```
-- create database if not exists zomato;
-- use zomato;

-- drop table if exists goldusers_signup;
-- drop table if exists users;
-- drop table if exists sales;
-- drop table if exists product;



-- CREATE TABLE goldusers_signup(
-- userid integer,
-- gold_signup_date date
-- );

-- CREATE TABLE users(
-- userid integer,
-- signup_date date
-- );

-- CREATE TABLE sales(
-- userid integer,
-- created_date date,
-- product_id integer
-- );

-- CREATE TABLE product(
-- product_id integer,
-- product_name text,
-- price integer
-- );

-- INSERT INTO users(userid, signup_date) VALUES
-- (1, '2014-09-02'),
-- (2, '2015-01-15'),
-- (3, '2014-04-11');

-- INSERT INTO goldusers_signup(userid, gold_signup_date) VALUES
-- (1, '2017-09-22'),
-- (3, '2017-04-21');

-- INSERT INTO product(product_id,product_name,price)
-- VALUES
-- (1,'p1',980),
-- (2,'p2',870),
-- (3,'p3',330);


-- INSERT INTO sales(userid, created_date, product_id)
-- VALUES
-- (1, '2017-04-19', 2),
-- (3, '2019-12-18', 1),
-- (2, '2020-07-20', 3),
-- (1, '2019-10-23', 2),
```

```
-- (1, '2018-03-19', 3),
-- (3, '2016-12-20', 2),
-- (1, '2016-11-09', 1),
-- (1, '2016-05-20', 3),
-- (2, '2017-09-24', 1),
-- (1, '2017-03-11', 2),
-- (1, '2016-03-11', 1),
-- (3, '2016-11-10', 1),
-- (3, '2017-12-07', 2),
-- (3, '2016-12-15', 2),
-- (2, '2017-11-08', 2),
-- (2, '2018-09-10', 3);



-- select * from users;
-- select * from goldusers_signup;
-- select * from product;
-- select * from sales;


-- Q1) What is the total amount of each customer spent on zomato
select userid, sum(p.price) as user_sales from sales s inner join
product p on s.product_id = p.product_id group by userid;

-- Q2) How many days each customer visited zomato
select userid, count(distinct created_date) as no_of_visits from
sales group by userid;

-- Q3) What was the first product purchased by each customer
with cte as
(select userid, product_id, created_date, row_number()
over(partition by userid order by created_date) as rn from sales)
select userid, product_name as first_product from cte as c join
product as p on c.product_id = p.product_id where rn = 1 ;

-- Q4) What is the most purchase item on the menu and how many times
was it purchased by all customers.
with cte as
(select product_id, count(*) as cnt from sales group by product_id
order by cnt desc limit 1)
select userid, count(*) as cnt from cte as c join sales as s on
c.product_id = s.product_id group by userid;

-- Q5) Which item was most popular for each customer
with cte2 as
(with cte as
(select userid, product_id, count(*) as cnt from sales group by
userid, product_id)
select userid, product_id, cnt, row_number() over(partition by
userid order by cnt desc) as rn from cte
)
select userid, product_id, cnt as no_of_time_purchased from cte2
```

```
where rn = 1;


-- Q6) Which item was purchased first by customer when they became
gold member
with cte as
(select g.userid, gold_signup_date, created_date, product_id,
row_number() over(partition by userid order by created_date) as rn
from goldusers_signup g left join sales s on g.userid = s.userid
where created_date > gold_signup_date)
select userid, product_id from cte where rn = 1;


-- Q7) Which item was purchased just before the customer became
member
with cte as
(select g.userid, created_date, gold_signup_date, product_id,
row_number() over(partition by userid order by created_date desc) as
rn from goldusers_signup g left join sales s on g.userid = s.userid
where created_date < gold_signup_date)
select userid, product_id from cte where rn = 1;

-- Q8) What is the total orders and amount spent for each member
before they become gold member
with sales_before_gold as
(select g.userid, created_date, gold_signup_date, product_id,
row_number() over(partition by userid order by created_date desc) as
rn from goldusers_signup g left join sales s on g.userid = s.userid
where created_date < gold_signup_date)
select userid, count(*) as total_orders, sum(price) as total_spent
from sales_before_gold s left join product p on s.product_id =
p.product_id group by userid;


--  Q9) Calculated points collected by each customer and for which
product most points have been given so far.

select userid,
sum(case when s.product_id = 1 then round(price * .2)
when s.product_id = 2 then round(price * .5)
when s.product_id = 3 then round(price * .2)
end) as points
from sales s join product p on s.product_id = p.product_id group by
userid;

select s.product_id,
sum(case when s.product_id = 1 then round(price * .2)
when s.product_id = 2 then round(price * .5)
when s.product_id = 3 then round(price * .2)
end) as points
from sales s join product p on s.product_id = p.product_id group by
s.product_id order by points desc limit 1;
```

```
-- Q10) In the first year after member joins gold program(including
joining date) irrespective of whate customer has purchased they earn
5 zomato points for every 10rs. Who earned more, 1 or 3? What was
their points earning in their first year.

with cte as
(select g.userid, gold_signup_date, created_date, product_id from
goldusers_signup g left join sales s on g.userid = s.userid where
created_date between gold_signup_date AND DATE_ADD(gold_signup_date,
INTERVAL 1 YEAR)
)
select userid, sum(round(price*.5)) as points_earned from cte c join
product p on c.product_id = p.product_id group by userid order by
points_earned;


-- Q12) Rank all transactions for each member if they are gold
member. If he/she is not gold-member then put NA
-- Below two ways of solving same problem

select s.*,
case when gold_signup_date is not null then rank() over(partition by
userid order by created_date desc) else "na" end as rn
from sales s left join goldusers_signup g on s.userid = g.userid and
s.created_date > g.gold_signup_date;

with cte as
(select s.*,
rank() over(partition by userid order by created_date) as rn,
gold_signup_date from sales s left join goldusers_signup g on
s.userid = g.userid where created_date > gold_signup_date)
select s.*, case when rn is null then "na" else rn end as rn from
sales s left join cte c on s.userid = c.userid and s.created_date =
c.created_date and s.product_id = c.product_id;
```