**Relational Database Analytics Platform for Ecommerce**

Team: Owen Randolph, Gabe Tharp and Marcos Fernandez

**Purpose & Motivation**

Today’s ecommerce businesses create huge amounts of transactional data. This project utilizes Online Transaction Processing (OLTP) as a scalable, secure and production-ready solution to track and store ecommerce sales data. Business leaders also need a reliable Online Analytical Processing (OLAP) system specifically built for ecommerce to analyze product performance, understand trends, segment customers, customer behavior, etc.

As more sales as all kinds move online, it’s more important than ever to be able to leverage database technology to operate a business. This project combines secure transaction processing with the ability to analyze and visualize data.

**Target users & Benefits**

The OLAP portion of our project is primarily used by sales and business intelligence (BI) analysts to run queries based on business questions for general status reporting and special business questions. For example, customer buying pattern data could be used by a marketing team to strategically place ads or optimize the product or product category in a targeted marketing campaign. In short, better understanding our customers to drive engagement and use of our ecommerce site.

Data scientists could use processed data to build models for customer behavior prediction and customer segmentation that could further empower the company to retain customers and gain an edge over competitors. This data can be used in advance machine learning models in site features, such as personalized recommender systems, that could further drive sales and engagement.

Downstream business and sales leaders use these data reports to understand key performance indicator (KPI) use. The dashboard is a visual way to easily understand high level data through trends, geographic areas of opportunity or concern, profit margins, best-selling products, etc.

Finally, a developer/IT team could integrate our system into the infrastructure of the company, using CI/CD pipelines to automate data ingestion, report creation, and monitoring. The architecture could also be expanded into inventory tracking, supply chain/logistics tracking, customer service needs, real-time notifications for customers or business stakeholders, among many other features.

**Data source:**

https://www.kaggle.com/datasets/abdul0haadi/e-commerce-sales-dataset-csv?resource=download

**Database structure & preprocessing:**

File: [E-commerce sales dataset .csv](https://www.kaggle.com/datasets/abdul0haadi/e-commerce-sales-dataset-csv?resource=download)

Proposed Table Name: Orders

Key Fields: Row ID, Order ID, Customer ID, Product ID

Columns in dataset:Row ID, Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country, City, State, Postal Code, Region, Product ID, Category, Sub-Category, Product Name. Sales, Quantity, Discount, Profit

**Data Cleaning:**

* Check for missing values, handle missing values if needed
* Format dates from DD/MM/YYYY to YYYY-MM-DD for MySQL.
* Drop Country column (all orders are within the United States).
* Format Sales and Profit column values to 2 significant figures accounting for cents

**Preliminary Design:**

Following normalization, the design of the database will follow a star schema as follows, with table name primary keys (PK), foreign keys (FK), and attributes:

* Fact Table: Sales: Row\_Id (PK), Order ID (FK), Customer ID (FK), Product ID (FK)
* Dimension Table: Orders: Order\_ID (PK), Order\_Date, Ship\_Date, Ship\_Mode
* Dimension Table: Customers: Customer\_ID (PK), Customer\_Name, Segment, City, State, Postal\_Code, Region
* Dimension Table: Products: Product\_ID (PK), Category, Sub\_Category, Product\_Name

**Database Features:**

* One-to-many relationship: customer -> many orders
* One-to-many relationship: product -> many orders
* One-to-many relationship: order -> multiple products

**Project Organization**

**Database design**

Using MySQL Workbench - Database design for loading data, design schema, clean/transform data, normalization, ER diagram, queries

**Backend**

Launch in a Virtual Private Cloud (VPC)

**OLTP**

Export the schema and data to be imported into AWS Aurora MySQL

This will be our cloud OLTP database

Aurora can handle real-time transactions like new orders and updating customer data

\*API Gateway could be added for data importing from the internet

**OLAP**

ETL?

Amazon Redshift data warehouse

**Visualization & Deployment**

From Redshift -> Quicksight for dashboard analytics

WEB UI dashboard with Quicksight

**Technology stack:**

We will be using a relational database because the dataset is organized in such a way that it can be normalized and queried. This will lead to better efficiency when running queries, efficient reads and writes. Ensure atomicity for sales transactions

MySQL Workbench for loading data, design schema, clean/transform data, normalization, ER diagram, queries

AWS Aurora – high throughput, high volume relational database engine

AWS Redshift – data warehouse

AWS Quicksight – Dashboard analytics

Internally (IAM-authenticated users), Externally (via embedded dashboards in apps or websites

**Timeline:**

Schema design

Query/function development

Interface or script prototyping (Note for Option 2: you must design a script that can be replicated

Integration & testing

Demo preparation

References:

Lucid.app (ER Diagram)

**Appendix**

ER Diagram

A diagram of a data flow

AI-generated content may be incorrect.