A. CREATE TABLE

```
drop table if exists goldusers_signup;

CREATE TABLE goldusers_signup(userid integer,gold_signup_date date);


drop table if exists users;

CREATE TABLE users(userid integer,signup_date date);


drop table if exists sales;

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


drop table if exists product;

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

B. INSERT DATA

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


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




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,'04-19-2017',2),
(3,'12-18-2019',1),
(2,'07-20-2020',3),
(1,'10-23-2019',2),
(1,'03-19-2018',3),
(3,'12-20-2016',2),
(1,'11-09-2016',1),
(1,'05-20-2016',3),
(2,'09-24-2017',1),
(1,'03-11-2017',2),
(1,'03-11-2016',1),
(3,'11-10-2016',1),
(3,'12-07-2017',2),
(3,'12-15-2016',2),
(2,'11-08-2017',2),
(2,'09-10-2018',3);
```

C. SELECT TABLE

```
select * from sales;

select * from product;

select * from goldusers_signup;

select * from users;
```

D. QUERY

1. What is the total amount each customer spent on Zomato ?

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

userid	Amount_Spent
1	5230
2	2510
3	4570

2. How many days has each customer visited Zomato ?

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

userid	distinct_days
1	7
2	4
3	5

3. What was the first product purchased by each customer ?

```
select * from
(select *, RANK() over(partition by userid order by created_date) rnk from sales)
a where rnk = 1;
```

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

4. What is the most purchased item on the menu and how many times was it purchased by all customer?

```
Select userid, count(product_id) cnt from sales where product_id =
(select TOP 1 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

5. Which item was the most popular for each customer?

```
select * from
(select *, rank() over(partition by userid order by cnt desc) rnk
from
(select userid, product_id, count(product_id) cnt from sales group by userid,
product_id) A) B where rnk = 1;
```

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

6. Which item was purchased first by the customer after they become a member ?

```
Select * from
(Select c.*, rank() over(partition by userid order by created_date) 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) c)d where rnk = 1;
```

userid	created_date	product_id	gold_signup_date	rnk
1	2018-03-19	3	2017-09-22	1
3	2017-12-07	2	2017-04-21	1

7. Which item was purchased first before the customer become a member ?

```
Select * from
(Select c.*, rank() over(partition by userid order by created_date desc) 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) c)d where rnk = 1;
```

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

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

```
select userid, count(created_date) order_purchased, sum(price) total_amt_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) c inner join product d on c.product_id = d.product_id) e group
by userid
```

userid	order_purchased	total_amt_spent
1	5	4030
3	3	2720

9. 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 = 5 zomato points and p3 5rs = 1 zomato point. 2rs = 1 zomato point.

```
Select userid, sum(Total_Points)*2.5 as Total_Points_Earn from
(select e.*, amt/Points Total_Points from
(select d.*, case when product_id =1 then 5 when product_id =2 then 2 when
product_id =3 then 5 else 0 end as Points from
(select c.userid, c.product_id, sum(price) as amt from
(select a.*, b.price from sales a inner join product b on a.product_id =
b.product_id) c group by c.userid, c.product_id) d) e) d group by userid;
```

10. In the first one year after a customer joins the gold program (including their join date) irrespective of what the customer has purchased they earn 5 zomato points for every 10 rs spent who earned more 1 or 3 and what was their points earnings in their first year ?

    1 zomato point = 2 rs

```
select c.*, d.price*0.5 total_point_earned 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 and created_date<= DATEADD(year, 1, gold_signup_date)) c
inner join product d on c.product_id = D.product_id;
```

userid	created_date	product_id	gold_signup_date	total_point_earned
1	2018-03-19	3	2017-09-22	165.0
3	2017-12-07	2	2017-04-21	435.0

11. Rank all the transaction of the customer

```
select *, DENSE_RANK() over(partition by userid order by created_date) rnk from
sales;
```

userid	created_date	product_id	mk
1	2016-03-11	1	1
1	2016-05-20	3	2
1	2016-11-09	1	3
1	2017-03-11	2	4
1	2017-04-19	2	5
1	2018-03-19	3	6
1	2019-10-23	2	7
2	2017-09-24	1	1
2	2017-11-08	2	2
2	2018-09-10	3	3
2	2020-07-20	3	4
3	2016-11-10	1	1
3	2016-12-15	2	2
3	2016-12-20	2	3
3	2017-12-07	2	4
3	2019-12-18	1	5

12. Rank all the transaction for each member whenever they are a Zomato gold member for every non gold member transaction mark as NA

```
select e.userid,created_date, product_id,gold_signup_date, case when rnk =0 then
'na' else rnk end as New_Rank from
(select c.*, cast((case when gold_signup_date is null then 0 else Dense_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) c) e
```

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