



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Strategic Insights from Data-Driven Sales Patterns: Redesigning Supplier-Customer Relationships through Analytical Innovation

Dr. Raviteja Kocherla¹, S.Mohith Srinivas², J.vaishnavi³, V. Venkata Rohith Varma⁴,

A. Venkata Krishna Reddy⁵

Associate Professor, Malla Reddy University, Hyderabad , Telangana, India¹

Malla Reddy University Hyderabad , Telangana, India^{2,3,4,5}

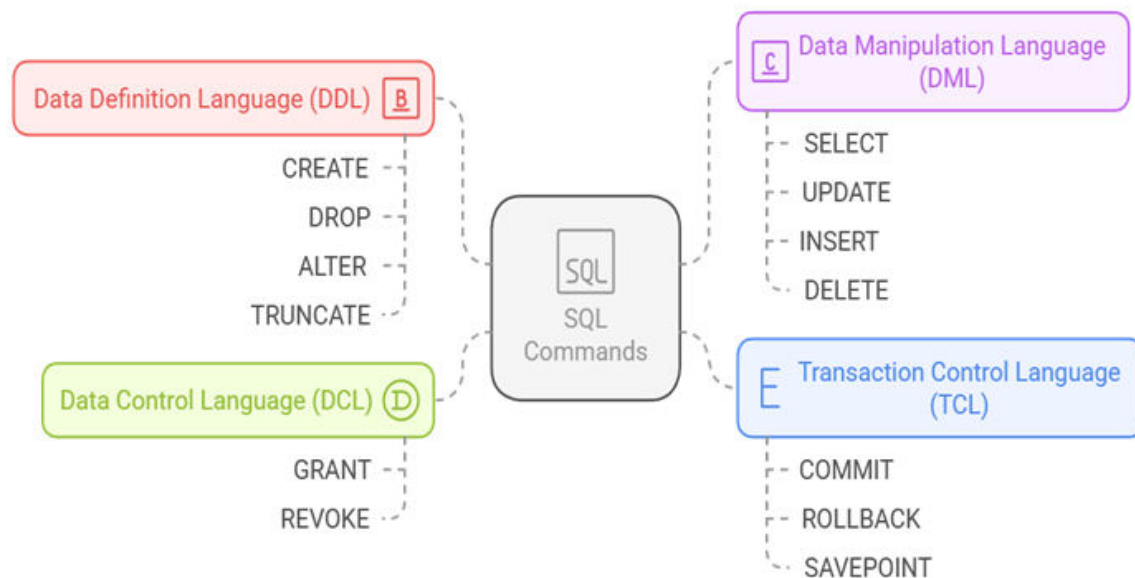
ABSTRACT: This project addresses the complexities of the e-commerce ecosystem, such as demand uncertainty, customer behavior, and inventory management, by integrating Advanced SQL techniques with Agent-Based Modeling (ABM). SQL is utilized to analyze historical sales data, focusing on aspects like customer buying patterns, product performance, and the impact of promotional strategies. Advanced techniques, including window functions, Common Table Expressions (CTEs), and data aggregation, enable businesses to extract actionable insights. These insights are then used to inform ABM, which simulates customer behavior and interactions within the sales environment. By modeling traits like price sensitivity and brand loyalty, ABM creates a virtual landscape for testing various sales strategies, marketing adjustments, and pricing scenarios. This powerful combination of SQL's data analysis capabilities and ABM's dynamic simulation framework helps businesses enhance decision-making, improve sales and inventory management, and predict outcomes more effectively. Together, these tools provide a robust structure for navigating the fast-paced, data-driven world of e-commerce, empowering businesses to stay competitive and responsive to changing market conditions.

I.INTRODUCTION

In the dynamic and rapidly growing scenario with e-commerce, companies face a lot of challenges. These include managing the demand from unexpected customers, optimizing stock levels, changing customer preferences and distributing effective marketing strategies. When intensifying the competitive environment, it is important for companies to use data -driven methods that enable them to make informed decisions and respond to market fluctuations with agility.

The project benefits from data collected from the Amazon sales sites through scraping of the web, using Python and AI Web Scraper. The data collected includes customer purchasing history, product pricing, discount trends and other sales -related matrix. The purpose of gathering this data is to gain deep insight into customer behavior, evaluate product trends and analyze the efficiency of sales preparation. The attention is to identify patterns in the display of sales and understand how customers interact with different campaigns.

To analyze this data, the project implemented SQL techniques, including window functions, common table expressions (CTE) and aggregation. These techniques enable the extraction of action -rich insights from data, and help predict sales trends and assess the success of different sales strategies. In addition, agent -based modeling (ABM) is used to simulate the interaction with customer behavior and products. By modeling different customer types and their reactions for prices and promotion, ABM provides real insight into how different sales strategies can work in real -world -scenarios. The main objective of this project is to demonstrate how to adapt to advanced SQL data analysis and ABM -Simulation, how can it improve the brand.



II. LITERATURE REVIEW

In the rapidly developed area for sales intelligence, the opportunity to extract meaningful insights from large amounts of data is important. This review focuses on existing research on advanced SQL techniques, which plays a basic role in analyzing sales data. There has been a great -scale discovery in recent literature during the scope of sales. SQL features such as Sum (), AVG (), Count () and window features such as Row_number (), Rank () and Team () are important tools for gaining insight from large datasets. These tasks are used to summarize sales performance, identify trends and make sophisticated analysis without the need for data discharge. Total labor companies help to measure the total revenues in tracking the product category, the average order price in areas and over time customer collection rates. This insight stations reported to determine inventory management, pricing strategies and marketing campaigns.

The use of window functions enables more fine analysis by allowing the calculation in a set of rows related to the current row, without collapse of rows in groups. This approach is beneficial to imagine driving totally, calculating the sliding average and ranking customers according to their total purchase value, and giving businesses a crop of the sales ecosystem. In addition, the paper sheds light on the use of regular table expressions (CTE) and subquests to simplify complex questions and to improve the readability of the query while maintaining performance. In addition, the paper Query discusses the importance of adaptation of data types to improve the efficiency of the execution. To ensure the use of appropriate data types (eg decimal for price calculations) stops problems.

III. METHODS

3.1. Data collection:

The data collection focuses on reconstructing extensive sales data from Amazon's online sales platform. Python was chosen as the primary programming language because of its versatility and extensive library support for scraping the grid. By using a combination of manual and AI-operated network techniques, we secured efficient and accurate data extraction. The online screaming process used libraries such as HTML content to automate interaction with dynamic pages and beautifules to pass the harness, which enabled us to capture main elements such as sales data, product listing and customer reviews. In addition, an AI networks improved the process by adapting to changes in the structure of the site, securing the extraction of complete data even in the dynamic environment. The most important features that were collected included product descriptions such as names, category and specifications -Sample with value, sales volume and customer response. Computer accuracy was a priority, which included random delays and user-agent replacement to deal with challenges such as CAPTCHA and Anti-scraping mechanisms. Despite the frequent

changes in the product list and prices on the platform, the approach allowed us to collect high quality data, which makes the basis for our analysis in this project.

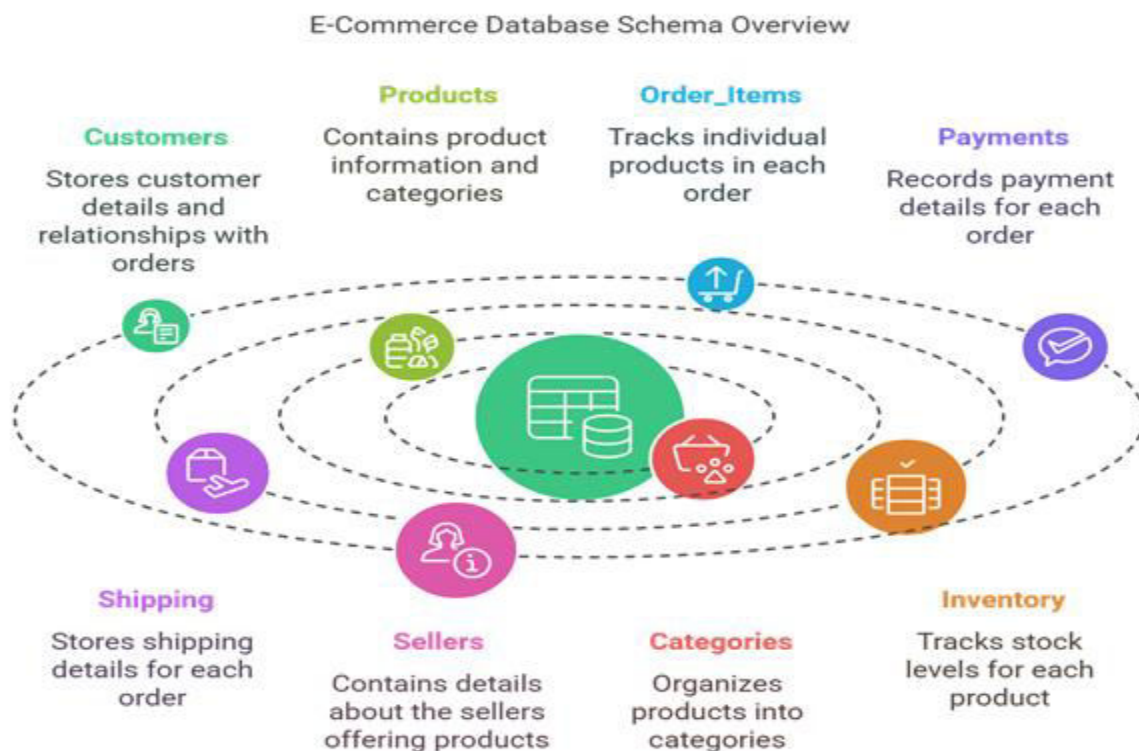
3.2. Data processing and Transformation:

When the raw data was collected from Amazon, the next phase included treatment and replaced it to change it. This phase was important, as it was ensured that the data was cleaned, structured and optimized for subsequent use in SQL-based questions and analysis. Data cleaning involves addressing problems such as duplicate entries, lack of values and inconsistent format. Powerful libraries of panda that panda were used to handle these works effectively. The duplicate record was removed, the missing values were either planted or left on the basis of their meaning, and constant formats were used in areas such as date, prices and product names. Subsequently, the data sale was converted to match the intelligence analysis needs. We broke large tables in small, related tables and normalized dataset, which follows a relationship with relationship design. This step was important to improve the efficiency of SQL questions and maintain the integrity of the data set. Instead of saving the product category information in the sales items, for example, we divorced them into a product category table and added them through foreign keys.

In addition, graded data, such as customer placement or product categories, were coded for compatibility with SQL function and analysis models. This organized structure enabled more effective questions and improved performance when collecting or filtering data. Overall, the treatment and change phase established a clean and adapted data set, and paved the way for advanced SQL-based analysis.

3.3. Database Schema Design:

A well -structured and efficient database scheme is fundamental to the organization, storage and recovery of large versions of Skima sales data. The database designs correspond to the main principles of database design, which focus on data integrity, scalability and efficiency to support advanced SQL analysis and agent-based modeling (ABM) used for sales intelligence information.



3.3.1. Database scheme design

The database was designed with a clear, generalized structure to support effective query and data analysis. Schima includes large tables such as: Customer: Save customer details (eg name, location, contact information).Product: Product information (eg product name, category, price).

Order: The transaction has data, connects customers with purchased products.

Order_items: A broad overview of each item in an order including volume, prices and discounts.

These tables are connected through foreign keys to ensure that the relationship (eg between orders and customers) is well defined and maintained. This generalized structure reduces data surplus and ensures that sales data can be easily collected and analyzed using advanced SQL questions.

3.3.2. Characterize

In order to maintain database efficiency and avoid inflated structures, the characteristic partition was carefully administered. For example, customer names were stored in logical regions (eg first and last names) without unnecessary division. This approach keeps the database lean and ensures particularly rapid query performance during data analysis with large -scale data.

3.3.3. Generalization and class hierarchy

Schima follows the principles of normalization, where the table was designed to reduce excess and maintain data integrity. Instead of storing the product categories repeatedly in each sales posts, for example, a separate product was formed a table with categories and was connected to the table of products through foreign keys. This design ensures that the sales system can effectively handle questions in multiple dimensions (eg product categories, regions and customer segments).In addition, a square hierarchy was used on the product categories, so that the products were grown logically (eg by brand, type or price limit). This structured approach simplifies the query when analyzing product performance in categories.

3.3.4. Referred to integrity and obstacles

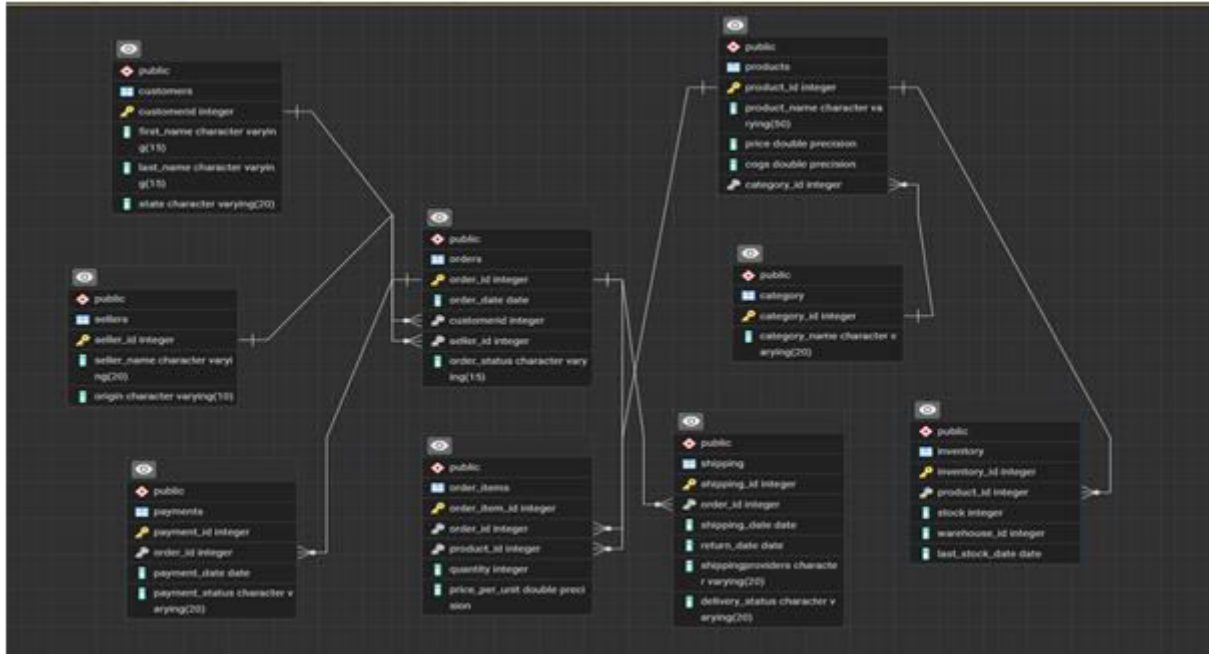
It was necessary to use contextual integrity to ensure the stability of the relationship between tables. Primary keys were installed for each table, such as Order_IDS in the order table and Customer_ID in the customer table, and ensured that each item can be identified specifically. Foreign major obstacles among tables ensured that, for example, an order could not exist without valid customer mail. This sale implements the database integrity, prevents orphans and ensures accurate analysis of sales trends.

3.3.5. Unique identifier and scalability

The design also used unique identifiers on important data points to allow scalable and reliable data management. For example, the identity columns were used for the order and automatic incarnation of primary keys by customers, ensuring that each entry has been traced unique. For scalability, unique identifiers were assessed globally for distributed systems, ensuring that data integration with external systems could basically be.

3.3.6. Customization of data type

Finally, the adjustment of the data type was carefully noticed. In order to ensure accuracy in financial calculations, prices and monetary values were stored using decimal (10.2) data type, while the customer's position field (eg active/passive) was stored as a Boolean for simplicity and performance. This adaptation increased the total speed and accuracy of the database, especially when handling complex questions and large datasets.



3.4. Advanced SQL functions

After cleaning and organizing the data, advanced SQL functions were used to highlight valuable sales performance and customer behavior insights. We used general functions such as sum (), AVG () and Count () to summarize the main matrix. For example, to calculate the total sales with the product category, we used the following Query:

For example:

```
SELECT category, SUM(total_amount) AS total_sales
FROM Products
JOIN Orders ON Products.product_id = Orders.product_id
GROUP BY category;
```

Query helped identify top -performing categories and caused the business to focus on products. Similarly, we calculated the average sequence value when using the area:

```
SELECT region, AVG(total_amount) AS avg_order_value
FROM Orders
JOIN Customers ON Orders.customer_id = Customers.customer_id
GROUP BY region;
```

Over time, it was used to monitor new customer collection:

```
SELECT MONTH(order_date) AS month,
COUNT(DISTINCT customer_id) AS new_customers
FROM Orders
GROUP BY month;
```

This insight helped to determine the inventory, prices and marketing strategies.

Then we used the window function to dive into the trends. Work like Row_number (), Rank () and Team () allowed us to analyze data in lines without a group. For example, the total sale of sales was obtained with the calculation:

```
SELECT order_date, SUM(total_amount)
OVER (ORDER BY order_date) AS running_total
FROM Orders;
```

This showed the trend of sales growth over time, and identified peaks during holiday sales. We also placed customers from their total purchases using Rank() feature:

```
SELECT customer_id, SUM(total_amount) AS total_spent,  
RANK() OVER (ORDER BY SUM(total_amount) DESC) AS customer_rank  
FROM Orders  
GROUP BY customer_id;
```

This helped to identify high -value customers, which enabled targeted marketing.

For more complex analysis, we used sliding average to reduce daily sales fluctuations and reveal long -term trends. A 7 -day window was implemented using a sliding average calculated:

```
SELECT order_date, AVG(total_amount)  
OVER (ORDER BY order_date ROWS BETWEEN  
6 PRECEDING AND CURRENT ROW) AS moving_avg  
FROM Orders;
```

We also used nested subcontracts and complex joints to highlight deep insights. To identify customers, who have not purchased the last six months, but have spent more than \$ 5,000 lately, we have used this combination of a subcontinent and clause:

```
SELECT customer_id, SUM(total_amount) AS total_spent  
FROM Orders  
WHERE customer_id NOT IN (  
    SELECT customer_id  
    FROM Orders  
    WHERE order_date > NOW() - INTERVAL '6 months'  
)  
GROUP BY customer_id  
HAVING SUM(total_amount) > 5000;
```

We also used regular table expressions (CTE) to make complex questions more readable. For example, the calculation of the customer's lifetime value (CLV) by collecting their total purchases over time:

```
WITH CustomerPurchases AS (  
    SELECT customer_id, SUM(total_amount) AS total_spent  
    FROM Orders  
    GROUP BY customer_id  
)  
SELECT customer_id, total_spent,  
CASE  
    WHEN total_spent > 10000 THEN 'High Value'  
    WHEN total_spent BETWEEN 5000 AND 10000 THEN 'Medium Value'  
    ELSE 'Low Value'  
END AS customer_value  
FROM CustomerPurchases;
```

Also challenge of finding the top 10 products with the highest declining income degree between 2022 and 2023 using advanced SQL techniques. Query will return product_ID, product name, category_name, 2022 revenue, 2023 revenue and decrease ratio (rounded).

```
WITH RevenueComparison AS (  
    -- Calculate total revenue per product for 2022  
    SELECT
```



```

p.product_id,
p.product_name,
c.category_name,
SUM(CASE WHEN YEAR(o.order_date) = 2022 THEN oi.quantity * oi.price ELSE 0 END) AS revenue_2022,
SUM(CASE WHEN YEAR(o.order_date) = 2023 THEN oi.quantity * oi.price ELSE 0 END) AS revenue_2023
FROM Products p
JOIN Order_Items oi ON p.product_id = oi.product_id
JOIN Orders o ON o.order_id = oi.order_id
JOIN Categories c ON p.category_id = c.category_id
GROUP BY p.product_id, p.product_name, c.category_name
)
SELECT
product_id,
product_name,
category_name,
ROUND(revenue_2022, 2) AS revenue_2022,
ROUND(revenue_2023, 2) AS revenue_2023,
ROUND((revenue_2023 - revenue_2022) / revenue_2022, 2) AS revenue_decrease_ratio
FROM RevenueComparison
-- Filter for products with decreasing revenue
WHERE revenue_2022 > 0 AND revenue_2023 < revenue_2022
-- Order by the highest decrease ratio
ORDER BY revenue_decrease_ratio ASC
-- Limit to top 10 products
LIMIT 10;

```

Identifying top 5 customers after orders in each state, you need:

Data group by state and customer: Calculate the number of orders and total sales for each customer. Ranking customers in each state depending on the number of orders. Limit the results to the 5 best customers for each state.

```

WITH CustomerOrders AS (
-- Calculate total orders and total sales for each customer in each state
SELECT
c.customer_id,
c.first_name || ' ' || c.last_name AS customer_name,
c.state,
COUNT(o.order_id) AS total_orders,
SUM(oi.quantity * oi.price) AS total_sales,
ROW_NUMBER() OVER (PARTITION BY c.state ORDER BY COUNT(o.order_id) DESC) AS order_rank
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Order_Items oi ON o.order_id = oi.order_id
GROUP BY c.customer_id, c.first_name, c.last_name, c.state
)
-- Select top 5 customers per state
SELECT
customer_id,
customer_name,
state,
total_orders,
ROUND(total_sales, 2) AS total_sales
FROM CustomerOrders
WHERE order_rank <= 5
ORDER BY state, order_rank;

```


IV. AGENT-BASED MODELING

The agent-based modeling (ABM) was integrated with SQL-powered analysis to simulate customer behavior and interaction with products, providing valuable insights to influence different factors. The ABM modeling complex is especially effective in the dynamic system, where different agents (eg customers) interact on the basis of different rules. By following these interactions, companies can better understand how prices, promotion and inventory customer decisions and the general sales result affect.

Agents were defined as individual customers, products and suppliers. Each agent was assigned unique characteristics and behaviors based on sales data drawn through SQL questions. For example:

4.1. Customer Agent:

Each customer agent was designed based on their historical procurement behavior, including the frequency of purchases, average order price and value recognition. Customer agents can be classified as price sensitive (more likely to respond to discounts) or branded (to buy specific products).

4.2. Product agents:

Product agents represented elements sold on the platform with properties such as price, category and stock level. These agents interacted with customer agents during simulation to buy models.

4.3. Seller agent:

Each seller agent had properties such as seller assessment and product range. These agents affected the customer's purchase decisions based on the availability of the product and the supplier's reputation.

The main objective of ABM (agent -based modeling) was to simulate customer behavior based on different sales scenarios, which provide a deep understanding of how these factors affect the general sales performance. These simulations were directly associated with the insight obtained from the historical sales data collected using SQL techniques

4.4. Price adjustment:

Analysis of one of the main simulations involves how customers reacted to changes in product pricing. This allowed us to test the scenarios where exemption or promotion on specific products were implemented. Customers model based on their sensitivity to their historical procurement pattern, especially their sensitivity to exemption. For example, some agents, identified as price sensitive through SQL query, had more likely to act when the price of a product fell, leading to an increase in sales. This helped the business determine the optimal pricing strategies for different customer segments.

4.5. Lack of inventory:

Another important scenario involves the test of how customers reacted to product shortages or stockout. By mimicking storage problems, we can see if customers chose to switch to the products or delay the purchase. This provided valuable insight into which products need strong inventory management and how share problems affected customer loyalty. Computer-controlled rules for this simulation were based on product preferences and procurement behavior from SQL analysis.

4.6. Marketing campaign:

The ABM model was also used to evaluate the efficiency of targeted marketing campaigns. SQL questions were used to identify high-value customers (who produced the most revenue or continuous shopping). These customers were simulated in different campaign scenarios, so we could see how individual suggestions, such as exclusive discounts, affect the procurement behavior. The results have shown that marketing measures led to an increase in high sales from commitment and high -value customer segments, and validated the importance of customer division in marketing strategies.

Agent interactions in the model were corrected according to rules made by real historical sales data. For example, it was more likely that customers who were usually purchased during the promotional period respond to future

promotion. This rule -based simulation allowed the project to copy the real buying behavior and test how different sales factors can affect the results in different commercial scenarios.

5. Sales Insights:

The combination of Advanced SQL techniques and Agent-Based Modeling (ABM) are provided rich insight into the sales ecosystem of the project, which enabled computer -controlled decisions in many commercial areas, including sales forecasts, inventory management and target marketing in many commercial sectors, including sales forecasts, inventory management and target marketing.

5.1. Sales forecasts and trend analysis:

By taking advantage of the SQL agiget features (for example, sum () and AVG () and window functions (Row_number (), Rank () and Team ()), the project on sales, revenue growth and customer purchase patterns generated. For example:

SQL-based driving Total calculation and relocation of the average helped to identify seasonal sales lodges and long-term development trends. This enabled the business to meet the needs of inventory and staffing. SQL Query identified top -bound products and under -portrated categories, where the marketing efforts should be focused or which products require storage optimization.

5.2. Customer Division and Behavior:

Through the SQL analysis and integration of ABM simulation, the project provided a detailed approach to customer behavior, and shared customers based on purchase history, preferences and reactions to promotion. Some great insights include:

The highly valued customers, identified using SQL's rankings, were more responsible for individual marketing campaigns, which led to long -term sales. Price-sensitive customers, who seen in ABM simulation, were quick to respond to the discount, which often led to short-term spikes in sales. However, these world -sensitive customers were less loyal, as the products were more likely to switch at the end.

5.3. Inventory Management:

The combination of SQL and ABM also provided strong insight into inventory management. SQL questions tracked product performance and stock levels, while ABM simulation models Customer behavior in response to stock shortages and stockout. These simulations revealed how customers reacted to inaccessible products:

When the preferred products were sold out, the replacement was purchased was common, it helps to identify which alternative products should be preferred in storage. Delayed purchases for specific high -raising products reported better restoration programs and inventory.

5.4. Price and reproduction adjustment:

By simulating customer responses for different prices and promotional strategies, ABM provided valuable insight into price optimization. Combined with SQL analysis of previous promotional data, the project helped determine the level of optimal discounts for different products:

Medium discounts continued to increase sales for high excellent products, while deep discounts on less popular goods did not provide long -term sales benefits. Income tracking of SQL in different pricing strategies showed which campaign provided the most benefits and what products should be painted in future promotion.

5.5. Targeted Marketing Insight:

SQL Query of the project divided customers into different groups based on buying behavior, such as buyers with high existence, relevant shop owners and price -sensitive customers. ABM simulation then tested the target marketing strategies on these sections:

The campaign campaign for high-value customers increased high storage rate and total sales, as identified by the SQL rankings for top customers. The e -mail and discount campaign focusing on customers with long procurement history led to a significant increase in reimbursing the sales of the customer, optimizing the marketing budget and improving customer loyalty.

VI. CONCLUSION

Power to combine advanced SQL techniques and agent-based modeling (ABM) to adapt sales intelligence in the e-commerce environment. By analyzing historical sales data through SQL Query and imitating customer behavior with ABM, the project provided action -rich insights, which increased the business decision in several dimensions, including sales forecasts, inventory management and target marketing strategies. The combination of SQL analysis and ABM - Simulation converted the company's approach to sales intelligence, so that more accurate and informed decisions are made. The insight received from the project helped to streamline operations, promote the results of sales and adapt marketing efforts, which led to high profitability and customer satisfaction.

REFERENCES

1. <https://ieeexplore.ieee.org/document/7030467/>
2. <https://www.dasca.org/world-of-data-science/article/advanced-sql-techniques-to-transform-data-analysis>
3. <https://ieeexplore.ieee.org/document/10251253/>
4. <https://ieeexplore.ieee.org/iel8/6287639/10820123/10849522.pdf>
5. <https://ieeexplore.ieee.org/iel8/6287639/7419931/7514992-aam.pdf>
6. file:///D:/Downloads/Volume_4_issue_3_1_AJBAS.pdf
7. <https://ieeexplore.ieee.org/document/10251253/>
8. <https://ieeexplore.ieee.org/iel8/6287639/10820123/10849522.pdf>
9. <https://ieeexplore.ieee.org/iel8/6287639/7419931/7514992-aam.pdf>
10. <https://smythos.com/ai-industry-solutions/marketing-automation/agent-based-modeling-in-marketing/>
11. <https://ieeexplore.ieee.org/document/7822080/>
12. <https://faculty.sites.iastate.edu/tesfatsi/archive/tesfatsi/ACERetailCustomerModeling.pdf>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details