertising and sales performance, such as declining product demand or oversaturated customer segments. By identifying these issues early, sellers can take proactive measures to mitigate losses and ensure sustainable growth.

Tableau serves as the primary visualization tool, converting the output from Hive SQL into interactive dashboards and dynamic reports. Then Streamlit is implemented as the front-end application interface, offering a user-friendly environment for clients—primarily sellers on the Alibaba platform—to interact with the analytics platform. With Streamlit, users can view personalized dashboards, access actionable insights, and run dynamic queries with minimal technical expertise. The platform provides real-time interactivity, enabling clients to filter data by Ad campaigns, sales categories, or customer demographics and instantly view results.

This combination of tools ensures a streamlined workflow, from data ingestion and processing to visualization and client-facing interactions. Hive SQL and Tableau handle complex data processing and visualization, while Streamlit delivers an engaging and accessible user experience. Together, they empower clients to make data-driven decisions, enhance their advertising strategies, and optimize inventory management with ease.

Decision Making and Insights

The star schema includes a central fact table, fact_sales, which records aggregated sales data at the intersection of four dimensions: dim_user, dim_shop, dim_time, and dim_product. This schema is optimized with the fact table containing keys (time_key, item_id, user_id, shop_id) linking to dimensions and metrics like total_views, total_purchases, and conversion_rate. Each dimension table provides descriptive attributes, such as user demographics, shop performance, time characteristics, and product details.

To build the analytical platform, we designed an ETL pipeline from transactional sources into a star schema-based data warehouse. The pipeline extracts raw data from MySQL and MongoDB, then performs key transformations, including calculating aggregated metrics like total views, total purchases, and CVR for the fact table and enriching dimension tables with standardized attributes for consistency. Finally, the transformed data is loaded into the star schema in the data warehouse.

Based on key queries, our actionable insights for Alibaba are as follows:

1. Inventory Management: Sellers can see products with the highest CVR to stock more of their best-selling items, ensuring availability to meet demand and capitalize on high-performing products.
2. Customer Relationship Management: Sellers can engage top customers based on total purchases with personalized promotional messages, exclusive discounts, or early access to new products to strengthen relationships and encourage repeat purchases.
3. Targeting Promotion: ASAP identifies customers who view a shop frequently but do not purchase. Sellers can target these potential buyers with strategic incentives, such as personalized coupons or limited-time offers, to convert them into active buyers and improve overall CVR.

While the Alibaba Shop Analytics Platform (ASAP) provides valuable insights, several challenges must be addressed to enhance its effectiveness. First, the system's recommendation of best-selling products may not align with high-margin items, leading to increased sales but not necessarily improved profitability for sellers. Second, the system defines "most valuable customers" based solely on purchase frequency, potentially overlooking high-value customers who make fewer but more substantial transactions or exhibit strong brand loyalty. This narrow definition risks missing critical opportunities to engage customers with high lifetime value. Lastly, while promotions are effective in attracting price-sensitive consumers, they may inadvertently encourage transactional rather than relational behavior, as these consumers are likely to purchase only during promotional periods, limiting the potential for building long-term customer loyalty.

Exhibits

```
CREATE TABLE clicked_samples (
    instance_id BIGINT PRIMARY KEY,
    is_trade TINYINT,
    item_id BIGINT,
    user_id BIGINT,
    context_id BIGINT,
    shop_id BIGINT,
    INDEX idx_is_trade (is_trade)
);

LOAD DATA LOCAL INFILE 'D:/emory/fall/bigdata/group_project/clicked_sample.csv'
INTO TABLE clicked_samples
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

Figure 1: MySQL Code

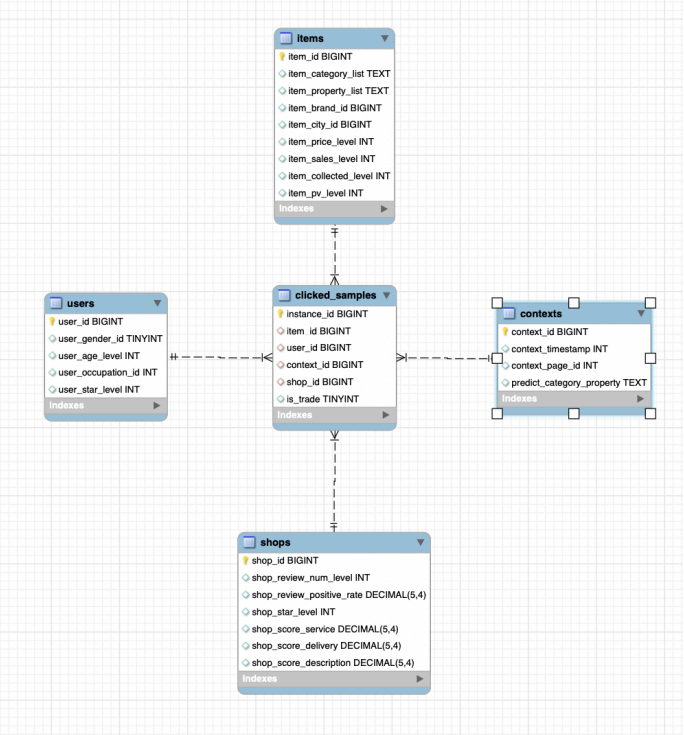


Figure 2: ER Model for RDBMS

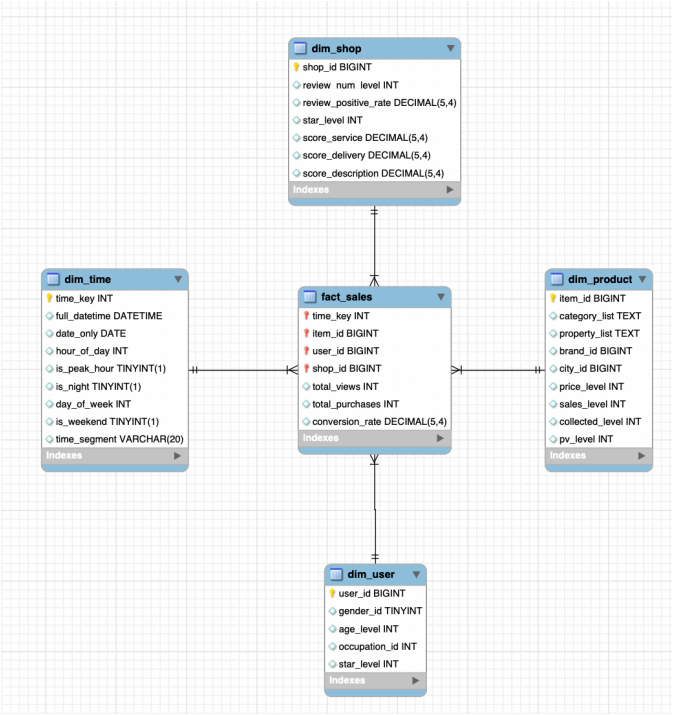


Figure 3: STAR Schema

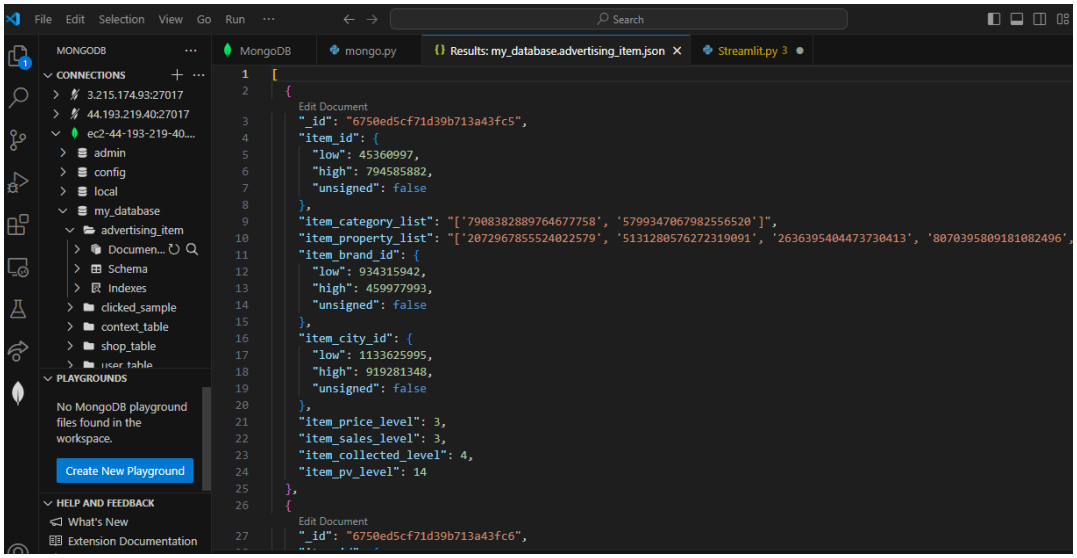


Figure 4: No SQL Mongo DB Instance

```sql
WITH ShopProductConversionRanking AS (
    SELECT
        shop_id, item_id, SUM(total_views) as total_shop_item_views, SUM(total_purchases) as total_shop_item_purchases,
        CASE
            WHEN SUM(total_views) > 10 THEN SUM(total_purchases) * 1.0 / SUM(total_views) ELSE 0
        END as overall_conversion_rate,
        RANK() OVER (PARTITION BY shop_id ORDER BY
                CASE WHEN SUM(total_views) > 10 THEN SUM(total_purchases) * 1.0 / SUM(total_views) ELSE 0
                END DESC) as conversion_rank
    FROM fact_sales GROUP BY shop_id, item_id
)
SELECT
    shop_id, item_id, total_shop_item_views, total_shop_item_purchases, overall_conversion_rate
FROM ShopProductConversionRanking
WHERE conversion_rank = 1
ORDER BY overall_conversion_rate DESC;
```

Figure 5: Key Query 1

```sql
WITH RankedUsers AS (
    SELECT shop_id, user_id, SUM(total_purchases) AS total_purchases,
        ROW_NUMBER() OVER (PARTITION BY shop_id ORDER BY SUM(total_purchases) DESC) AS user_rank
    FROM fact_sales
    GROUP BY shop_id, user_id)
SELECT
    shop_id, user_id AS most_valuable_user, total_purchases
FROM RankedUsers
WHERE user_rank = 1 and total_purchases >=1;

    shop_id, item_id, total_shop_item_views, total_shop_item_purchases, overall_conversion_rate
FROM ShopProductConversionRanking
WHERE conversion_rank = 1
ORDER BY overall_conversion_rate DESC;
```

Figure 6: Key Query 2

```sql
WITH RankedUsers AS (
    SELECT shop_id, user_id, SUM(total_views) AS total_views,
        ROW_NUMBER() OVER (PARTITION BY shop_id ORDER BY SUM(total_views) DESC) AS user_rank
    FROM fact_sales
    WHERE total_purchases = 0
    GROUP BY shop_id, user_id
)
SELECT
    shop_id, user_id AS most_view_user, total_views
FROM RankedUsers
WHERE user_rank = 1 AND total_views >= 1;
```

Figure 7: Key Query 3

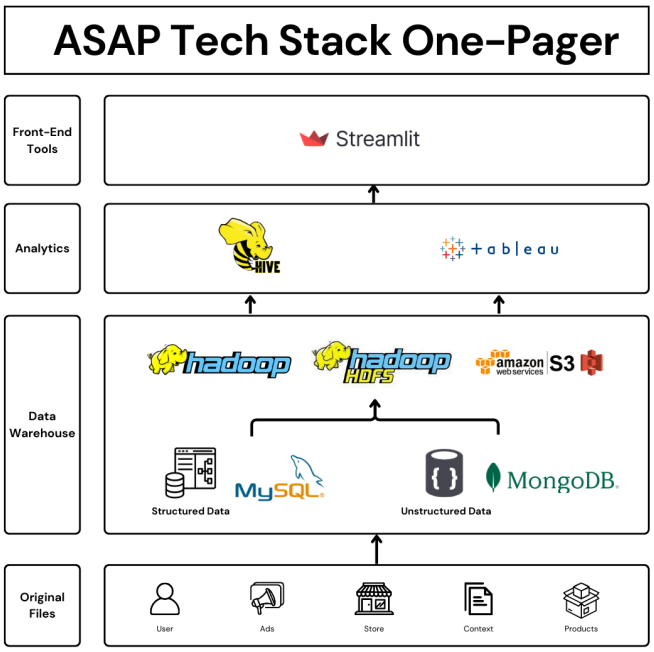


Figure 8: Application Architecture Diagram

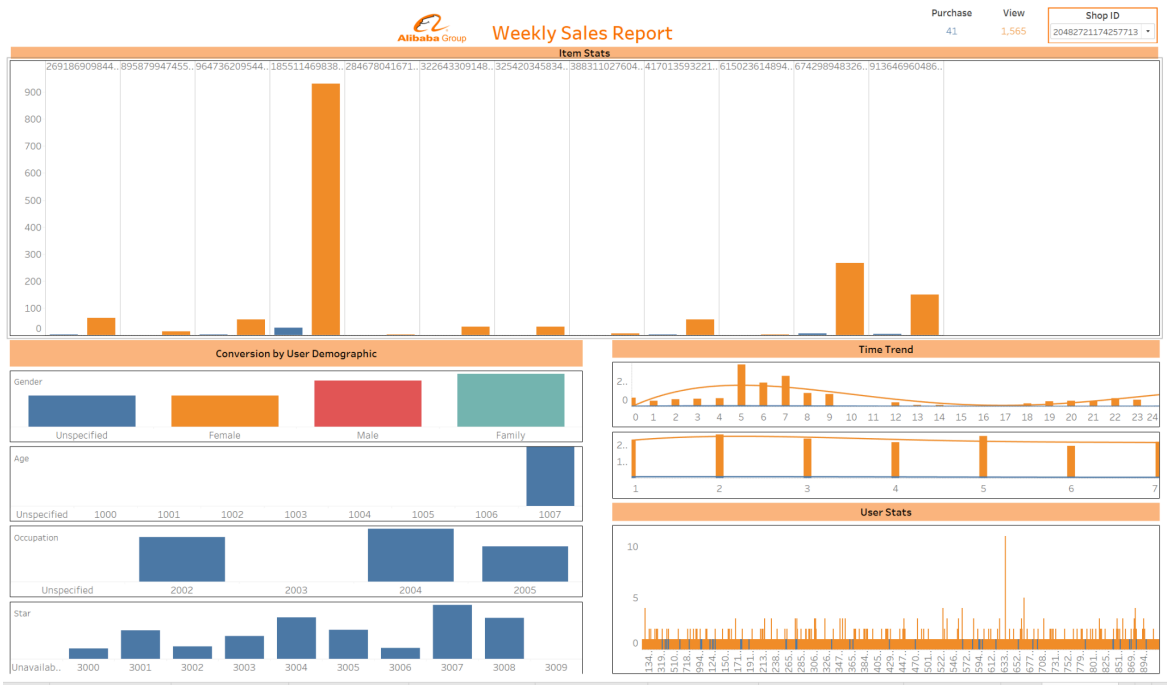
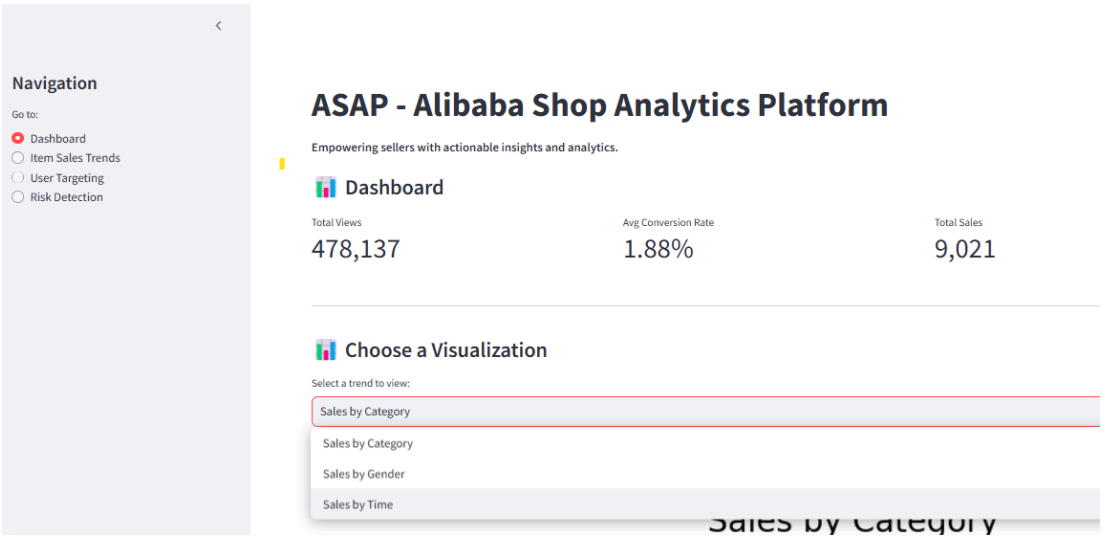


Figure 9: Analytical Dashboard Interface



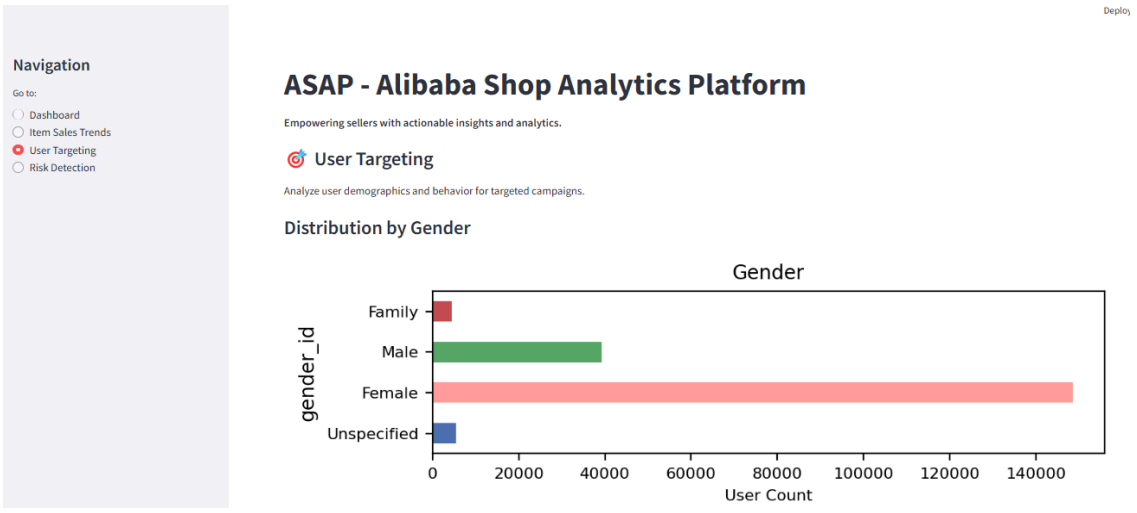


Figure 10: Front-End Tool Interface