booking table sample data:

booking_id	user_id	booking_timestamp	status	bike_id
4330255	201741	16/03/19 16:33	completed	21858
4330256	643899	16/03/19 16:33	completed	20830
4330258	371678	16/03/19 16:33	cancelled	22540

user_type table sample data:

user_id	type
201741	Regular
643899	Irregular
371678	Power

1. Find the date wise cancelled & completed bookings for the last 10 days. Output should look like as shown below. (10 Marks)

Date	Completed bookings	Cancelled bookings
X	Y	Z

```
select date(booking_timestamp) as 'Date',
        count(case status when 'completed' then 1 else null end) as 'Completed bookings',
        count(case status when 'cancelled' then 1 else null end) as  'Cancelled bookings'
from booking
where date(booking_timestamp) >= (curdate( ) - interval 10 day)
group by date(booking_timestamp)
```

2. For the last year 2020 for every user_ids, find the month in which a user did maximum number of completed bookings and also that month name. Output should look like as shown below. (20 Marks)

User_id	Month name	Completed bookings
123	March-20	100
125	Nov-20	250

```
With completed_bookings_table as (
select user_id,  date_format(booking_timestamp, '%b-%y') as month_name,
        count(case status when 'completed' then 1 else null end) as completed_bookings
from booking
where year(booking_timestamp) = 2020),

monthly_bookings as (
Select user_id, month_name, sum(completed_bookings) as bookings_completed
From completed_bookings_table
Group by user_id, month_name),

ranked_list as (
Select user_id, month_name, bookings_completed, rank() over (partition by user_id order by
month_name) as rank_order
From monthly_bookings
)

Select user_id, month_name, max(bookings_completed) as completed_bookings
From ranked_list
Group by user_id, month_name
Where rank_order = 1
```

3. For yesterday, find the count of users who did 1 completed bookings, 2 completed bookings & more than 2 completed bookings. Output should look like as shown below. (20 Marks)

Date	Booking bucket	Users count
Yesterday's date	1	x
Yesterday's date	2	y
Yesterday's date	>2	z

```
With counter as (
select  user_id,
        count(*) as user_count
from booking
where date(booking_timestamp) = SUBDATE(curdate(),1)
            and status = 'completed'
group by user_id),

final_table as (
        select
                case when user_count= 1 then '1'
                when user_count = 2 then '2'
                when user_count > 2 then '>2'  end as booking_count,
                count (user_count) as users_count
        from counter
        group by booking_count)

select SUBDATE(curdate(),1) as Date,
        booking_count,
        users_count
from final_table
```

4. For the last quarter, find the completed bookings per user per month for each user type. Output should look like as shown below. (30 Marks)

Month	User type	Completed bookings per user
July-21	Regular	x
July-21	Power	y
Aug-21	Regular	z

```
Set @last_quarter = 0,
Select @last_quarter = quarter(curdate())

With new_table as (
        Select b.booking_id as booking_id, b.user_id as user_id, b.booking_timestamp as
booking_timestamp, b.status as status, b.bike_id as bike_id, u.type as type
        From booking b
        Left join user_type u on b.user_id = u.user_id),

completed_table as (
        Select date_format(booking_timestamp, '%b-%y') as month_name, type as user_type,
count(*) as completed_bookings_per_user
        From new_table
        Where status = 'completed'
        group by 1,2 )

Select *
From completed_table
```

5. Suppose the booking table has all the lifetime booking data of all users. Find the count of new users and their completed bookings date wise for the last 10 days. A user will be called a new user for a date when his first completed booking of his lifetime is on that date, thus the next day if he does a booking he will be called as repeat users for the next day. Output should look like as shown below.(50 Marks)

Date	New user count	New users completed bookings
1-oct-21	x	x
2-oct-21	y	y
3-oct-21	z	z

```
With completed_booking_table as (
        Select *, rank() over(group by user_id, order by booking_timestamp desc) as bookings_rank
        From booking
        Where status = 'completed'),


User_count_table as (
        Select *, rank() over(group by user_id, order by booking_timestamp desc) as bookings_rank
        From booking),


completed_booking_past_10_days as (
        select *
        from completed_booking_table
        where date(booking_timestamp) >= (curdate( ) - interval 10 day)
        and bookings_rank = 1),

user_count_past_10_days as (
        select *
        from user_count_table
        where date(booking_timestamp) >= (curdate( ) - interval 10 day)
        and bookings_rank = 1),

new_users_completed as (
        select date(booking_timestamp) as date, count(*) as new_users_count
        from completed_booking_past_10_days
        group by 1),

new_users_count as (
        select date(booking_timestamp) as date, count(*) as new_users_completed_booking
        from user_count_past_10_days
        group by 1)


Select *
From new_users_count n
Left join new_users_completed c on n.date = c.date
```

6. Using the same booking table. For month Jan-2020, find the MoM retention of acquired users for the next 11 month i.e till Dec-2020. For example, suppose 1000 users acquired in Jan-20(acquired users= users who did their first completed booking of their lifetime in the month of Jan-2020) then how many of these 1000 users kept coming back in the following month of the year 2020. Output should look like as shown below. (70 Marks)

Month	Users count	% of users
Jan-20	1000	100%
Feb-20	800	80%
March-20	500	50%
--	--	--
--	--	--
Dec-20	300	30%

```
— getting user_id for all the users joining in Jan 2020
With new_users_from_jan as (
Select
        user_id, new_user_start_date

From (
        Select user_id,
                min(booking_timestamp) as new_user_start_date
        From booking
        Group by user_id) sq

Where date_part('year', new_user_start_date) = 2020
         and date_part('month, new_user_start_date) = 'January'),

— filtering new_users and restricting year to 2020
users_logging_details as (

        Select *
        From booking b
        Inner join new_users_from_jan n on b.user_id = n.user_id
        Where date_part('year', b.booking_timestamp) = 2020),

user_count_table as (

        Select date_format(booking_timestamp, '%b-%y') as Month, count(*) as Users_count
        from users_logging_details
        group by Month
)

Set @New_users = 0;
Select @New_users = Users_count from user_count_table where date_part('month', Month) =
'January')

Select month, users_count, (users_count/@New_users)* 100 as "% of users"
From user_count_table
```