

ZOMATO SALES ANALYSIS



PROJECT OVERVIEW OBJECTIVE

ANALYZE ZOMATO SALES DATA TO UNCOVER INSIGHTS, TRENDS, AND PATTERNS THAT CAN HELP IMPROVE BUSINESS DECISIONS. THIS MIGHT INVOLVE EVALUATING RESTAURANT PERFORMANCE, CUSTOMER BEHAVIOR, SALES TRENDS, AND MORE.



OFF
10%



we have four tables :-

1. Users Table:- userid,signup_date

2. Sales Table:- userid,created_date,product_id

3. Product Table:- product_id,product_name ,pri

4. Gold_user:- userid,gold_signup_date



Creating Database And Tables

```
create database zomato; -- creating a database
use zomato; -- use database
create table users (Userid int primary key auto_increment , signup_date date not null );
insert into users (signup_date) values ("2014-09-02"),("2015-01-15"),("2014-04-11");
select * from users;

-- creating sales table --

create table sales( userid int not null , created_date date not null , product_id int not null);
insert into sales values (1,"2017-4-19",2),(3,"2019-12-18",1),(2,"2020-7-20",3),(1,"2019-10-23",2),
(1,"2018-3-19",3),(3,"2016-12-20",2),(1,"2016-11-9",1),(1,"2016-5-20",3),(2,"2017-9-24",1),
(1,"2017-3-11",2),(1,"2016-3-11",1),(3,"2016-11-10",1),(3,"2017-12-7",2),(3,"2016-12-15",2),(2,"2017-11-8",2),(2,"2018-9-10",3);
select * from sales;

-- creating a product table

create table product (product_id int not null , product_name varchar(2) not null , price int not null);
insert into product values (1,"p1",980),(2,"p2",870),(3,"p3",330);

select * from product ;

-- creating gold user table

create table gold_user (userid int  not null ,gold_signup_date date not null);

insert into gold_user values (1,"2017-09-22"),(3,"2017-04-21");

select * from gold_user;
```

Q1.WHAT IS TOTAL AMOUNT EACH CUSTOMER SPENT ON ZOMATO ?



```
select userid , sum(price) as total_price from sales inner join product on sales .product_id = product.product_id group by userid;
```

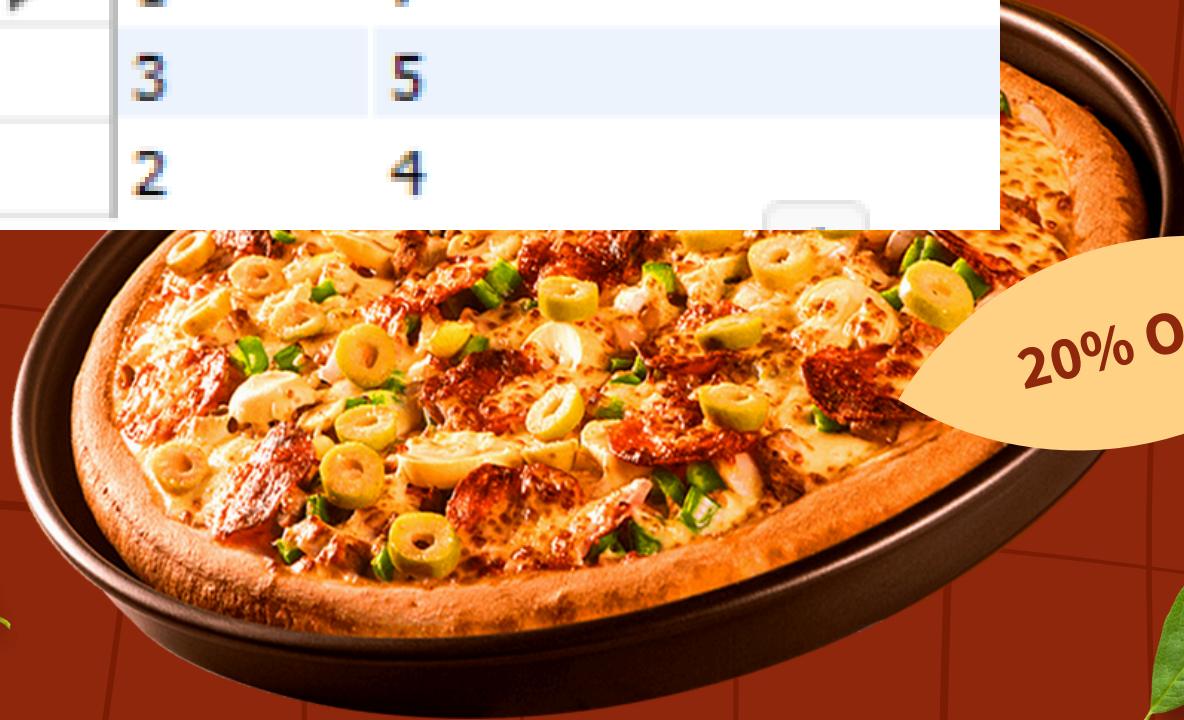
	userid	total_price
▶	1	5230
	3	4570
	2	2510



Q2. HOW MANY DAYS HAS EACH CUSTOMER VISITED ZOMATO?

```
select userid, count(created_date) from sales group by userid;
```

	Userid	count(created_date)
▶	1	7
	3	5
	2	4



Q3.WHAT WAS THE FIRST PRODUCT PURCHASED BY EACH CUSTOMER?



```
select * from
(select userid,product_id,created_date,ROW_NUMBER() OVER(PARTITION BY userid ORDER BY created_date) AS rn FROM sales) AS t WHERE rn=1 ;
```

	userid	product_id	created_date	m
▶	1	1	2016-03-11	1
	2	1	2017-09-24	1
	3	1	2016-11-10	1



Q4.WHAT IS MOST PURCHASED ITEM ON MENU & HOW MANY TIMES WAS IT PURCHASED BY ALL CUSTOMERS ?



```
select product_id ,count(product_id) from sales group by product_id;  
select userid ,product_id ,count(product_id) from sales group by userid,product_id having product_id =2;
```

userid	product_id	count(product_id)
1	2	3
3	2	3
2	2	1



Q5.WHICH ITEM WAS MOST POPULAR FOR EACH CUSTOMER?



```
select * from
(select userid ,product_id ,row_number() over(partition by userid order by count(product_id) desc) as rn
from sales group by userid,product_id) t where rn=1 ;
```

	userid	product_id	rn
	1	2	1
	2	3	1
	3	2	1

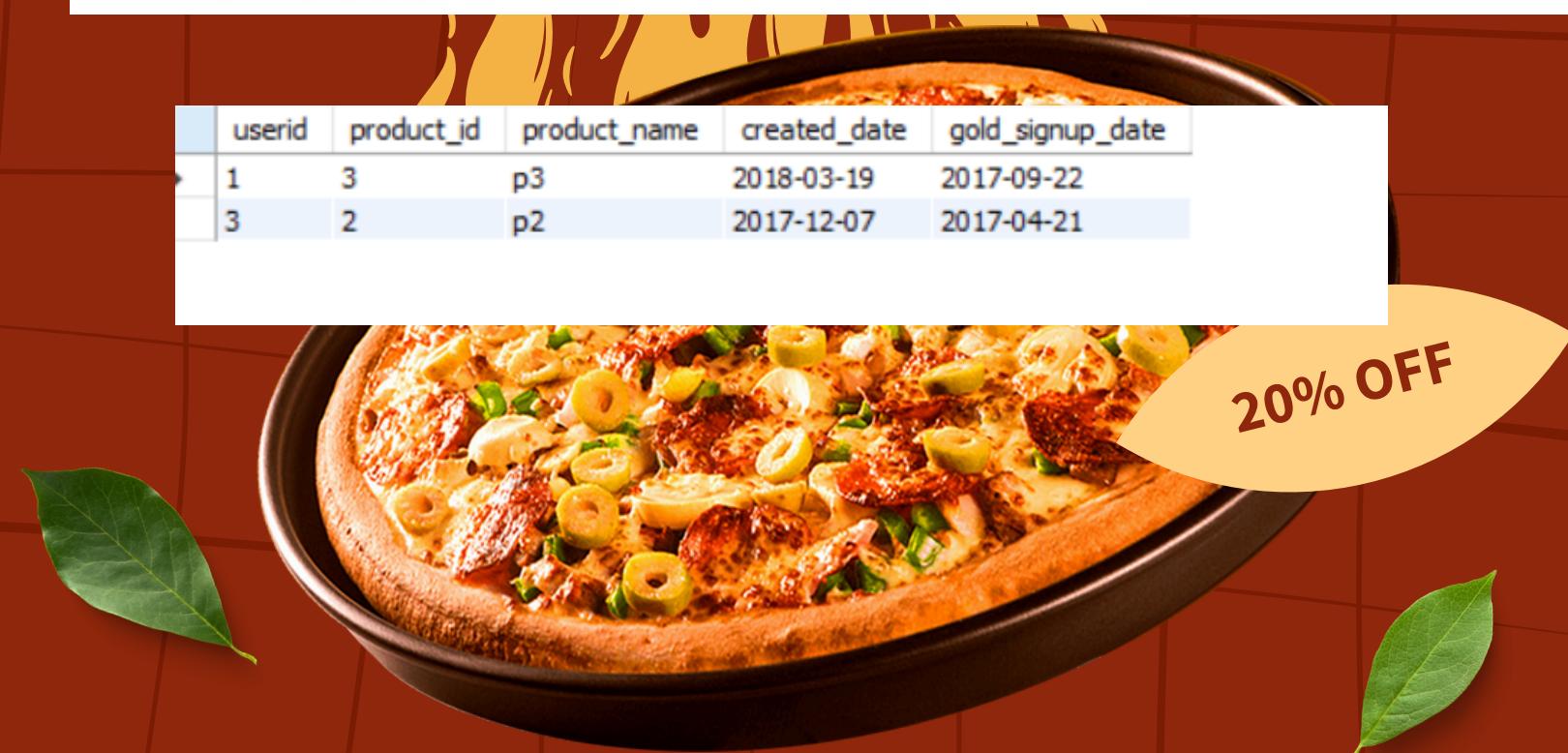


20% OFF

Q6.WHICH ITEM WAS PURCHASED FIRST BY CUSTOMER AFTER THEY BECOME A MEMBER ?

```
select * from sales;
select * from gold_user;
select * from product;
select userid,product_id,product_name ,created_date, gold_signup_date from
(select userid,product_id,product_name ,created_date, gold_signup_date ,row_number () over(partition by userid order by created_date) as rn from
(select sales.userid,sales.product_id,product.product_name,sales.created_date,gold_user.gold_signup_date  from sales
inner join product on sales.product_id = product.product_id
inner join gold_user on sales.userid=gold_user.userid
where created_date>= gold_signup_date  ) as t) as b where rn=1 ;
```

	userid	product_id	product_name	created_date	gold_signup_date
→	1	3	p3	2018-03-19	2017-09-22
	3	2	p2	2017-12-07	2017-04-21



Q7. WHICH ITEM WAS PURCHASED JUST BEFORE THE CUSTOMER BECAME A MEMBER?

```
select userid,product_id ,product_name,created_date,gold_signup_date from  
(select userid,product_id ,product_name,created_date,gold_signup_date, row_number() over(partition by userid order by created_date desc) as rn from  
(select sales.userid,sales.product_id ,product.product_name,sales.created_date,gold_user.gold_signup_date from sales inner join  
product on product.product_id=sales.product_id inner join gold_user on gold_user.userid =sales.userid  
where created_date<=gold_signup_date) as t) as b where rn=1 ;
```

userid	product_id	product_name	created_date	gold_signup_date
1	2	p2	2017-04-19	2017-09-22
3	2	p2	2016-12-20	2017-04-21

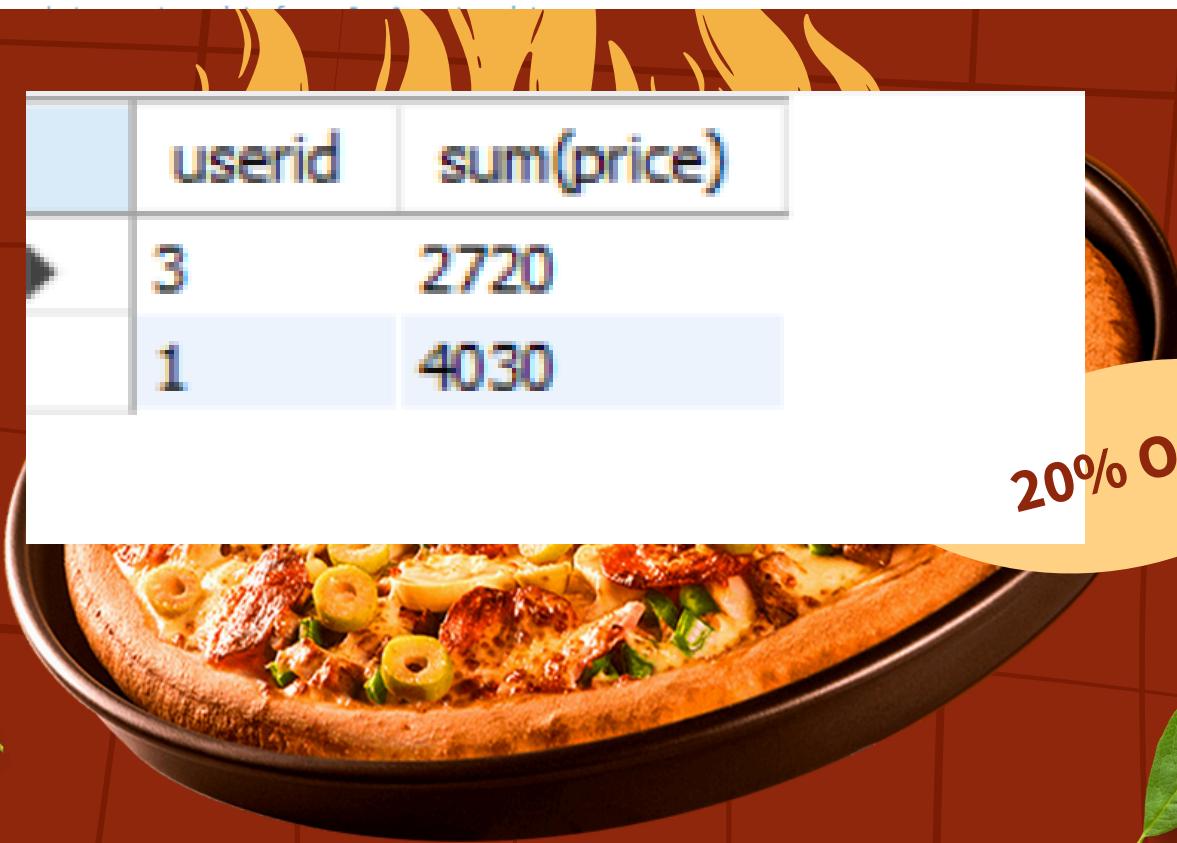


Q8. WHAT IS TOTAL ORDERS AND AMOUNT SPENT FOR EACH MEMBER BEFORE THEY BECOME A MEMBER?

```
select userid,sum(price) from
(select sales.userid,sales.product_id,product.product_name,sales.created_date,gold_user.gold_signup_date ,product.price from sales inner join
product on product.product_id=sales.product_id inner join gold_user on
gold_user.userid =sales.userid where created_date<=gold_signup_date) as a group by userid ;
```

	userid	sum(price)
	3	2720
	1	4030

20% OFF



**Q9 IF BUYING EACH PRODUCT GENERATES POINTS FOR EG 5RS=2 ZOMATO POINT
AND EACH PRODUCT HAS DIFFERENT PURCHASING POINTS FOR EG FOR P1
5RS=1 ZOMATO POINT, FOR P2 10RS=ZOMATO POINT AND P3 5RS=1 ZOMATO
POINT 2RS =1ZOMATO POINT, CALCULATE POINTS COLLECTED BY EACH
CUSTOMER AND FOR WHICH PRODUCT MOST POINTS HAVE BEEN GIVEN TILL
NOW.**

```
select * from product;
select * from sales;
select userid,product_id, round(p/point,1) from
(select userid,product_id,p ,case when product_id=1 then 5 when product_id =2 then 2 when product_id=3 then 5 else 0 end as point from
(select userid,product_id ,sum(price) as p from
(select sales.product_id,sales.userid,product.price from product inner join sales on sales.product_id = product.product_id)
as a group by userid ,product_id) as b) as c;
```

	userid	product_id	round(p/point,1)
	1	2	1305.0
	3	1	392.0
	2	3	132.0
	1	3	132.0
	3	2	1305.0
	1	1	392.0
	2	1	196.0
	2	2	435.0

20% OFF

10. RNK ALL TRANSACTION OF THE CUSTOMERS:

```
select * from sales;
select * from product;
select sales.userid,sales.product_id,product.product_name,product.price,sales.created_date,dense_rank()over( partition by userid order by created_date)
from sales inner join product on product.product_id=sales.product_id;
```

userid	product_id	product_name	price	created_date	dense_rank()over(partition by userid order by created_date)
1	1	p1	980	2016-03-11	1
1	3	p3	330	2016-05-20	2
1	1	p1	980	2016-11-09	3
1	2	p2	870	2017-03-11	4
1	2	p2	870	2017-04-19	5
1	3	p3	330	2018-03-19	6
1	2	p2	870	2019-10-23	7
2	1	p1	980	2017-09-24	1
2	2	p2	870	2017-11-08	2
2	3	p3	330	2018-09-10	3
2	3	p3	330	2020-07-20	4
3	1	p1	980	2016-11-10	1
3	2	p2	870	2016-12-15	2
3	2	p2	870	2016-12-20	3
3	2	p2	870	2017-12-07	4

20% OFF



Q11 RANK ALL TRANSACTION FOR EACH MEMBER WHENEVER THEY ARE ZOMATO GOLD MEMBER FOR EVERY NON GOLD MEMBER TRANSACTION MARK AS NA

```
- select userid,created_date,gold_signup_date ,case when gold_signup_date is null then "na" else rank() over(partition by userid order by created_date desc )
(select sales.userid,sales.product_id,sales.created_date,gold_user.gold_signup_date  from sales
left join gold_user on sales.userid=gold_user.userid
and created_date>= gold_signup_date )as t
```

userid	created_date	gold_signup_date	rankk
1	2019-10-23	2017-09-22	1
1	2018-03-19	2017-09-22	2
1	2017-04-19	NULL	na
1	2017-03-11	NULL	na
1	2016-11-09	NULL	na
1	2016-05-20	NULL	na
1	2016-03-11	NULL	na
2	2020-07-20	NULL	na
2	2018-09-10	NULL	na
2	2017-11-08	NULL	na
2	2017-09-24	NULL	na
3	2019-12-18	2017-04-21	1
3	2017-12-07	2017-04-21	2
3	2016-12-20	NULL	na
3	2016-12-15	NULL	na
3	2016-11-10	NULL	na

20% OFF