**Project 1 (Project SQL): Relational Database Management System (RDBMS) Design & Implementation using PostgresSQL**

**INTRODUCTION**

**Project Title:**

RDBMS Design and Implementation using SQL

**Case Scenario:**

XYZ SuperStore is launching a new online shopping platform that offers a diverse range of products including cosmetics, electronics, and jewellery. The platform aims to provide a seamless shopping experience by capturing and analyzing user behavior, purchase history, and events across different product categories.

In order to achieve the aforementioned ambition, the client is looking into consolidating historical sales data and extracting insights from them using

a well-designed relational database and seamlessly integrating it to an analytical platform for a real time insight mining.

**Objective:**

* Extract historical data, transform and load to a postgres SQL database using python.
* Design a relational database schema and implement it using postgres SQL.
* Connect the data to PowerBI Analytical platform, design a sales dashboard.

METHODOLOGY

Requirement gathering phase

1. **Technologies Used:**

PostgresSQL, PowerBI

**2. Dataset:**

Source: https://www.kaggle.com

Name & *Link****:***

* [E-Commerce Events History in Cosmetics Shop](file:///C:\Users\admin\AppData\Roaming\Microsoft\Word\eCommerce%20Events%20History%20in%20Cosmetics%20Shop%20(kaggle.com))
* [eCommerce behavior data from multi category store](https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store)
* [eCommerce purchase history from electronics store](https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-electronics-store)
* [eCommerce purchase history from jewelry store](https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-jewelry-store)
* [eCommerce events history in electronics store](https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-electronics-store)

***Description:***

1. E-Commerce Events History in Cosmetics Shop: This 5-month dataset (Oct 2019 – Feb 2020) contains e-commerce events data from a medium cosmetics shop. The data is a many-to-many relation between products and users.
2. eCommerce behavior data from multi category store: This file contaisn behavior data for 7 months (from October 2019 to April 2020) from a large multi-category online store. Each row in the file represents an event. All events are related to products and users. Each event is like many-to-many relation between products and users.
3. eCommerce purchase history from electronics store: This file contains purchase data from April 2020 to November 2020 from a large home appliances and electronics online store. Each row in the file represents an event. All events are related to products and users. Each event is like many-to-many relation between products and users.
4. eCommerce purchase history from jewelry store: This file contains purchase data from December 2018 to December 2021 (3 years) from a medium sized jewelry online store. Each row in the file represents a purchased product. Several products from the same order/purchase are listed in separate lines and joined by order\_id field.
5. eCommerce events history in electronics store: This file contains behavior data for 5 months (Oct 2019 – Feb 2020) from a large electronics online store. Each row in the file represents an event. All events are related to products and users. Each event is like many-to-many relation between products and users.

***File structure:***

* event\_time: Time when event happened at (in UTC).
* event\_type: Only one kind of event: purchase.
* product\_id: ID of a product
* category\_id: Product's category ID
* category\_code: Product's category taxonomy (code name)
* brand: Downcased string of brand name. Can be missed.
* price: Float price of a product. Present.
* user\_id: Permanent user ID.
* :user\_session: Temporary user's session ID. Same for each user's session. Is changed every time user come back to online store from a long pause.

***Event types***

* view - a user viewed a product
* cart - a user added a product to shopping cart
* remove\_from\_cart - a user removed a product from shopping cart
* purchase - a user purchased a product

Conceptual design phase

**3. Database:**

1. Requirements:

Cosmetics Shop:

Track user interactions with cosmetics products.

Capture events such as views, cart additions, and purchases.

Store information about cosmetics products, brands, and categories.

Multi-Category Store:

Centralized database for user behavior across multiple categories.

Include data from cosmetics, electronics, and jewelry stores.

Capture events, products, and user details for comprehensive analysis.

Electronics Store:

Record purchase history specific to the electronics category.

Track user behavior, events, and interactions with electronics products.

Jewelry Store:

Manage purchase history and events for the jewelry category.

Include information about different types of jewelry products.

Multi-Category Events (Electronics):

Focus on events history for the electronics category.

Record user interactions, views, and cart additions for electronics products.

1. Design:
2. Tables:

CosmeticsEvents, CosmeticsProducts, CosmeticsBrands, CosmeticsCategories

MultiCategoryEvents, MultiCategoryProducts, MultiCategoryUsers

ElectronicsPurchaseHistory, ElectronicsEvents, ElectronicsProducts

JewelryPurchaseHistory, JewelryEvents, JewelryProducts

MultiCategoryEventsElectronics

1. Relationships:

Establish relationships between tables based on common keys (e.g., product\_id, user\_id).

Ensure referential integrity for smooth data management.

1. Normalization:

Apply normalization techniques to avoid data redundancy.

Ensure that each table serves a specific purpose.

Logical design phase

Physical design phase

1. Indexes:

Add indexes to columns frequently used in queries for optimized performance.

1. Constraints:

Define primary keys, foreign keys, and check constraints.

Enforce data integrity rules to maintain the quality of the data.

1. Views (Optional):

Create views to simplify complex queries or provide summarized information.

SQL Implementation and loading phase

Use SQL queries to insert, update, and retrieve data from the database.

Leverage PostgreSQL for its robust support of relational database features.

Testing and evaluation phase

Analysis:

Analyze user behavior across different categories.

Understand popular products, brands, and user demographics.

Identify trends in purchase history and events.

PowerBI Integration:

Integrate the PostgreSQL database with PowerBI for dynamic visualizations.

Create interactive dashboards showcasing key performance indicators.

**Challenges Faced:**

Discuss any challenges encountered during the database design and implementation.

**Learnings:**

Highlight key learnings and insights gained from working with SQL databases.

**Conclusion:**

The relational database design and implementation will provide the foundation for our client's e-commerce platform, enabling effective data management, analysis, and decision-making. The comprehensive database structure will facilitate insights into user behavior, product popularity, and overall platform performance across multiple product categories.

Problems to solve:

* Data integrity issues
  + Entity integrity
  + Referential integrity
  + Domain integrity
* Security bridge
* Db falling apart
* Update anomalies

Jewellery event history table:

Entities

Attributes:

1. Categories: category\_id (FK), category\_code, category\_code\_id
2. purchase history from jewelry store
3. JewelryUser: user\_id, gender, row\_id (PK Surrogate)
4. JewelryEvent: event\_time, order\_id, product\_id (FK), row\_id (FK)
5. JewelryProduct: product\_id, row\_id (FK), quantity, price, colo, metal, gem
6. JewelryCategory: category\_id, category\_code\_id (FK), brand, row\_id (FK)
7. purchase history from electronics store
8. ElectronicUser: user\_id (PK), row\_id (FK), user\_session
9. ElectronicEvent: event\_time, event\_type, product\_id, row\_id (FK)
10. ElectronicProduct: product\_id, row\_id, price
11. ElectronicCategory: category\_id, category\_code\_id, brand, row\_id
12. Events History in Cosmetics Shop
13. CosmeticsUser: user\_id, row\_id, user\_session
14. CosmeticsEvent: event\_time, event\_type, product\_id, row\_id
15. CosmeticsProduct: product\_id, row\_id, price
16. CosmeticsCategory: category\_id, category\_code\_id, brand, row\_id
17. Events History in Multi Shop
18. MultiStoreUser: user\_id, row\_id, product\_id
19. MultiStoreEvent: event\_time, order\_id, product\_id, row\_id
20. MultiStoreProduct: product\_id, row\_id, price
21. MultiStoreCategory: category\_id, category\_code\_id, brand, price, row\_id

| **Activity** | **Action** | **Resources** | **Details** | **Results** |
| --- | --- | --- | --- | --- |
| **1. Define Requirements** | Identify Key Metrics | N/A | List metrics: total sales, revenue breakdown, etc. | Metrics: Total Sales, Revenue Breakdown, ... |
|  | Choose Visualizations | N/A | Determine visualizations for each metric. | Visualization for each metric: Bar, Pie, ... |
| **2. Set Up Power BI** | Install Power BI | Power BI Desktop | Download and install the Power BI application. | Power BI Desktop installed successfully. |
|  | Connect to Database | Power BI Desktop | Use data connection features to link to the DB. | Connected to the database successfully. |
| **3. Design Data Model** | Import Tables | Power BI Desktop | Import User, Event, Product, Category tables. | Tables imported: User, Event, Product, ... |
|  | Establish Relationships | Power BI Desktop | Define relationships based on common keys. | Relationships established successfully. |
| **4. Create Key Metrics** | Total Sales Measure | Power BI Desktop | Write DAX measure for total sales. | DAX measure created for Total Sales. |
|  | Revenue Breakdown Measure | Power BI Desktop | Create a measure for revenue breakdown. | DAX measure created for Revenue Breakdown. |
| **5. Design Visualizations** | Sales Overview Visuals | Power BI Desktop | Add card visual for total sales, line chart, etc. | Visualizations added for Sales Overview. |
|  | Revenue Breakdown Visuals | Power BI Desktop | Use pie chart/bar chart for revenue distribution. | Visualizations added for Revenue Breakdown. |
|  | Top Selling Products Table | Power BI Desktop | Create a table with relevant columns. | Table created for Top Selling Products. |
| **6. Customer Metrics** | Customer Acquisition Measure | Power BI Desktop | Write DAX for new vs. returning customers. | DAX measure created for Customer Acquisition. |
|  | Order Fulfillment Visuals | Power BI Desktop | Visualize order status and processing time. | Visualizations added for Order Fulfillment. |
| **7. Geographic Visualization** | Sales Map Visualization | Power BI Desktop | Use map visualization for geographic sales. | Map visualization added for Sales. |
| **8. Conversion Analysis** | Conversion Rate Funnel | Power BI Desktop | Implement funnel chart for conversion rates. | Funnel chart added for Conversion Rate. |
| **9. Time Comparison** | Monthly/Yearly Comparison Visuals | Power BI Desktop | Create bar chart for monthly/yearly data. | Bar chart added for Time Comparison. |
| **10. Refine and Test** | Review Design | Power BI Desktop | Ensure cohesive color scheme and branding. | Design reviewed and refined successfully. |
|  | Test Interactivity | Power BI Desktop | Verify user interaction for detailed insights. | Interactivity tested successfully. |
|  | Gather Feedback | Stakeholder Feedback | Share prototype for input and refinements. | Feedback gathered for further refinement. |

ments:

1. **Cart Abandonment Rate:**
   * **Calculation:** (����������������������/�����������������)(*NumberofAbandonedCarts*/*TotalCartsCreated*) \* 100
   * **Chart:** Funnel Chart or Donut Chart
   * **Explanation:** Measure the percentage of carts that are abandoned during the checkout process.
2. **Total Daily Users:**
   * **Calculation:** Count of unique users each day
   * **Chart:** Line Chart or Area Chart
   * **Explanation:** Track the total number of users visiting your platform daily.
3. **Daily Sales:**
   * **Calculation:** Sum of sales transactions each day
   * **Chart:** Line Chart or Bar Chart
   * **Explanation:** Monitor the total sales made on your platform on a daily basis.
4. **Total Orders:**
   * **Calculation:** Count of all completed orders
   * **Chart:** Bar Chart or Line Chart
   * **Explanation:** Track the overall number of orders placed on your platform.
5. **New Users:**
   * **Calculation:** Count of users who made their first purchase or visited for the first time
   * **Chart:** Bar Chart or Pie Chart
   * **Explanation:** Monitor the acquisition of new users to your platform.
6. **Average Order Value (AOV):**
   * **Calculation:** Total Revenue / Total Number of Orders
   * **Chart:** Line Chart or Bullet Chart
   * **Explanation:** Evaluate the average value of each order.
7. **Return Rate:**
   * **Calculation:** (Number of Returning Users / Total Number of Users) \* 100
   * **Chart:** Line Chart or Donut Chart
   * **Explanation:** Measure the percentage of users who return to your platform.
8. **Lifetime Value (LTV):**
   * **Calculation:** (Average Revenue per User) \* (Average Lifespan of a Customer)
   * **Chart:** Gauge Chart or Line Chart
   * **Explanation:** Estimate the total revenue a user is expected to generate during their entire relationship with your platform.
9. **Total Sessions:**
   * **Calculation:** Count of user sessions
   * **Chart:** Line Chart or Area Chart
   * **Explanation:** Track the total number of sessions (interactions) on your platform.
10. **Funnel Overview:**
    * **Calculation:** Map the conversion rates at each stage of the user journey
    * **Chart:** Funnel Chart
    * **Explanation:** Visualize the user journey from initial engagement to conversion.
11. **Average Revenue per User (ARPU):**
    * **Calculation:** Total Revenue / Total Number of Active Users
    * **Chart:** Line Chart or Bar Chart
    * **Explanation:** Evaluate the average revenue generated by each active user.
12. **Inventory Turnover Rate:**
    * **Calculation:** Cost of Goods Sold (COGS) / Average Inventory
    * **Chart:** Bar Chart or Radar Chart
    * **Explanation:** Measure how efficiently inventory is managed.
13. **Customer Acquisition Cost (CAC):**
    * **Calculation:** Total Marketing Costs / Number of New Customers Acquired
    * **Chart:** Line Chart or Bar Chart
    * **Explanation:** Assess the cost of acquiring each new customer.
14. **Website Traffic and Page Views:**
    * **Calculation:** Count of website visits and page views
    * **Chart:** Line Chart or Area Chart
    * **Explanation:** Monitor overall website traffic and engagement.
15. **Social Media Engagement:**
    * **Calculation:** Count of social media interactions (likes, shares, comments)
    * **Chart:** Stacked Bar Chart or Line Chart
    * **Explanation:** Track user engagement on different social media platforms.