**Case Study: Exploring Customer Behavior**

**1. Introduction**

This case study aims to analyze customer behavior patterns by leveraging data from an online food delivery platform. By exploring various queries, the study identifies trends, product popularity, revenue streams, and customer preferences. The dataset contains tables representing products, sales, users, gold memberships, and user names, allowing us to derive insights into customer actions and their impact on revenue.

**2. ERD Diagram**

The following Entity-Relationship Diagram (ERD) represents the relationships between the database tables:  
  
A screenshot of a computer

Description automatically generated

**3. Data Analysis Questions and Answers**

**Q1. What is the total sales revenue generated by each product?**

select p.product\_id, p.product\_name, sum(price) as sales\_revenue from product p

join sales s on p.product\_id =s.product\_id

group by p.product\_name, p.product\_id;

A screenshot of a computer

Description automatically generated

**Observation:** Dal Makani generates the highest sales revenue compared to others.

**Q2. Which 3 product has the highest sales revenue?**

select top 3 p.product\_id, p.product\_name, sum(price) as sales\_revenue from product p

join sales s on p.product\_id =s.product\_id

group by p.product\_name, p.product\_id

order by sales\_revenue desc;

A screenshot of a computer

Description automatically generated

**Observation:** Butter Chicken, Shahi Paneer, and Mutton Biriyani are the top-performing products.

**Q3. How many users have signed up for the service and has taken the gold membership?**

select (select count(\*) from users) as no\_of\_users\_signed\_up, count(\*) as no\_of\_users\_taken\_gold\_membership from goldusers\_signup;

A screenshot of a computer

Description automatically generated

**Observation:** Out of 10 users, 8 have taken the gold membership.

**Q4. What is the revenue generated from gold users?**

select g.userid ,un.Names ,SUM(p.price) as total\_revenue\_by\_gold\_users from product p

join sales s on p.product\_id =s.product\_id

join goldusers\_signup g on s.userid = g.userid

join user\_name un on g.userid = un.userid

group by g.userid, un.Names;

A screenshot of a computer

Description automatically generated

**Observation:** Gold users have contributed significantly to the overall revenue.

**Q5. What is the total revenue generated from gold users?**

select SUM(p.price) as total\_revenue\_by\_gold\_users from product p

join sales s on p.product\_id =s.product\_id

join goldusers\_signup g on s.userid = g.userid;

A screenshot of a computer

Description automatically generated

**Observation:** The total revenue from gold users is 5560.

**Q6. Which users has been a gold user for the How much of time?**

select g.userid,un.Names, g.gold\_signup\_date, CAST(GETDATE() AS DATE) as today , DATEDIFF(DAY,g.gold\_signup\_date,GETDATE()) as period\_of\_membership\_in\_days

from goldusers\_signup g

join user\_name un on g.userid = un.userid

order by period\_of\_membership\_in\_days desc;

A screenshot of a computer

Description automatically generated

**Observation:** Sara has been a gold member for the longest period.

**Q7. What is the most popular product among gold users?**

select p.product\_name, count(\*) as orderd\_times from product p

join sales s on p.product\_id =s.product\_id

join goldusers\_signup g on s.userid = g.userid

group by p.product\_name

order by orderd\_times desc;

A screenshot of a computer

Description automatically generated

**Observation:** Shahi Paneer is the most popular product among gold users.

**Q8. What is the total sales revenue generated in each year?**

select YEAR(created\_date) as year, sum(p.price) as sales\_revenue from sales s

join product p on p.product\_id = s.product\_id

group by YEAR(created\_date);

A screenshot of a computer

Description automatically generated

**Observation:** Sales revenue was highest in 2017

**Q9. How has the sales revenue changed over the years?**

with revenue\_change as(

select YEAR(created\_date) as sales\_year, sum(p.price) as sales\_revenue from sales s

join product p on p.product\_id = s.product\_id

group by YEAR(created\_date)

)

select sales\_year, sales\_revenue ,

lag(sales\_revenue) over (order by sales\_year) as Previous\_year\_revenue,

sales\_revenue - lag(sales\_revenue) over (order by sales\_year) as revenue\_change,

case

when lag(sales\_revenue) over (order by sales\_year) is null then null

else

((sales\_revenue - lag(sales\_revenue) over (order by sales\_year)) \* 100) /lag(sales\_revenue) over (order by sales\_year)

end as percentage\_change

from revenue\_change;

A screenshot of a report

Description automatically generated

**Observation:** Sales increased by 232% in 2016 but dropped by 46% in 2019.

**Q10. What is the average Gold-signup compare to just sign up for the users?**

with user\_comparison as (

select

(select

count(\*) from users) as no\_of\_users\_signed\_up,

count(\*) as no\_of\_users\_taken\_gold\_membership from goldusers\_signup

)

select no\_of\_users\_signed\_up, no\_of\_users\_taken\_gold\_membership,

round((cast(no\_of\_users\_taken\_gold\_membership as float) \* 100.0) / no\_of\_users\_signed\_up,1) as gold\_sign\_up\_percentage

from user\_comparison;

A screenshot of a computer

Description automatically generated

**Observation:** 80% of total users opted for the gold membership.

**Q11. How many gold members users have order how many numbers of time ?**

select g.userid, un.Names, count(\*) as Total\_orders from sales s

join goldusers\_signup g on g.userid = s.userid

join user\_name un on un.userid = g.userid

group by g.userid,un.Names

order by Total\_orders desc;

A screenshot of a computer

Description automatically generated  
**Observation:** Anshul and Shreya placed the highest number of orders among gold members.

**Q12. What is the total amount each customer spend on Online Food Delivery?**

select u.userid, un.Names, sum(p.price) as Total\_spends from sales s

join users u on u.userid = s.userid

join user\_name un on un.userid = u.userid

join product p on p.product\_id = s.product\_id

group by u.userid, un.Names;

A screenshot of a computer

Description automatically generated

**Observation:** Shreya spent the most on online food delivery.

**Q13. What is the frequency of customer visits to the online platform?**

select u.userid, un.Names, count(\*) as Total\_visits from sales s

join users u on u.userid = s.userid

join user\_name un on un.userid = u.userid

group by u.userid, un.Names;

A screenshot of a computer

Description automatically generated

**Observation:** Anshul visited the platform more frequently.

**Q14. What was the first order purchase by each customer ?**

with purchase\_cte as (

select u.userid, un.Names, s.created\_date,

rank() over (partition by u.userid order by s.created\_date) as purchased\_rank

from sales s

join users u on u.userid = s.userid

join user\_name un on un.userid = u.userid

)

select userid,names,created\_date from purchase\_cte where purchased\_rank = 1 order by created\_date;

A screenshot of a computer

Description automatically generated

**Observation:** Sahil placed the first order on 2014-04-02.

**Q15. What is the most purchase item on the menu and how many times was it purchased by all customers?**

select top 1 p.product\_name, count(s.product\_id) as total\_purchases from sales s

join product p on p.product\_id = s.product\_id

group by p.product\_name

order by total\_purchases desc;

A screenshot of a computer

Description automatically generated

**Observation:** Shahi Paneer is the most purchased item.

**Q16. Which item was most popular for each customer ?**

with popular\_cte as (

select un.Names,p.product\_name , count(p.product\_id) as total\_purchases ,

rank() over (partition by un.names order by count(p.product\_id) desc) as purchased\_rank --can use row\_number() as well but that will ignore products with same number of purchases

from sales s

join users u on u.userid = s.userid

join user\_name un on un.userid = u.userid

join product p on p.product\_id = s.product\_id

group by un.names,p.product\_name

)

select names,product\_name from popular\_cte where purchased\_rank = 1;

A screenshot of a menu

Description automatically generated

**Q17. Which item was purchase first by the customer after they become a member ?**

with purchase\_cte as (

select un.Names, s.created\_date as purchase\_date, p.product\_name ,

rank() over (partition by u.userid order by s.created\_date) as purchased\_rank

from sales s

join goldusers\_signup u on u.userid = s.userid

join user\_name un on un.userid = u.userid

join product p on p.product\_id = s.product\_id

where s.created\_date > u.gold\_signup\_date

)

select Names,product\_name as first\_purchase from purchase\_cte where purchased\_rank = 1;

A screenshot of a computer

Description automatically generated

**Q18.Which item was purchase before the customer become a member ?**

with purchase\_cte as (

select u.userid ,un.Names, s.created\_date as purchase\_date, p.product\_name ,u.gold\_signup\_date ,

rank() over (partition by u.userid order by s.created\_date desc) as purchased\_rank

from sales s

join goldusers\_signup u on u.userid = s.userid

join user\_name un on un.userid = u.userid

join product p on p.product\_id = s.product\_id

where s.created\_date < u.gold\_signup\_date

)

select Names,product\_name as first\_purchase from purchase\_cte where purchased\_rank = 1;

A screenshot of a computer

Description automatically generated

**Q19. What is the total orders and amount spent for each member before they become a member ?**

select u.userid ,un.Names, count(s.product\_id) as times\_purchased ,sum(p.price) as total\_amount

from sales s

join goldusers\_signup u on u.userid = s.userid

join user\_name un on un.userid = u.userid

join product p on p.product\_id = s.product\_id

where s.created\_date < u.gold\_signup\_date

group by u.userid ,un.Names;

A screenshot of a computer

Description automatically generated

**Observation:** Members spent a significant amount and placed several orders before becoming gold members.

**Q20. Rank all the transactions for each member whenever they are a XYZ gold member for every non gold member Transaction marks as na ?**

select un.Names, p.product\_name, p.price,

case

when g.userid is null then 'NA'

else cast(dense\_rank() over (partition by un.names order by p.price desc) as varchar(2))

end as Purchase\_rank

from sales s

join users u on u.userid = s.userid

join product p on p.product\_id = s.product\_id

left join goldusers\_signup g on u.userid = g.userid

join user\_name un on u.userid = un.userid;

A table with numbers and letters

Description automatically generated

**Observation:** Transactions made during gold membership are ranked, while non-gold transactions are marked as "NA."