**Relational Database and 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 of 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 indicators (KPI). 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: Sale\_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: customer -> many sales
* One-to-many relationship: order -> many sales
* One-to-many relationship: product -> many sales
* Many-to-many relationship: products <-> orders

**Project Organization**

**Team Roles & Expertise**

* Gabe Tharp: data analysis lead, db design, OLAP
* Marcos Fernandez: data visualization dashboard lead, db design, backend scripts
* Owen Randolph: cloud architecture and deployment lead, db design, project management

**Database Design Workflow**

Using MySQL Workbench – create database using MySQL, clean/transform data using Python with pandas library for use in database, load data from csv file, use reverse engineering to normalized schema mentioned above, create a formal Entity Relationship (ER) diagram, test the database with queries to ensure integrity and no duplicate values.

**Backend**

Once the database is created in MySQL Workbench, we will migrate it to Amazon Web Services (AWS) for hosting and added features. It will be hosted in a Virtual Private Cloud (VPC) with security features like Identity & Access Management.

**OLTP**

We will export the schema and data to be imported into Amazon Aurora MySQL. This will be our cloud-based OLTP database engine. Aurora can handle real-time transactions like new orders and updating customer data, as a high-performing, parallel-processing database engine.

\*Amazon API Gateway could be added to allow endpoints for data importing from an outside source

**OLAP**

We may need to do Extract, Transform, Load (ETL) process using AWS Glue to transform data to a columnar data structure for data analytics. Columnar data will improve the performance of queries when scaling the database to include massive amounts of data.

At this point such data will be stored in a data warehouse, Amazon Redshift. This data warehouse service can do complex analytical queries and create BI reports.

**Visualization & Deployment**

From Redshift the data can be used by Amazon QuickSight for dashboard BI analytics.

QuickSight has interactive dashboards that can filter, drill down, and of course show a variety of data visualizations to end users for analysis. It is also capable of being shared via a URL and with user permissions within AWS. We will use a direct URL for user-based access. Quicksight generates a URL which can be copied and shared.

**Technology stack:**

Database Type: We will be using a relational database because the dataset is organized in such a way that it can be normalized and queried. It has organized, tabular data and is structured in an inherently relational way. This will lead to better efficiency when running queries, efficient reads and writes. This also ensures compliance with ACID properties which are necessary for sales transactions.

* Jupyter Lab - clean/transform data in preparation for loading to MySQL Workbench
* MySQL Workbench - loading data, database design, schema design, normalization, ER diagram, testing queries
* Amazon Aurora MySQL– high throughput, high volume relational database engine
* AWS Glue - ETL
* Amazon Redshift – data warehouse
* Amazon Quicksight – dashboard analytics
* Amazon Identity & Access Management (IAM) – security features
* Amazon Virtual Private Cloud (VPC) – secure cloud hosting environment
* Other services such as storage buckets (Amazon S3 buckets) and workflow automation functions (AWS Lambda) may be used to satisfy data pipeline architecture requirements

**Timeline:**

|  |  |  |
| --- | --- | --- |
| **Timeline** | | |
| **Phase** | **Target Date** | **Week** |
| Schema Design | 11-Jul | 9 |
| Query/Function Development | 11-Jul | 9 |
| Cloud Service Setup | 18-Jul | 10 |
| Interface | 18-Jul | 10 |
| Integration & Testing | 25-Jul | 11 |
| Demo Preparation | 25-Jul | 11 |
| Demo & Final Report | 1-Aug | 12 |

**References**

Amazon Web Services. (n.d.). *Amazon QuickSight: Business intelligence and dashboard service*. AWS. <https://aws.amazon.com/quicksight/>

Amazon Web Services. (n.d.). *The difference between OLAP and OLTP*. AWS. <https://aws.amazon.com/compare/the-difference-between-olap-and-oltp/>

GeeksforGeeks. (2022, October 4). *Normal forms in DBMS*. GeeksforGeeks. <https://www.geeksforgeeks.org/dbms/normal-forms-in-dbms/>

GeeksforGeeks. (2022, October 3). *Star schema in data warehouse modeling*. GeeksforGeeks. <https://www.geeksforgeeks.org/dbms/star-schema-in-data-warehouse-modeling/>

Haadi, A. (2022). *E-commerce sales dataset [CSV]* [Data set]. Kaggle. <https://www.kaggle.com/datasets/abdul0haadi/e-commerce-sales-dataset-csv>

Lucid Software Inc. (n.d.). *Lucidchart* [Diagramming application]. <https://www.lucidchart.com/>

Microsoft. (2023, January 25). *Understand star schema and the importance for Power BI*. Microsoft Learn. <https://learn.microsoft.com/en-us/power-bi/guidance/star-schema>

Sprinkle Data. (2023, May 8). *OLTP vs OLAP: What’s the difference?* SprinkleData. <https://www.sprinkledata.com/blogs/oltp-vs-olap>

**Appendix**

ER Diagram

A diagram of a data flow

AI-generated content may be incorrect.