



Blinkit Sales Analysis

ABSTRACT

This project analyzes Blinkit's sales data using SQL to uncover key business insights. It examines sales trends, customer behavior, and revenue performance to optimize decision-making. By identifying top-selling products, peak sales periods, and high-value customers, the analysis helps improve marketing strategies, inventory management, and overall business growth. The project showcases data-driven insights that can enhance operational efficiency and revenue generation.

Project Title: Blinkit Sales Analysis

Project Description:

This project focuses on analyzing **Blinkit's** sales data using **SQL** to derive actionable business insights. The goal is to explore **sales trends, customer behavior, and revenue performance** to help optimize decision-making.

By leveraging SQL queries, I extracted and analyzed transactional data to identify **top-selling products, peak sales periods, and high-value customers**. The insights derived can be used to **enhance marketing strategies, improve inventory management, and drive business growth**.

Key Highlights:

- Sales Performance:** Analyzing revenue trends over time.
- Top-Selling Products:** Identifying the most popular items.
- Customer Purchase Behavior:** Understanding buying patterns and high-value customers.

Technologies Used:

- **SQL** for data extraction and transformation
- **PostgreSQL/MySQL** for database management

Outcome:

The project provides **data-driven recommendations** to optimize sales strategies and improve Blinkit's overall business performance.

```
-- What is the total revenue generated from all sales?
SELECT
    SUM(a.price * b.quantity) AS total_revenue
FROM
    blinkit_products a
        JOIN
    blinkit_order_items b ON a.product_id = b.product_id;
```

total_revenue
1043018.22

```
-- What is the total revenue for each day?
SELECT
    Date(blinkit_orders.order_date) AS order_day,
    SUM(blinkit_products.price * blinkit_order_items.quantity) AS total_revenue
FROM
    blinkit_orders
        JOIN
    blinkit_order_items ON blinkit_order_items.order_id = blinkit_orders.order_id
        JOIN
    blinkit_products ON blinkit_products.product_id = blinkit_order_items.product_id
GROUP BY order_day
LIMIT 10;
```

order_day	total_revenue
2023-03-16	4445.09
2023-03-17	530.85
2023-03-18	2893.8
2023-03-19	166.36
2023-03-21	2139.5
2023-03-22	491.39
2023-03-23	2003.6
2023-03-24	1043.67
2023-03-25	2524.94
2023-03-26	1897.29

```
-- Which are the top 5 best-selling products by quantity sold?
SELECT
    blinkit_products.product_name,
    SUM(blinkit_order_items.quantity) AS quantity_sold
FROM
    blinkit_products
    JOIN
        blinkit_order_items ON blinkit_products.product_id = blinkit_order_items.product_id
GROUP BY blinkit_products.product_name
ORDER BY quantity_sold DESC
LIMIT 5;
```

product_name	quantity_sold
Vitamins	96
Lotion	88
Pet Treats	86
Baby Wipes	85
Cough Syrup	75

```
-- Which product categories contribute the most revenue?
SELECT
    blinkit_products.category,
    SUM(blinkit_order_items.quantity * blinkit_order_items.unit_price) AS revenue
FROM
    blinkit_products
    JOIN
        blinkit_order_items ON blinkit_products.product_id = blinkit_order_items.product_id
GROUP BY blinkit_products.category
ORDER BY revenue DESC;
```

category	revenue
Dairy & Breakfast	146792.34
Pharmacy	145334
Fruits & Vegetables	130902.58
Pet Care	96031.9
Cold Drinks & Juices	85835.21
Snacks & Munchies	84487.7
Baby Care	80018.65
Household Care	76378.85
Grocery & Staples	71961.55
Personal Care	64781.02
Instant & Frozen Food	60494.42

```
-- What are the peak sales hours based on order timestamps?  
SELECT  
    HOUR(order_date), COUNT(order_id)  
FROM  
    blinkit_orders  
GROUP BY HOUR(order_date);
```

HOUR(order_date)	COUNT(order_id)
0	53
1	52
2	35
3	37
4	53
5	37
6	44
7	48
8	49
9	40
10	44
11	40
12	37
13	43
14	44
15	32
16	52
17	44
18	45
19	49
20	45
21	47
22	43
23	48

```
-- What is the average delivery time delay?
```

```
SELECT  
    ROUND(AVG(TIMESTAMPDIFF(MINUTE,  
        promised_delivery_time,  
        actual_delivery_time)),  
    2) AS avg_delivery_delay  
FROM  
    blinkit_orders;
```

avg_delivery_delay
4.32

```
-- Which top 5 cities or areas generate the highest revenue?
SELECT
    blinkit_customer.area,
    SUM(blinkit_order_items.quantity * blinkit_order_items.unit_price) AS revenue
FROM
    blinkit_customer
    JOIN
    blinkit_orders ON blinkit_customer.customer_id = blinkit_orders.customer_id
    JOIN
    blinkit_order_items ON blinkit_order_items.order_id = blinkit_orders.order_id
GROUP BY blinkit_customer.area
ORDER BY revenue DESC
LIMIT 5;
```

area	revenue
Orai	14827.94
Ghaziabad	11899.83
Nandyal	11060.23
Haldia	10882.8
Bareilly	10286.15

```
-- How many products were sold at a discount
SELECT
    sum(blinkit_order_items.quantity) AS total_product_sold,
    ROUND(((blinkit_products.mrp - blinkit_order_items.unit_price) / (blinkit_products.mrp)) * 100,
    2) AS discount_percentage
FROM
    blinkit_products
    JOIN
    blinkit_order_items ON blinkit_order_items.product_id = blinkit_products.product_id;
```

total_product_sold	discount_percentage
2117	35.00

```
-- How many unique customers have placed orders?  
SELECT  
    COUNT(DISTINCT (customer_id)) AS unique_customer  
FROM  
    blinkit_customer  
WHERE  
>     customer_id IN (SELECT customer_id FROM blinkit_orders  
>         WHERE order_id);
```

unique_customer
885

```
-- How many customers have placed more than one order  
SELECT COUNT(customer_id) as More_than_one_purchase  
FROM   blinkit_customer  
WHERE customer_id IN (SELECT customer_id FROM  
                      (SELECT customer_id, COUNT(order_id) AS order_count  
                       FROM blinkit_orders  
                       GROUP BY customer_id) AS order_c  
                      WHERE order_count > 1);
```

More_than_one_purchase
159

```
-- What is the average order value (AVO) ?  
SELECT  
    ROUND(SUM(a.quantity * b.price) / COUNT(a.order_id),2)  
    AS AVO  
FROM  
    blinkit_order_items a  
    JOIN  
    blinkit_products b ON a.product_id = b.product_id;
```

AVO
983.05

```
-- Who are the top 10 high-value customers based on total spending ?  
SELECT  
    a.customer_name, SUM(b.order_total) AS spend  
FROM  
    blinkit_customer a  
    JOIN  
    blinkit_orders b ON a.customer_id = b.customer_id  
GROUP BY customer_name  
ORDER BY spend DESC  
LIMIT 10;
```

customer_name	spend
Charvi Tripathi	12764.73
Chasmum Dhaliwal	10203.09
Samarth Wagle	10156.18
Yashvi Dua	9931.9
Xalak Goyal	9371.26
William Guha	8772.28
Charita Comar	8757.15
Eta Srinivasan	8397.66
Xalak Sangha	8220.41
Vasana Chandran	7962.01

```
-- What are the most common customer segments
SELECT DISTINCT
    customer_segment AS segment,
    COUNT(customer_id) AS comman_customer
FROM
    blinkit_customer
GROUP BY segment
ORDER BY comman_customer DESC;
```

segment	comman_customer
Regular	639
Premium	633
New	628
Inactive	600

```
-- What is the most common payment method among high-value customers?
SELECT DISTINCT
    payment_method AS payment,
    COUNT(customer_id) AS customer_count
FROM
    blinkit_orders
GROUP BY payment
ORDER BY customer_count DESC;
```

payment	customer_count
Cash	280
Card	268
UPI	266
Wallet	247

Conclusion:

The **Blinkit Sales Analysis** project successfully demonstrates the power of **data-driven decision-making** in optimizing business performance. By leveraging **SQL queries**, we identified key trends such as **top-selling products, peak sales periods, and customer purchasing behavior**. These insights can help businesses improve **inventory management, marketing strategies, and customer engagement**.

Moving forward, integrating **predictive analytics and visualization tools** can further enhance the depth of insights, allowing for more **proactive business strategies**. This analysis underscores the importance of **data analytics in driving growth and operational efficiency**.