zomato

is it zoMaito, zoMaato or jomato?

Data Analysis

SQL Case Study

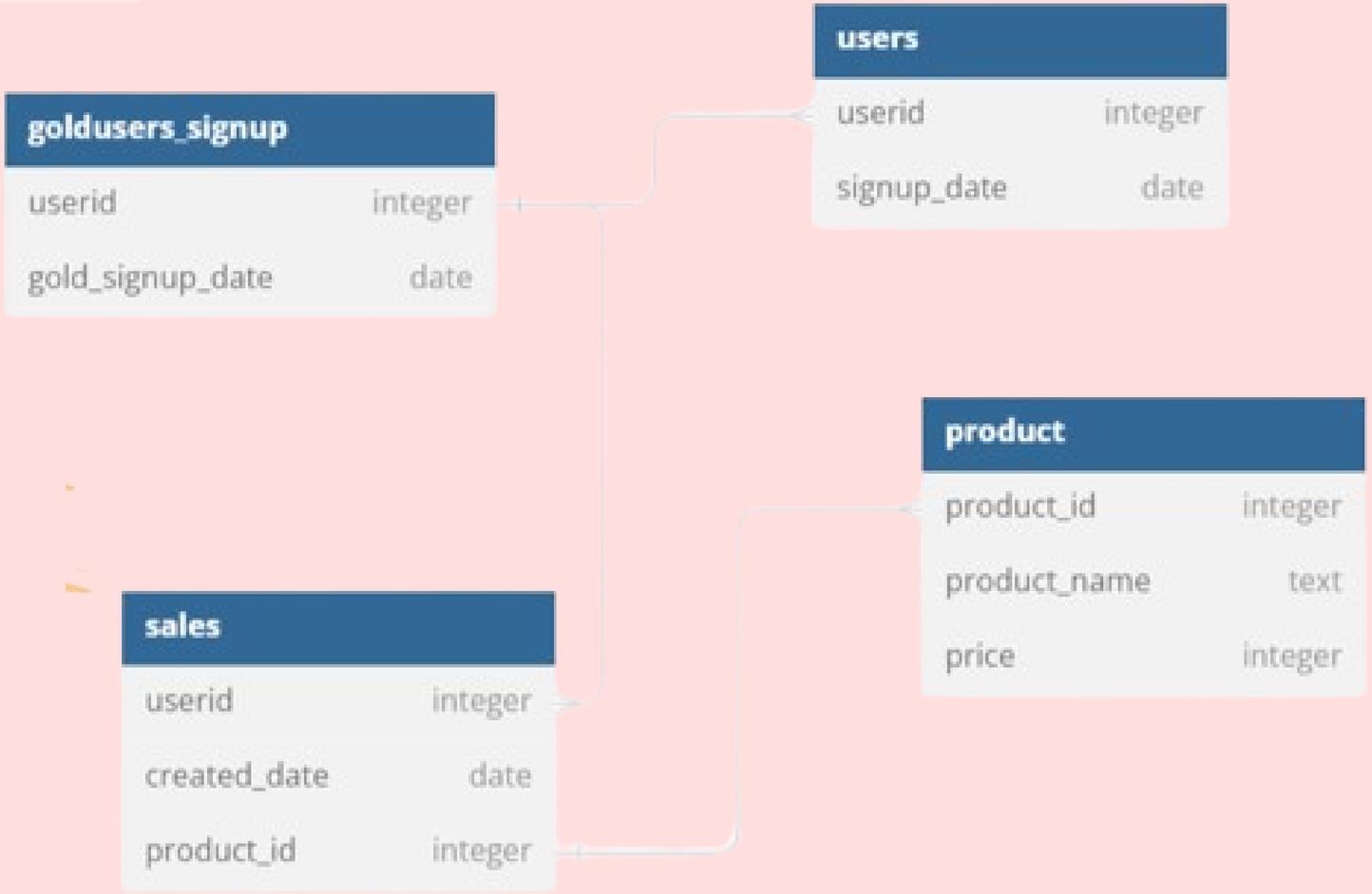
Project By: Arun Kumar

Overview: Unveiling Insights from Zomato with SQL

In this project, I delved into the fascinating world of Zomato, a leading online food delivery and restaurant discovery platform. Utilizing the power of SQL, I explored and analyzed Zomato's data, aiming to:

- **Uncover hidden patterns and trends:** Discover key insights about customer behavior, restaurant performance, and overall platform usage.
- **Enhance decision-making:** Provide valuable information to stakeholders to help them make informed decisions about marketing strategies, resource allocation, and service improvements.
- **Sharpen my SQL skills:** Gain practical experience with SQL queries, data manipulation techniques, and data analysis approaches.

Architecture



Here are the topics analyzed in the case study:

- what is the total amount each customer spent on Zomato ?
- how many days has each customer visited zomato ?
- what was the first product purchased by each customer ?
- what is the most purchased item on the menu and how many times was it purchased by all customers?
- what item was the most popular for each customer?
- which item was purchased first by the customer after they became a member?
- which item was purchased just before the customer became a member?
- what is the total order and amount spent for each member before they became a member?
- rank all the transaction of the customers
- rank all the transaction for each member whenever they are a Zomato gold member for every non gold member transaction mark as ‘NA‘

what is the total amount each customer spent on zomato ?

```sql
SELECT
  a.userid,
  SUM(b.price) AS Total_amount_spent
FROM sales a
INNER JOIN product b ON a.product_id = b.product_id
GROUP BY a.userid;
```

userid	Total_amount_spend
1	5230
2	2510
3	4570

How many days has each customer visited zomato ?

```sql
SELECT userid,
COUNT(DISTINCT created_date) as distinct_days
from sales
GROUP by userid
```

userid	distinct_days
1	7
2	4
3	5

what was the first product purchased by each customer ?

```sql
SELECT *
FROM (
  SELECT *,
         RANK() OVER (PARTITION BY userid ORDER BY created_date) AS rank
  FROM sales
) ranked_sales
WHERE rank = 1;
```

userid	created_date	product_id	rank
1	03-11-2016	1	1
2	07-20-2020	3	1
3	11-10-2016	1	1

what is the most purchased item on the menu and how many times was it purchased by all customers?

```sql
SELECT userid, count(product_id) as cnt
from sales
WHERE product_id = ( SELECT product_id
                     from sales
                     GROUP by product_id
                     ORDER by COUNT(product_id) DESC)
GROUP by userid;
```

userid	cnt
1	3
2	1
3	3

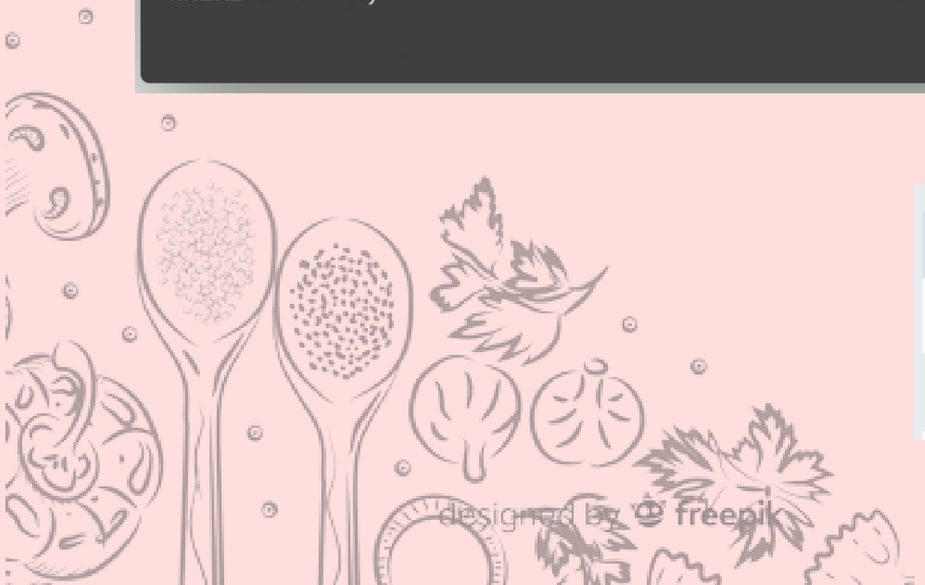
what item was the most popular fro each customer?

```sql
SELECT *
FROM (
  SELECT s.*,
         RANK() OVER (PARTITION BY s.userid ORDER BY s.cnt DESC) AS rnk
  FROM (
    SELECT userid, product_id, COUNT(*) AS cnt
    FROM sales
    GROUP BY userid, product_id
  ) AS s
) ranked_sales
WHERE rnk = 1;
```

userid	product_id	cnt	rnk
1	2	3	1
2	3	2	1
3	2	3	1

which item was purchased first by the customer after they became a member?

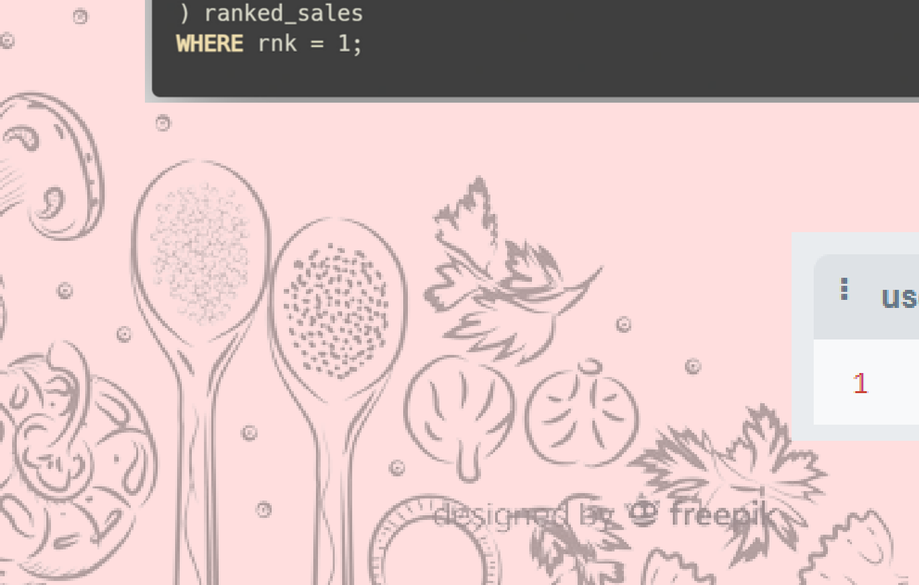
```sql
SELECT *
FROM (
  SELECT c.*,
         RANK() OVER (PARTITION BY c.userid ORDER BY c.created_date DESC) AS rnk
  FROM (
    SELECT a.userid, a.created_date, a.product_id, b.gold_signup_date
    FROM sales a
    INNER JOIN goldusers_signup b ON a.userid = b.userid
                                 AND a.created_date >= b.gold_signup_date
  ) AS c
) ranked_sales
WHERE rnk = 1;
```



userid	created_date	product_id	gold_signup_date	rnk
1	10-23-2019	2	09-22-2017	1
3	11-10-2016	1	04-21-2017	1

which item was purchased just before the customer became a member?

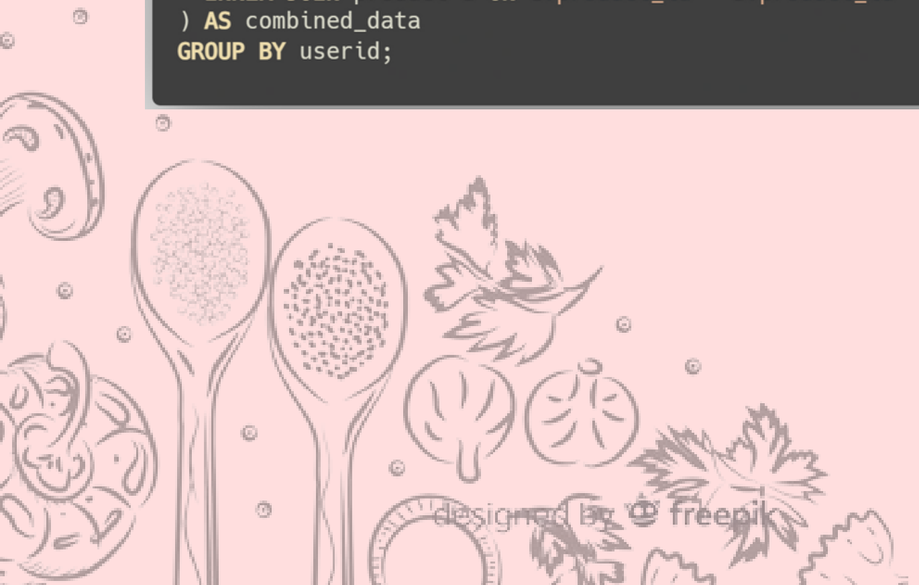
```sql
SELECT *
FROM (
 SELECT c.*,
        RANK( ) OVER (PARTITION BY c.userid ORDER BY c.created_date DESC) AS rnk
 FROM (
   SELECT a.userid,
          a.created_date,
          a.product_id,
          b.gold_signup_date
   FROM sales a
   INNER JOIN goldusers_signup b ON a.userid = b.userid
                                AND a.created_date <= b.gold_signup_date
 ) AS c
) ranked_sales
WHERE rnk = 1;
```



userid	created_date	product_id	gold_signup_date	rnk
1	05-20-2016	3	09-22-2017	1

what is the total order and amount spent for each member before they became a member?

```sql
SELECT
  userid,
  COUNT(created_date) AS order_purchased,
  SUM(price) AS total_spent
FROM (
  SELECT c.*, d.price
  FROM (
    SELECT a.userid, a.created_date, a.product_id, b.gold_signup_date
    FROM sales a
    INNER JOIN goldusers_signup b ON a.userid = b.userid
                                 AND a.created_date <= b.gold_signup_date
  ) AS c
  INNER JOIN product d ON c.product_id = d.product_id
) AS combined_data
GROUP BY userid;
```



userid	order_purchased	total_apent
1	5	3380

rank all the transaction of the customers

```sql
SELECT *,
      RANK() OVER (PARTITION BY userid ORDER BY created_date) AS rnk
FROM sales;
```

userid	created_date	product_id	rnk
1	03-11-2016	1	1
1	03-11-2017	2	2
1	03-19-2018	3	3
1	04-19-2017	2	4
1	05-20-2016	3	5
1	10-23-2019	2	6
1	11-09-2016	1	7
2	07-20-2020	3	1
2	09-10-2018	3	2
2	09-24-2017	1	3

rank all the transaction for each member whenever they are a zomato gold member for every non gold member tranction mark as ‘NA’

```sql
SELECT
  e.*,
  CASE WHEN rnk = 0 THEN 'NA'
       ELSE rnk
  END AS rnkk
FROM (
  SELECT
    c.*,
    CAST(
      CASE WHEN gold_signup_date IS NULL THEN 0
           ELSE RANK() OVER (PARTITION BY userid ORDER BY created_date DESC)
      END AS VARCHAR
    ) AS rnk
  FROM (
    SELECT
      a.userid,
      a.created_date,
      a.product_id,
      b.gold_signup_date
    FROM sales a
    LEFT JOIN goldusers_signup b ON a.userid = b.userid
                                AND a.created_date >= b.gold_signup_date
  ) AS c
) AS e;
```

userid	created_date	product_id	gold_signup_date	rnk	rnkk
1	11-09-2016	1	09-22-2017	1	1
1	10-23-2019	2	09-22-2017	2	2
1	05-20-2016	3	NULL	0	NA
1	04-19-2017	2	NULL	0	NA
1	03-19-2018	3	NULL	0	NA
1	03-11-2017	2	NULL	0	NA
1	03-11-2016	1	NULL	0	NA
2	11-08-2017	2	NULL	0	NA
2	09-24-2017	1	NULL	0	NA
2	09-10-2018	3	NULL	0	NA

Thank You