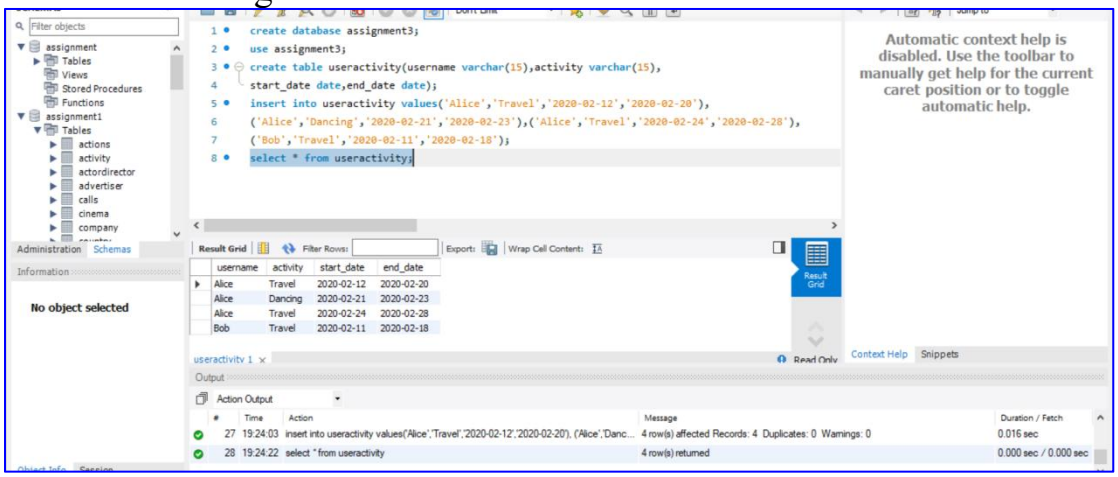
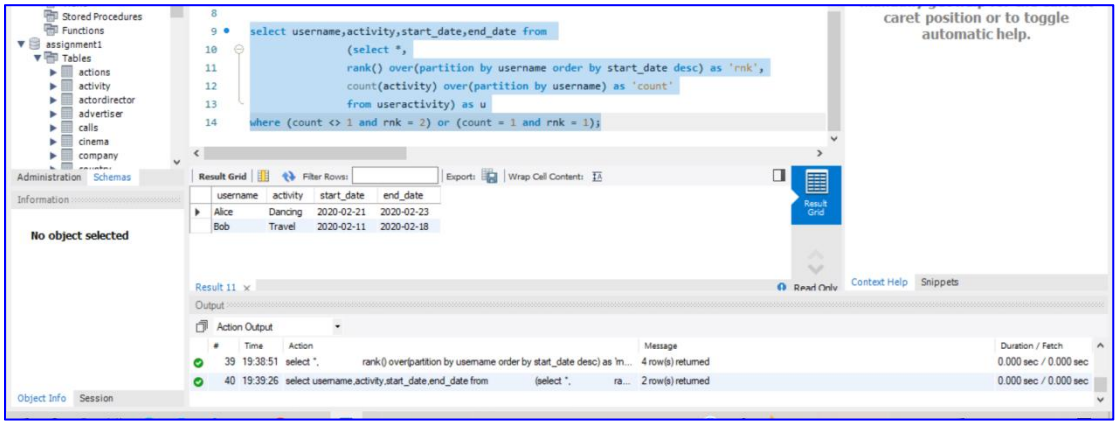
SQL ASSIGNMENT SET 3

Q101) Write an SQL query to show the second most recent activity of each user. If the user only has one activity, return that one. A user cannot perform more than one activity at the same time.
Return the result table in any order. The query result format is in the following example.

Sol) At first create an empty table named 'UserActivity' and then insert records in it using multi insert statement.



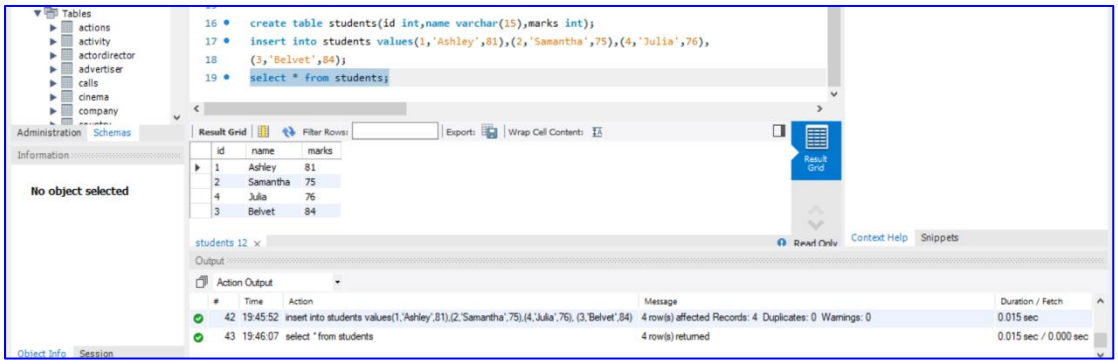

```
Select username,activity,start_date,end_date from
            (select *,
            rank() over(partition by username order by start_date desc)
              as 'rnk',
           count(activity) over(partition by username) as 'count'
           from useractivity) as u
where (count <> 1 and rnk = 2) or (count = 1 and rnk = 1);
```



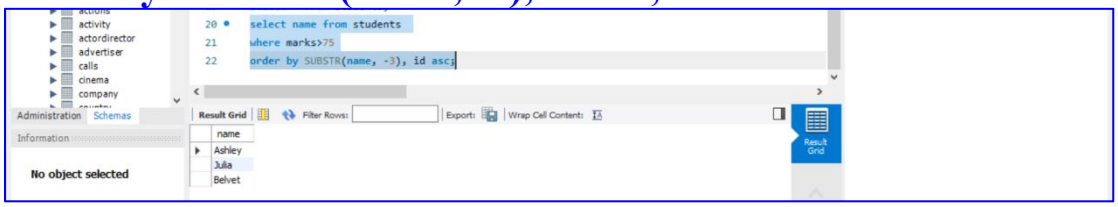
Q102) SAME QUESTION AS Q101

Q103) Query the Name of any student in STUDENTS who scored higher than 75 Marks. Order your output by the last three characters of each name. If two or more students both have names ending in the same last three characters (i.e.: Bobby, Robby, etc.), secondary sort them by ascending order.

Sol) At first create an empty table named ‘students’ and then insert records in it using multi insert statement.




```
select name from students
where marks>75
order by SUBSTR(name, -3), id asc;
```



Q104) SAME QUESTION AS Q82

Q105) SAME QUESTION AS Q83

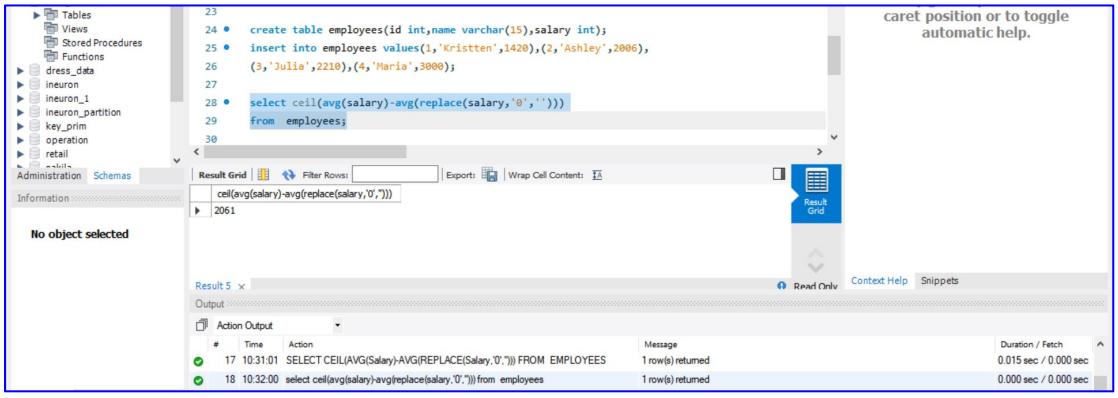
Q106) SAME QUESTION AS Q84

Q107) Samantha was tasked with calculating the average monthly salaries for all employees in the EMPLOYEES table, but did not realize her keyboard's 0 key was broken until after completing the calculation. She wants your help finding the difference between her miscalculation (using salaries with any zeros removed), and the actual average salary.

Write a query calculating the amount of error (i.e.: actual - miscalculated average monthly salaries),
and round it up to the next integer.

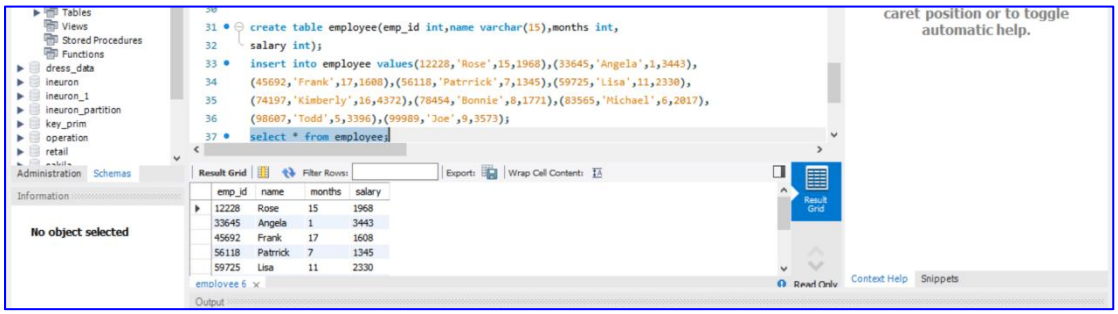
Sol) At first create an empty table named ‘employees’ and then insert records in it using multi insert statement.

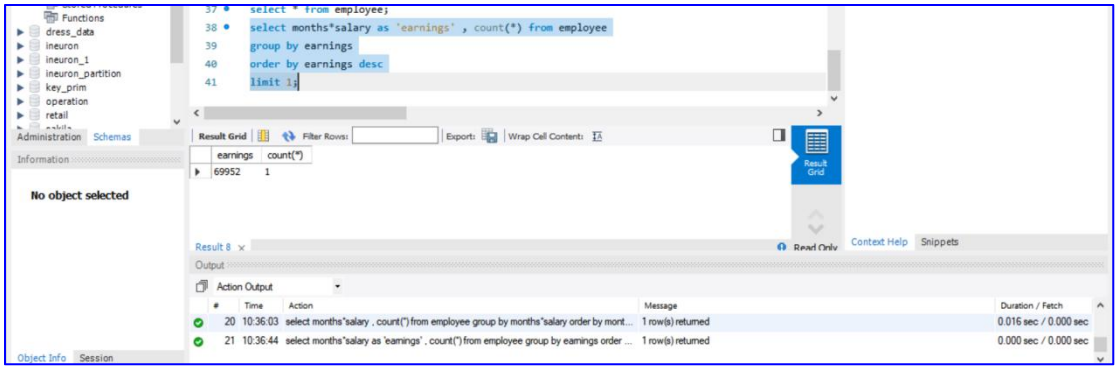
```
select ceil(avg(salary)-avg(replace(salary,'0','')))
From employees
```



Q108) Write a query to find the maximum total earnings for all employees as well as the total number of employees who have maximum total earnings. Then print these values as 2 space-separated integers.

Sol) At first create an empty table named ‘employee’ and then insert records in it using multi insert statement.

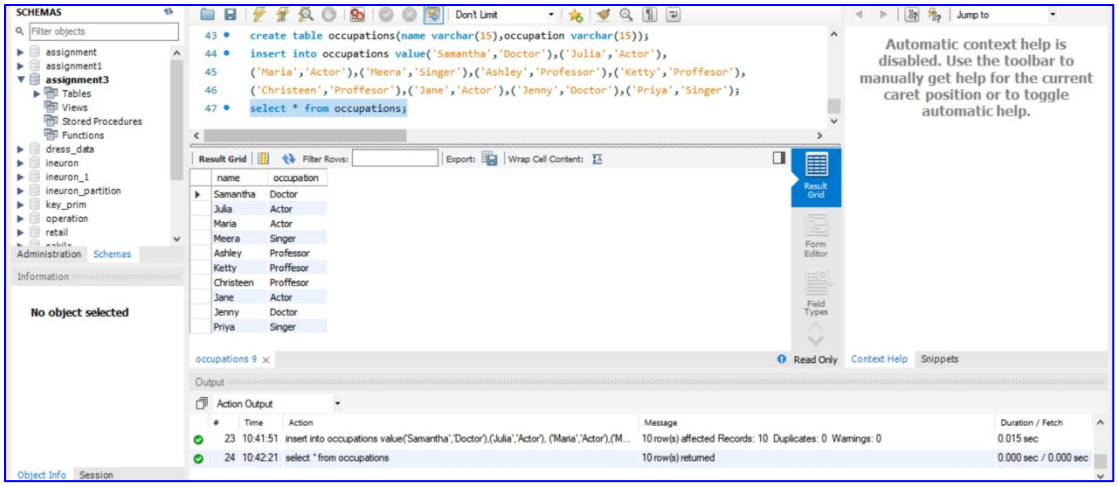




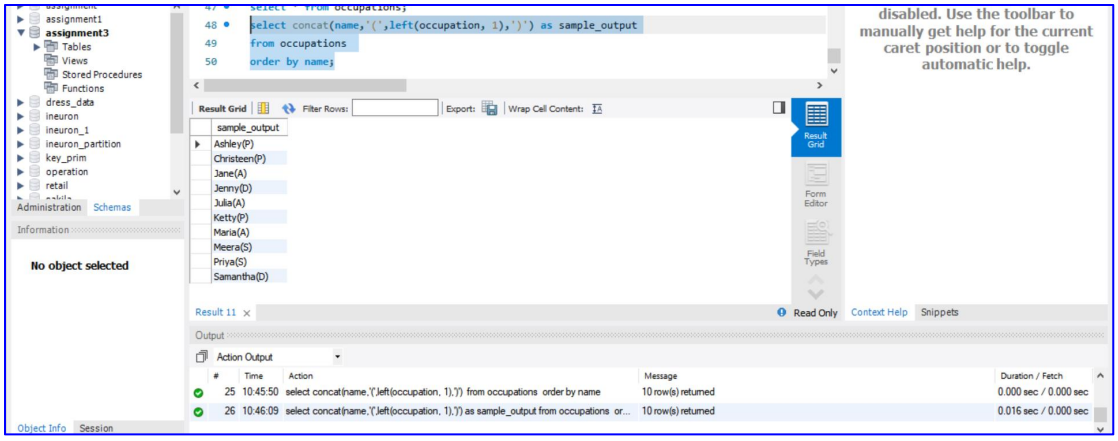

```
select months*salary as 'earnings' , count(*) from employee
group by earnings
order by earnings desc
limit 1;
```

Q109) Query the number of occurrences of each occupation in OCCUPATIONS. Sort the occurrences in ascending order, and output them in the following format:

Sol) At first create an empty table named ‘occupations’ and then insert records in it using multi insert statement.



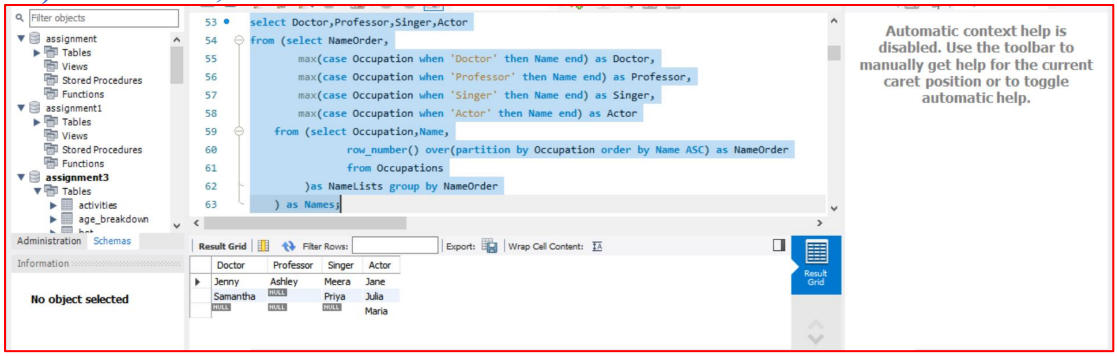

```
select concat(name,'(',left(occupation, 1),')') as
sample_output
from occupations
order by name;
```



Q110) Pivot the Occupation column in OCCUPATIONS so that each Name is sorted alphabetically and displayed underneath its corresponding Occupation. The output column headers should be Doctor, Professor, Singer, and Actor, respectively.
Note: Print NULL when there are no more names corresponding to an occupation.

Sol)The table 'Occupations' used for this query is already created in previous question.

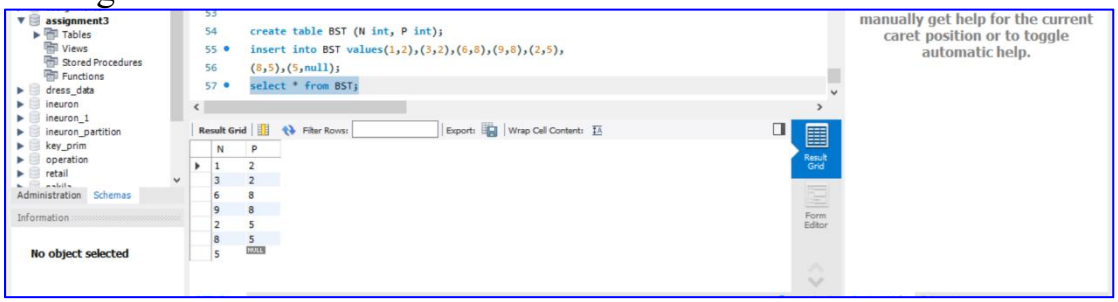
```
select Doctor,Professor,Singer,Actor
from (select NameOrder,
      max(case Occupation when 'Doctor' then Name end) as Doctor,
      max(case Occupation when 'Professor' then Name end) as
Professor,
      max(case Occupation when 'Singer' then Name end) as Singer,
      max(case Occupation when 'Actor' then Name end) as Actor
   from (select Occupation,Name,
            row_number() over(partition by Occupation order by Name
ASC) as NameOrder
                        from Occupations
       )as NameLists group by NameOrder
   ) as Names;
```



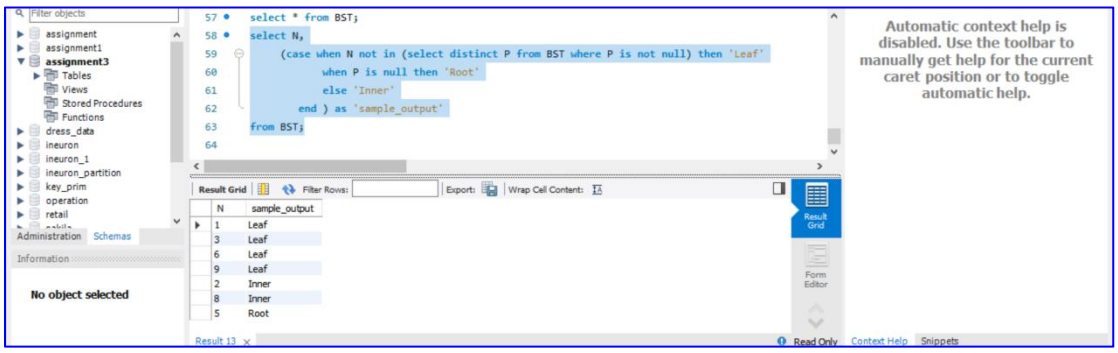
Q111) Write a query to find the node type of Binary Tree ordered by the value of the node. Output one of the following for each node:

- **Root: If node is root node.**
- **Leaf: If node is leaf node.**
- **Inner: If node is neither root nor leaf node**

Sol) At first create an empty table named 'BST' and then insert records in it using multi insert statement.



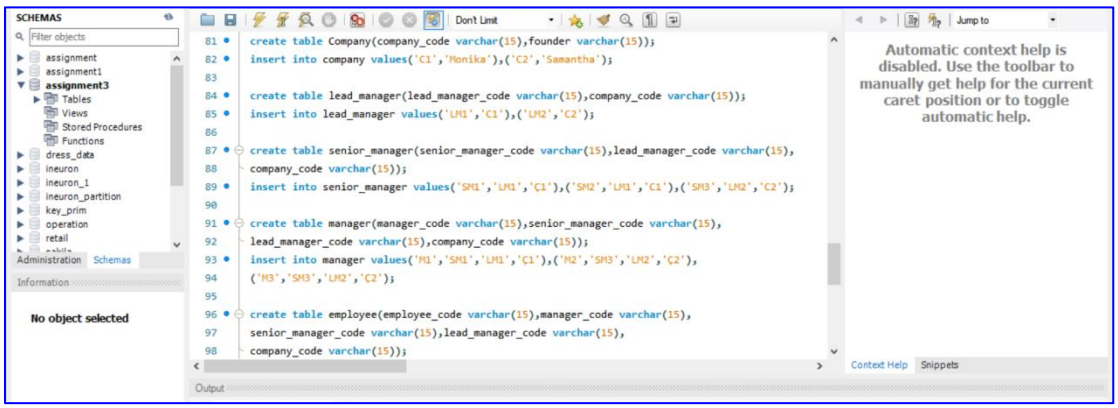

```
select N,
    (case when N not in (select distinct P from BST where P is
                not null) then 'Leaf'
          when P is null then 'Root'
          else 'Inner'
        end ) as 'sample_output'
from BST;
```



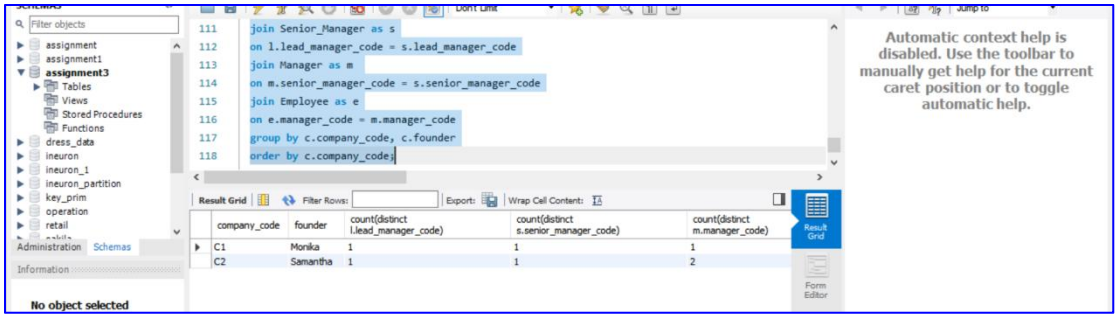
Q112) Given the table schemas below, write a query to print the company_code, founder name, total number of lead managers, total

number of senior managers, total number of managers, and total number of employees. Order your output by ascending company_code.

Sol) In this question, we have to first create 5 empty tables naming 'Company','Lead_manager','Senior_manager','Manager', 'Emp' and then insert records in it using multi insert statement.




```
select c.company_code, c.founder,
       count(distinct l.lead_manager_code),
       count(distinct s.senior_manager_code),
       count(distinct m.manager_code),
       count(distinct e.employee_code)
from Company as c join Lead_Manager as l
on c.company_code = l.company_code
join Senior_Manager as s on l.lead_manager_code
=s.lead_manager_code
join Manager as m on m.senior_manager_code =
s.senior_manager_code
join Employee as e on e.manager_code = m.manager_code
group by c.company_code, c.founder
order by c.company_code;
```



Q113) Write a query to print all prime numbers less than or equal to 1000. Print your result on a single line, and use the ampersand () character as your separator (instead of a space). For example, the output for all prime numbers <=10 would be: 2&3&5&7

Q114) P(R) represents a pattern drawn by Julia in R rows. The following pattern represents P(5):

*****

*** ***

*** * ***

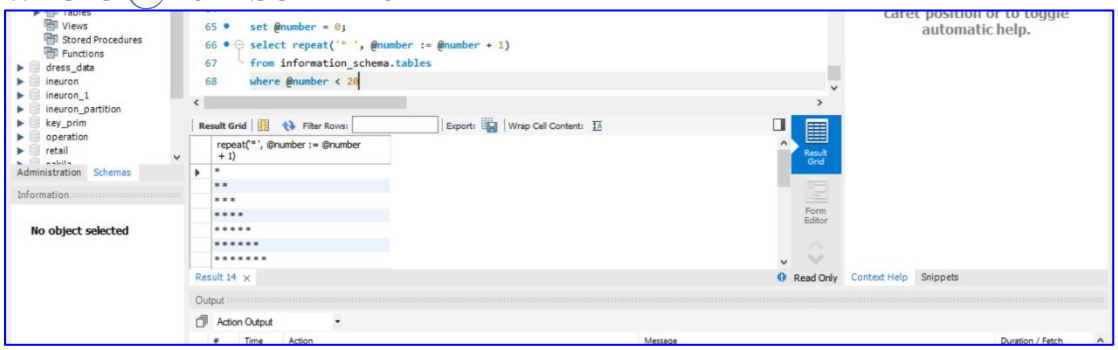
*** * * ***

*** * * * ***

Write a query to print the pattern P(20)

Sol)

```
set @number = 0;
select repeat('* ', @number := @number + 1)
from information_schema.tables
where @number < 20
```



Q115) P(R) represents a pattern drawn by Julia in R rows. The following pattern represents P(5):

*** * * * ***

* * * *

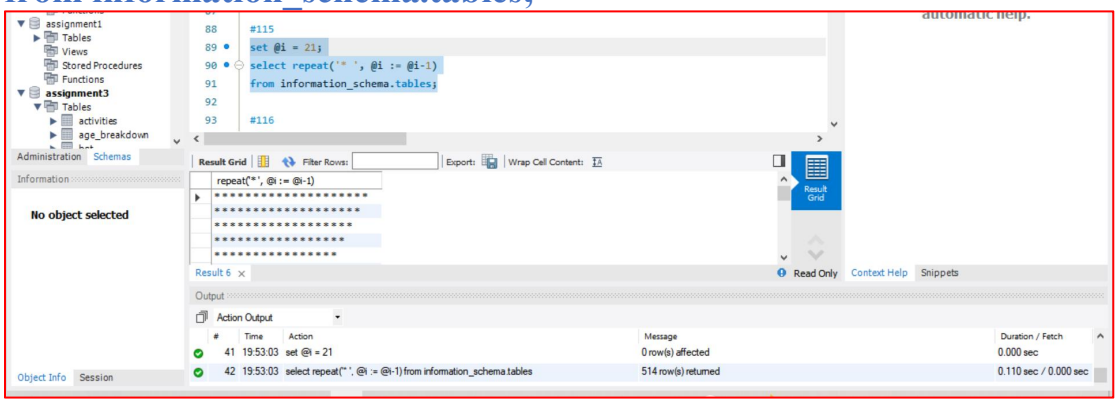
* * *

* *

*

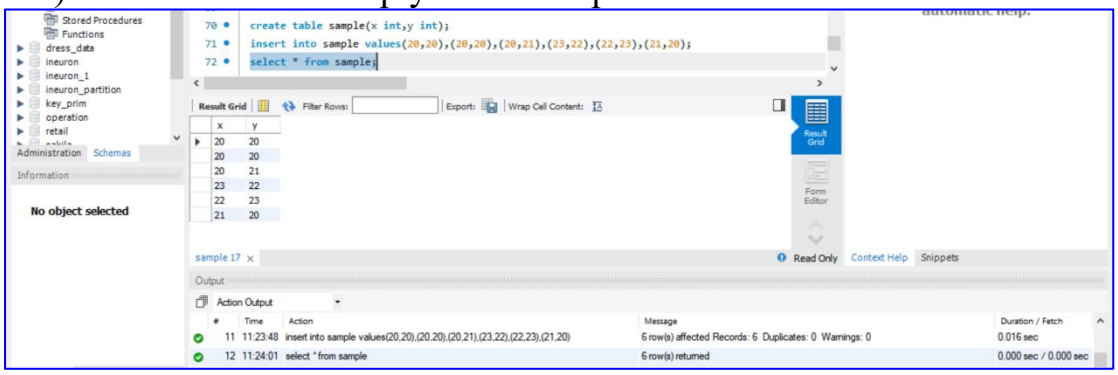
Write a query to print the pattern P(20)

Sol)
```
set @i = 21;
select repeat('* ', @i := @i-1)
from information_schema.tables;
```

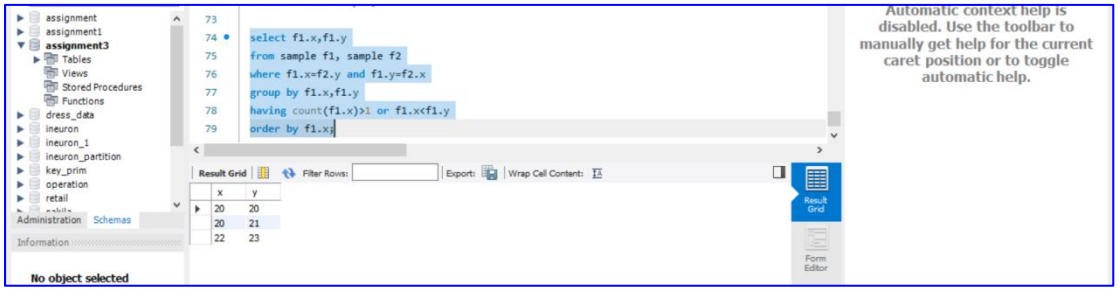


Q116) Two pairs (X1, Y1) and (X2, Y2) are said to be symmetric pairs if X1 = Y2 and X2 = Y1.
Write a query to output all such symmetric pairs in ascending order by the value of X. List the rows such that X1 ≤ Y1.

Sol) At first create an empty table ‘sample’ and insert values in it.




```
select f1.x,f1.y
from sample f1, sample f2
where f1.x=f2.y and f1.y=f2.x
group by f1.x,f1.y
having count(f1.x)>1 or f1.x<f1.y
order by f1.x;
```



Q117) SAME QUESTION AS Q103

Q118) SAME QUESTION AS Q104

Q119) SAME QUESTION AS Q105

Q120) SAME QUESTION AS Q106

Q121) SAME QUESTION AS Q85

Q122) SAME QUESTION AS Q86

Q123) SAME QUESTION AS Q87

Q124) SAME QUESTION AS Q88

Q125) SAME QUESTION AS Q89

Q126) SAME QUESTION AS Q90

Q127) SAME QUESTION AS Q91

Q128) SAME QUESTION AS Q92

Q129) SAME QUESTION AS Q93

Q130) SAME QUESTION AS Q94

Q131) SAME QUESTION AS Q95

Q132) SAME QUESTION AS Q96

Q133) SAME QUESTION AS Q97

Q134) SAME QUESTION AS Q98

Q135) SAME QUESTION AS Q99

Q136) SAME QUESTION AS Q100

Q137) SAME QUESTION AS Q101

Q138) SAME QUESTION AS Q137

Q139) SAME QUESTION AS Q103

Q140) SAME QUESTION AS Q104

Q141) SAME QUESTION AS Q105

Q142) SAME QUESTION AS Q106

Q143) SAME QUESTION AS Q107

Q144) SAME QUESTION AS Q108

Q145) SAME QUESTION AS Q109

Q146) SAME QUESTION AS Q110

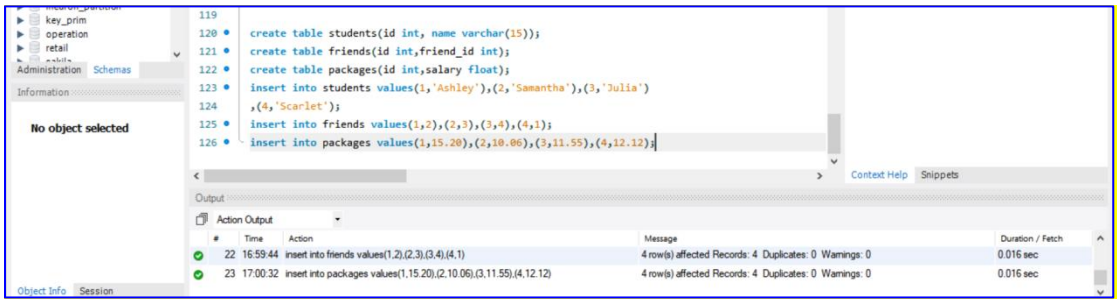
Q147) SAME QUESTION AS Q111

Q148) SAME QUESTION AS Q112

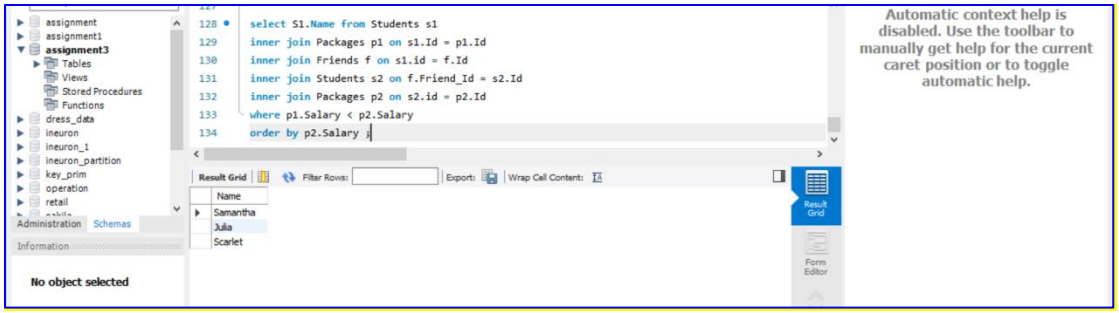
Q149) SAME QUESTION AS Q113

Q150) Write a query to output the names of those students whose best friends got offered a higher salary than them. Names must be ordered by the salary amount offered to the best friends. It is guaranteed that no two students get the same salary offer.

Sol) At first create an empty 3 table ‘Students’,’friends’ & ‘packages’ and insert values in it.

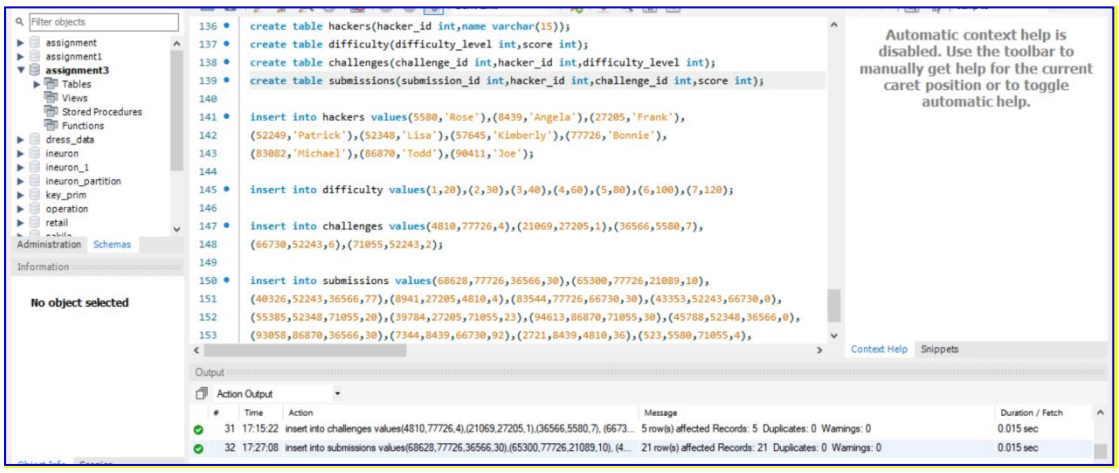



```
select S1.Name from Students s1
inner join Packages p1 on s1.Id = p1.Id
inner join Friends f on s1.id = f.Id
inner join Students s2 on f.Friend_Id = s2.Id
inner join Packages p2 on s2.id = p2.Id
where p1.Salary < p2.Salary
order by p2.Salary ;
```

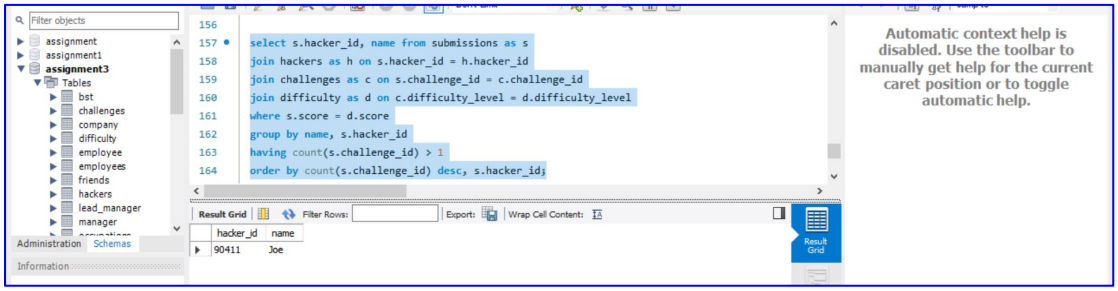


Q151) Julia just finished conducting a coding contest, and she needs your help assembling the leaderboard! Write a query to print the respective hacker_id and name of hackers who achieved full scores for more than one challenge. Order your output in descending order by the total number of challenges in which the hacker earned a full score. If more than one hacker received full scores in the same number of challenges, then sort them by ascending hacker_id.

Sol) At first create an empty 4 table'Hackers','Difficulty' ,'Challenges'& 'Submissions' and insert values in it.

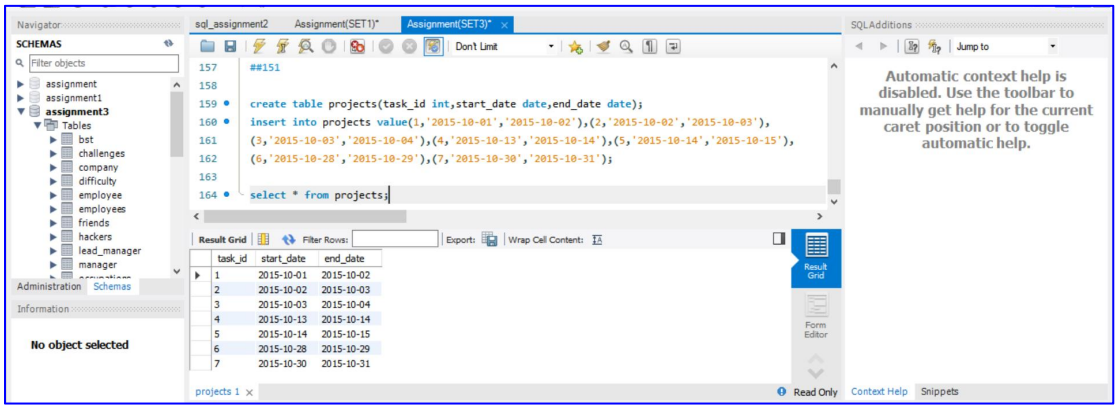



```
select s.hacker_id, name from submissions as s
join hackers as h on s.hacker_id = h.hacker_id
join challenges as c on s.challenge_id = c.challenge_id
join difficulty as d on c.difficulty_level = d.difficulty_level
where s.score = d.score
group by name, s.hacker_id
having count(s.challenge_id) > 1
order by count(s.challenge_id) desc, s.hacker_id;
```

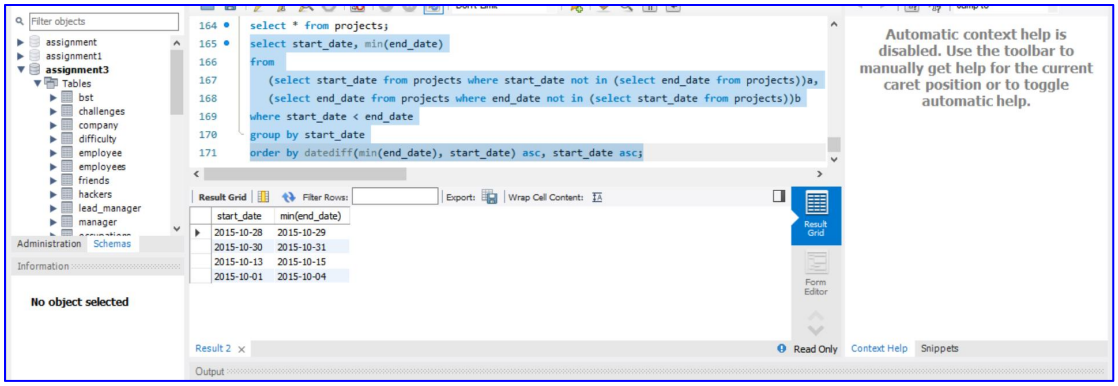


Q152) Write a query to output the start and end dates of projects listed by the number of days it took to complete the project in ascending order. If there is more than one project that have the same number of completion days, then order by the start date of the project.

Sol) At first create an empty table 'projects' and insert values in it using multi insert statement.

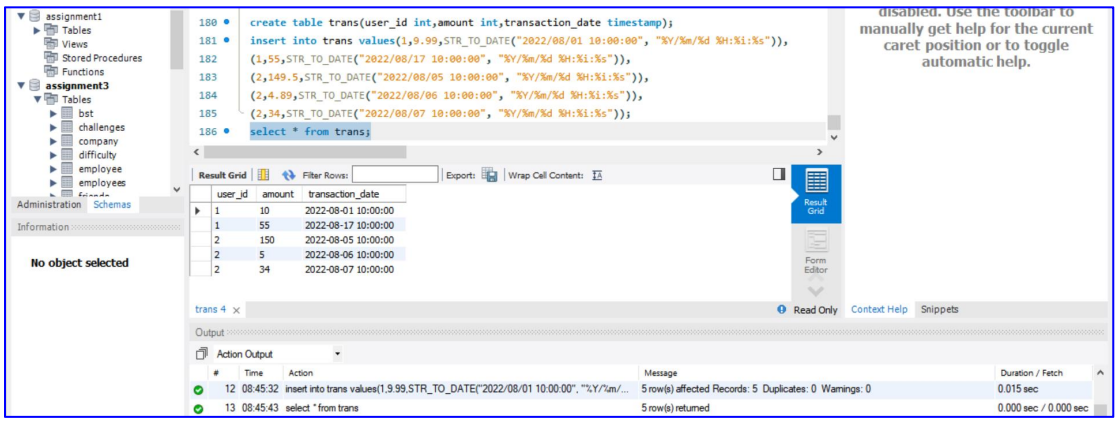



```
select start_date, min(end_date)
from
  (select start_date from projects where start_date not in
(select end_date from projects))a,
  (select end_date from projects where end_date not in
(select start_date from projects))b
where start_date < end_date
group by start_date
order by datediff(min(end_date), start_date) asc, start_date
asc;
```



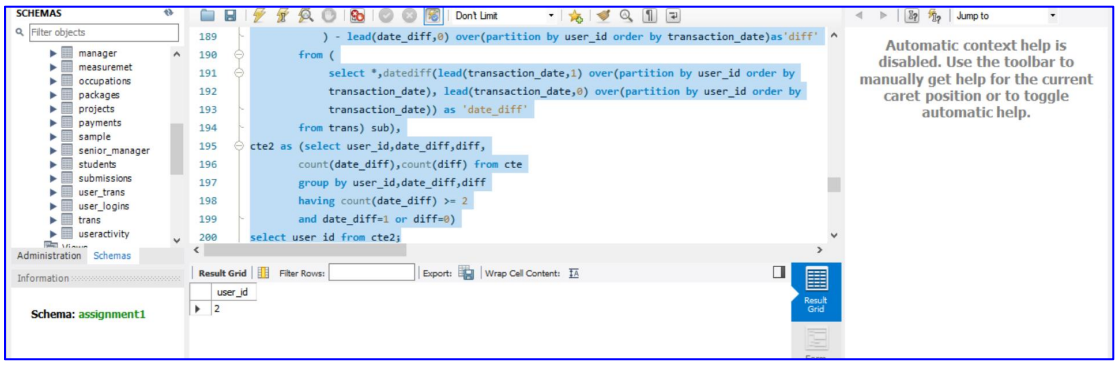
Q153) In an effort to identify high-value customers, Amazon asked for your help to obtain data about users who go on shopping sprees. A shopping spree occurs when a user makes purchases on 3 or more consecutive days. List the user IDs who have gone on at least 1 shopping spree in ascending order.

Sol) At first create an empty an table ‘trans’ and insert values in it.




```
with cte as (
        select *, lead(date_diff,1) over(partition by user_id order by
transaction_date
        ) - lead(date_diff,0) over(partition by user_id order by
transaction_date)as'diff'
                from (
         select *,datediff(lead(transaction_date,1) over(partition by
user_id order by
         transaction_date), lead(transaction_date,0) over(partition by
user_id order by
         transaction_date)) as 'date_diff'
                from trans) sub),
cte2 as (select user_id,date_diff,diff,
```

```
        count(date_diff),count(diff) from cte
    group by user_id,date_diff,diff
    having count(date_diff) >= 2
    and date_diff=1 or diff=0)
select user_id from cte2;
```

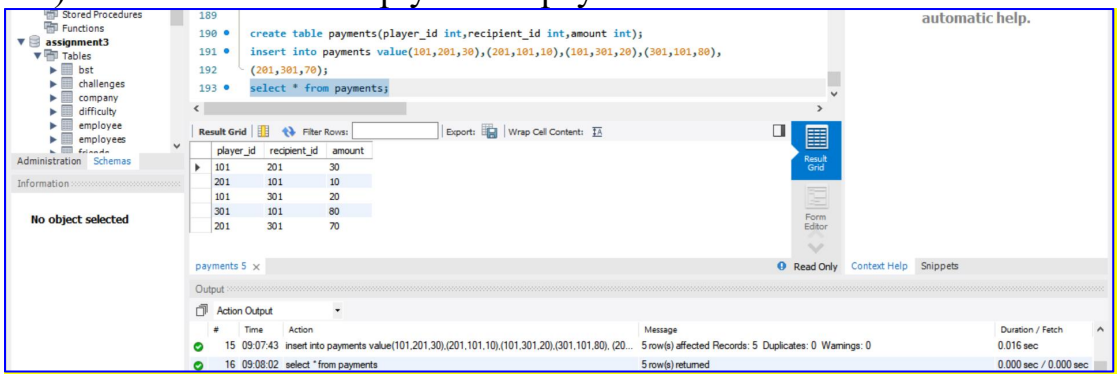


Q154) You are given a table of PayPal payments showing the payer, the recipient, and the amount paid. A two-way unique relationship is established when two people send money back and forth. Write a query to find the number of two-way unique relationships in this data.

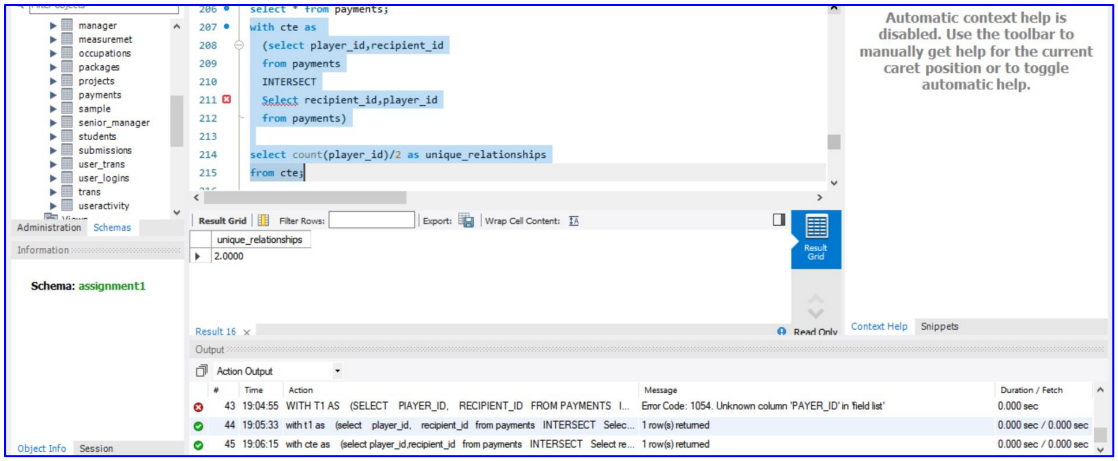
Assumption:

- **A payer can send money to the same recipient multiple times**

Sol) At first create an empty table ‘payments’ and insert values in it.




```
with cte as
  (select player_id, recipient_id from payments
  intersect
  Select recipient_id,player_id
  from payments)

Select count(payer_id)/2 as unique_relationships
From cte;
```

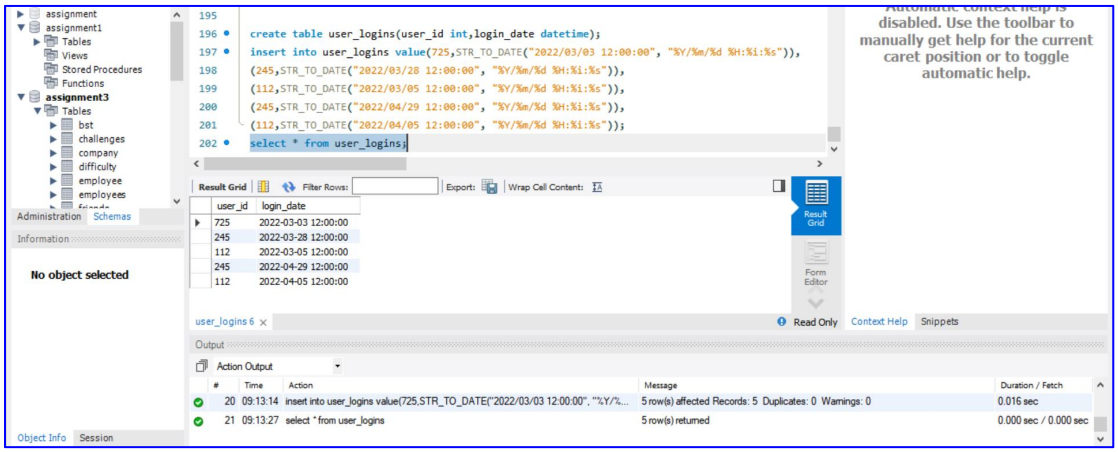


Q155)Assume you are given the table below containing information on Facebook user logins. Write a query to obtain the number of reactivated users (which are dormant users who did not log in the previous month, then logged in during the current month).
Output the current month (in numerical) and number of reactivated users. Assumption:

- **The rows in the user_logins table are complete for the year of 2022 and there are no missing**

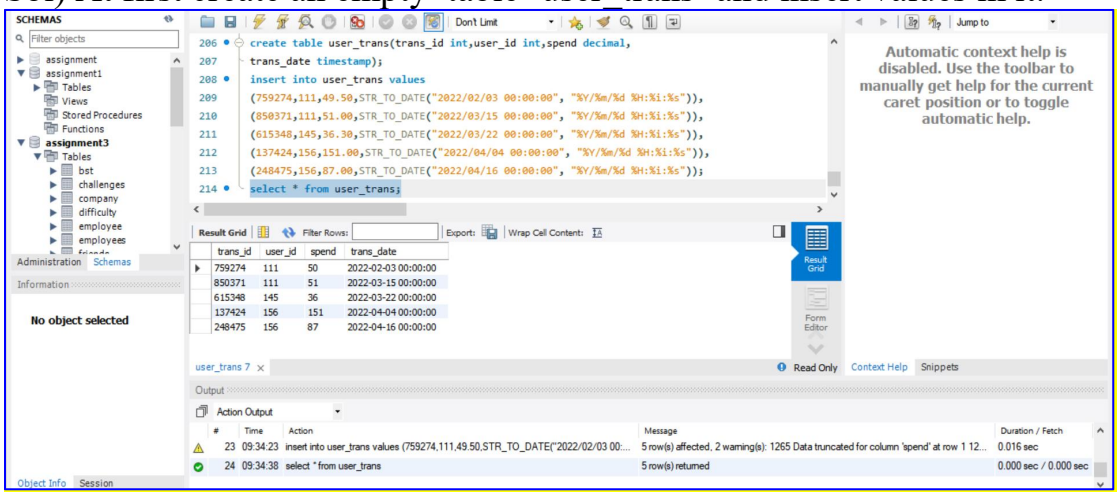
dates.

Sol) At first create an empty table ‘user_logins’ and insert values in it using multi insert statement.

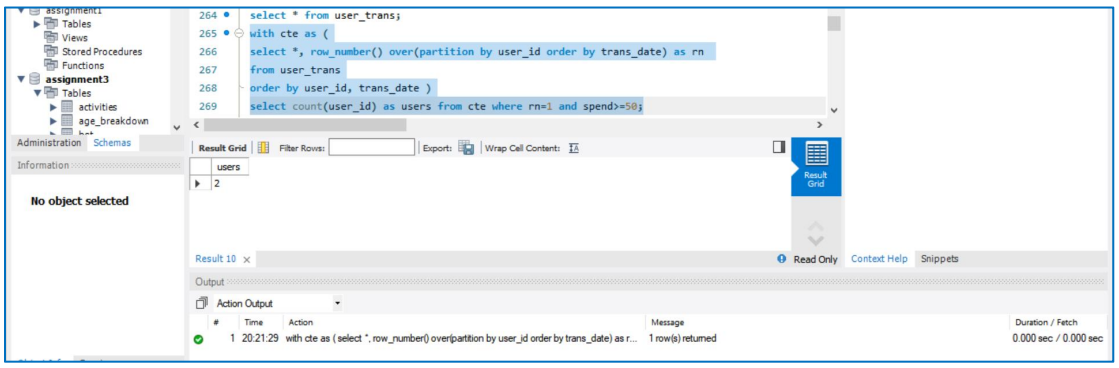


Q156) Assume you are given the table below on user transactions. Write a query to obtain the list of customers whose first transaction was valued at $50 or more. Output the number of users.
Clarification:
● Use the transaction_date field to determine which transaction should be labeled as the first
for each user.
● Use a specific function (we can't give too much away!) to account for scenarios where a user
had multiple transactions on the same day, and one of those was the first.

Sol) At first create an empty table 'user_trans' and insert values in it.

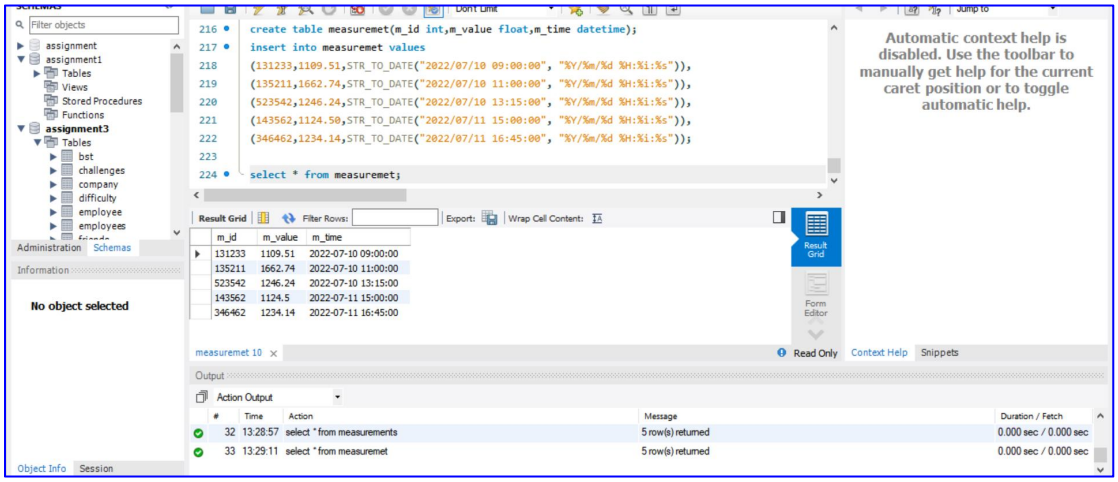



```
with cte as (
select *, row_number() over(partition by user_id order by trans_date) as rn
from user_trans
order by user_id, trans_date )
select count(user_id) as users from cte where rn=1 and spend>=50;
```

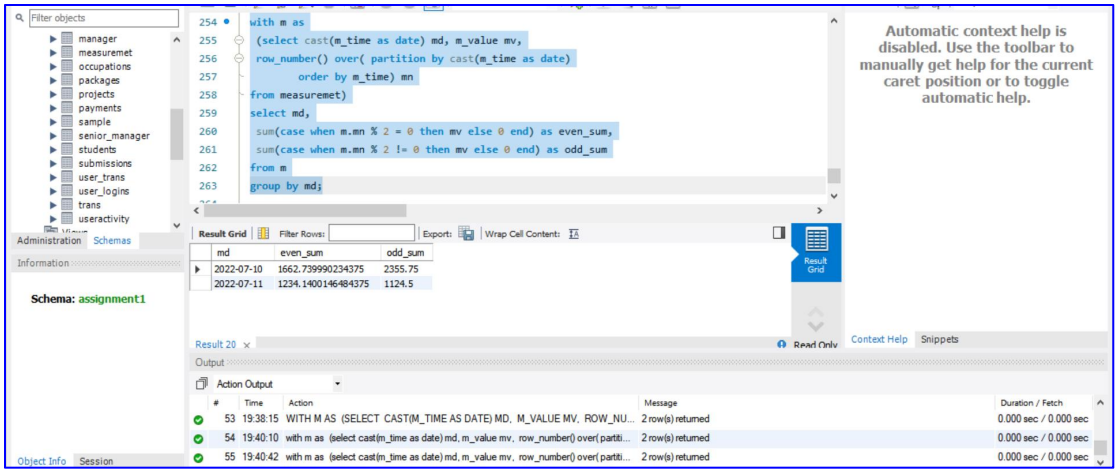


Q157) Assume you are given the table below containing measurement values obtained from a sensor over several days. Measurements are taken several times within a given day. Write a query to obtain the sum of the odd-numbered and even-numbered measurements on a particular day, in two different columns.
Note that the 1st, 3rd, 5th measurements within a day are considered odd-numbered measurements and the 2nd, 4th, 6th measurements are even-numbered measurements.

Sol) At first create an empty table 'Measurements' and insert values in it.

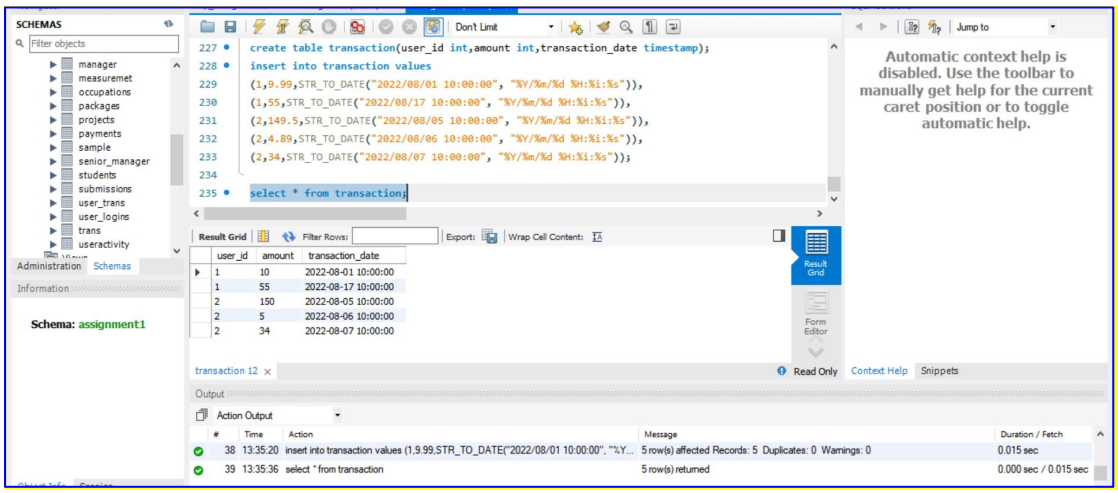



```
with m as
 (select cast(m_time as date) md, m_value mv,
 row_number() over( partition by cast(m_time as date)
       order by m_time) mn
from measuremet)
select md,sum(case when m.mn % 2 = 0 then mv else 0 end) as
even_sum,sum(case when m.mn % 2 != 0 then mv else 0 end) as
odd_sum from m
group by md;
```



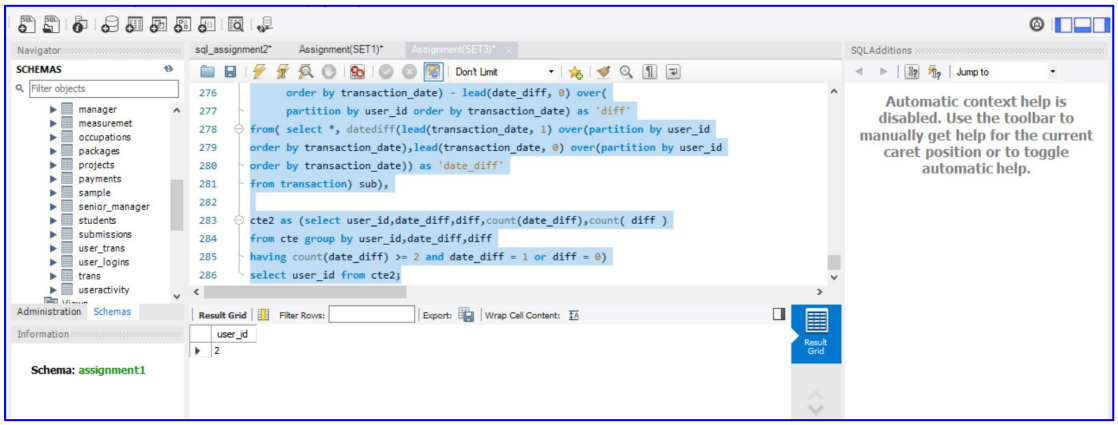
Q158) In an effort to identify high-value customers, Amazon asked for your help to obtain data about users who go on shopping sprees. A shopping spree occurs when a user makes purchases on 3 or more consecutive days. List the user IDs who have gone on at least 1 shopping spree in ascending order.

Sol) At first create an empty table ‘Transaction’ and insert values in it.




```
with cte as (
    select *,lead(date_diff, 1) over(partition by user_id
    order by transaction_date) - lead(date_diff, 0) over(
    partition by user_id order by transaction_date) as 'diff'
from( select *, datediff(lead(transaction_date, 1) over(partition by
user_id
order by transaction_date),lead(transaction_date, 0) over(partition
by user_id
order by transaction_date)) as 'date_diff'
from transaction) sub),
```

```
cte2 as (select user_id,date_diff,diff,count(date_diff),count( diff )
from cte group by user_id,date_diff,diff
having count(date_diff) >= 2 and date_diff = 1 or diff = 0)
select user_id from cte2;
```



Q159) The Airbnb Booking Recommendations team is trying to understand the "substitutability" of two rentals and whether one rental is a good substitute for another. They want you to write a query to find the unique combination of two Airbnb rentals with the same exact amenities offered. Output the count of the unique combination of Airbnb rentals.

Assumptions:

● If property 1 has a kitchen and pool, and property 2 has a kitchen and pool too, it is a good substitute and represents a unique matching rental.

● If property 3 has a kitchen, pool and fireplace, and property 4 only has a pool and fireplace, then it is not a good substitute.

Sol) At first create an empty 4 table‘Hackers’,’Difficulty’ ,’Challenges’& ‘Submissions’ and insert values in it.

Q160) Google marketing managers are analysing the performance of various advertising accounts over the last month. They need your help to gather the relevant data. Write a query to calculate the return on ad spend (ROAS) for each advertiser across all ad campaigns. Round your answer to 2 decimal places, and order your output by the advertiser_id.

Sol) At first create an empty 4 table‘Hackers’,’Difficulty’ ,’Challenges’& ‘Submissions’ and insert values in it.

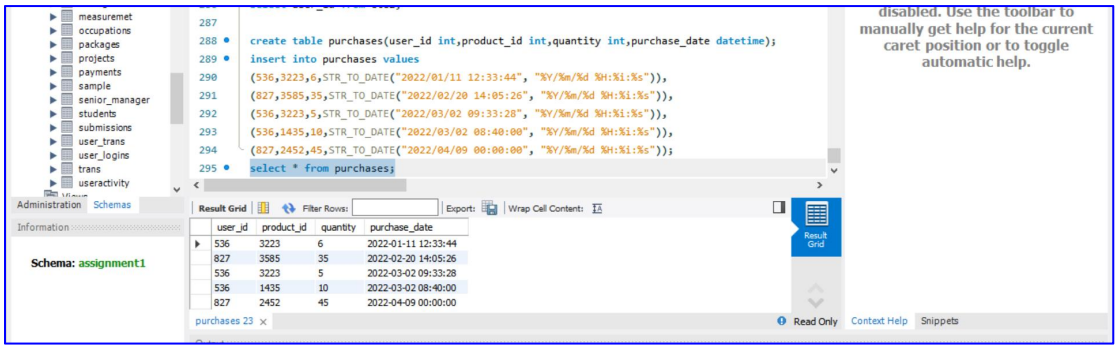
Q161) An employee is considered to be potentially overpaid if they earn more than 2 times the average salary for people with the same title. Similarly, an employee might be underpaid if they earn less than half of the average for their title. We'll refer to employees who are both underpaid and overpaid as compensation outliers for the purposes of this problem.
Write a query that shows the following data for each compensation outlier: employee ID, salary, and
whether they are potentially overpaid or potentially underpaid (refer to Example Output below).

Sol)

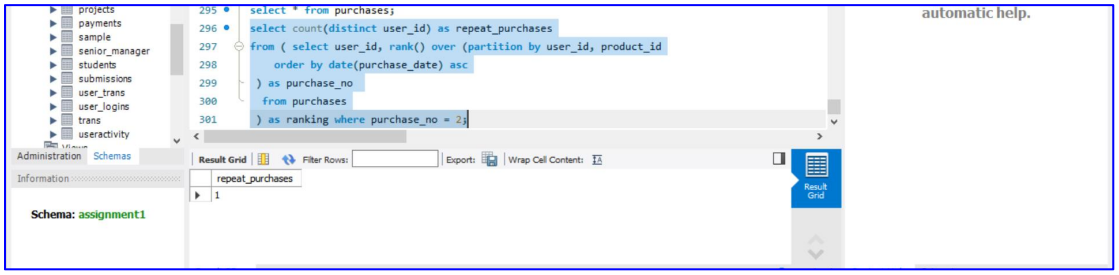
Q162) SAME AS Q154)

Q163) Assume you are given the table below containing information on user purchases. Write a query to obtain the number of users who purchased the same product on two or more different days. Output the number of unique users.
PS. On 26 Oct 2022, we expanded the purchases data set, thus the official output may vary from before.

Sol) At first create an empty table 'purchases' and insert values in it.

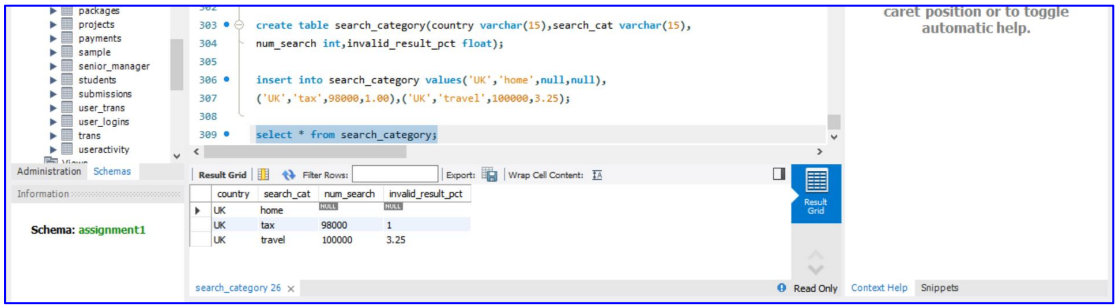



```
select count(distinct user_id) as repeat_purchases
from ( select user_id, rank() over (partition by user_id,
product_id
    order by date(purchase_date) asc
 ) as purchase_no
  from purchases
 ) as ranking where purchase_no = 2;
```

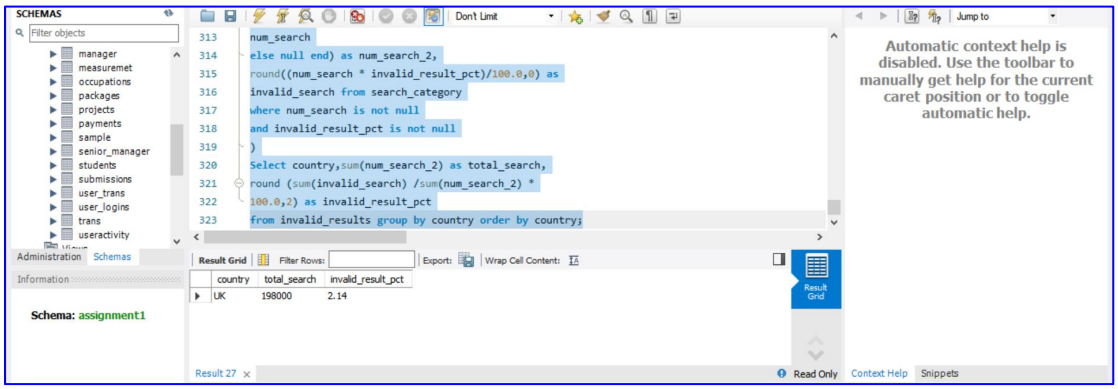


Q164) Assume you are given the table below containing the information on the searches attempted and the percentage of invalid searches by country. Write a query to obtain the percentage of invalid searches. Output the country in ascending order, total searches and overall percentage of invalid searches rounded to 2 decimal places.

Sol) At first create an empty table ‘Search_category’ and insert values in it.




```
with invalid_results
as (select country,num_search,invalid_result_pct,
(case when invalid_result_pct is not null then
num_search
else null end) as num_search_2,
round((num_search * invalid_result_pct)/100.0,0) as
invalid_search from search_category
where num_search is not null
and invalid_result_pct is not null
)
Select country,sum(num_search_2) as total_search,
round (sum(invalid_search) /sum(num_search_2) *
100.0,2) as invalid_result_pct
from invalid_results group by country order by country;
```

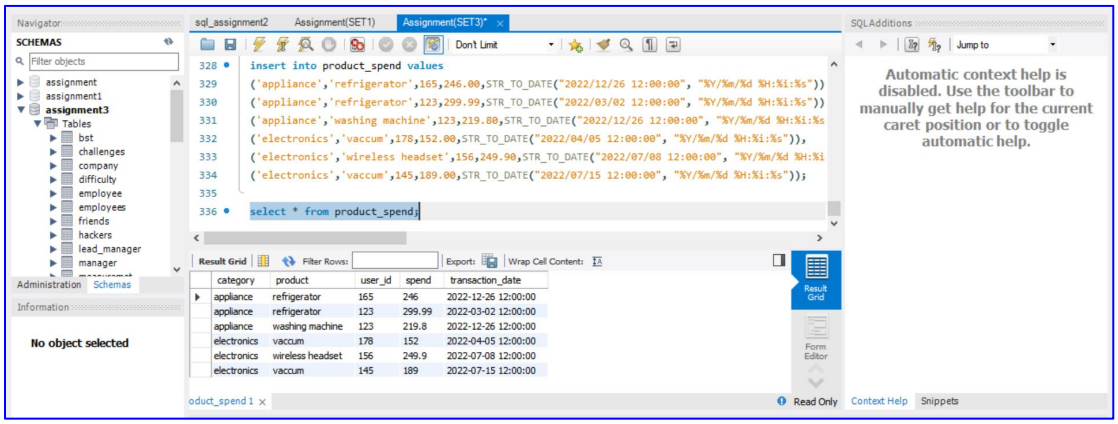


Q165) Say you have access to all the transactions for a given merchant account. Write a query to print the cumulative balance of the merchant account at the end of each day, with the total balance reset back to zero at the end of the month. Output the transaction date and cumulative balance.
Hint-You should use CASE.

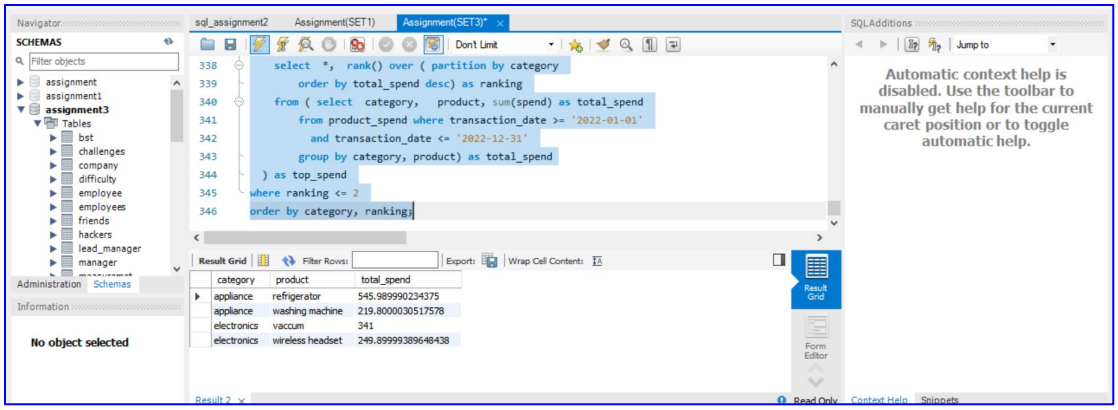
Sol)

Q166) Assume you are given the table below containing information on Amazon customers and their spend on products belonging to various categories. Identify the top two highest-grossing products within each category in 2022. Output the category, product, and total spend.

Sol) At first create an empty 4 table 'product_spend and insert values in it.




```
select category, product, total_spend from (
   select  *,  rank() over ( partition by category
      order by total_spend desc) as ranking
   from ( select  category,   product, sum(spend) as total_spend
      from product_spend where transaction_date >= '2022-01-01'
        and transaction_date <= '2022-12-31'
      group by category, product) as total_spend
  ) as top_spend
where ranking <= 2
order by category, ranking;
```

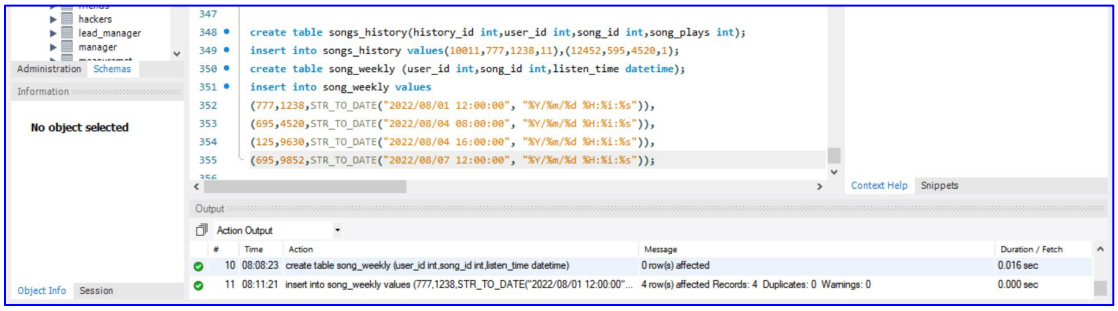


Q167) Facebook is analysing its user signup data for June 2022.
Write a query to generate the churn rate by week in June 2022.
Output the week number (1, 2, 3, 4, ...) and the corresponding churn rate rounded to 2 decimal places.
For example, week number 1 represents the dates from 30 May to 5 Jun, and week 2 is from 6 Jun to 12 Jun.
Hint- Use Extract.
Assumptions:
● If the last_login date is within 28 days of the signup_date, the user can be considered
churned.
● If the last_login is more than 28 days after the signup date, the user didn't churn.
users Table:

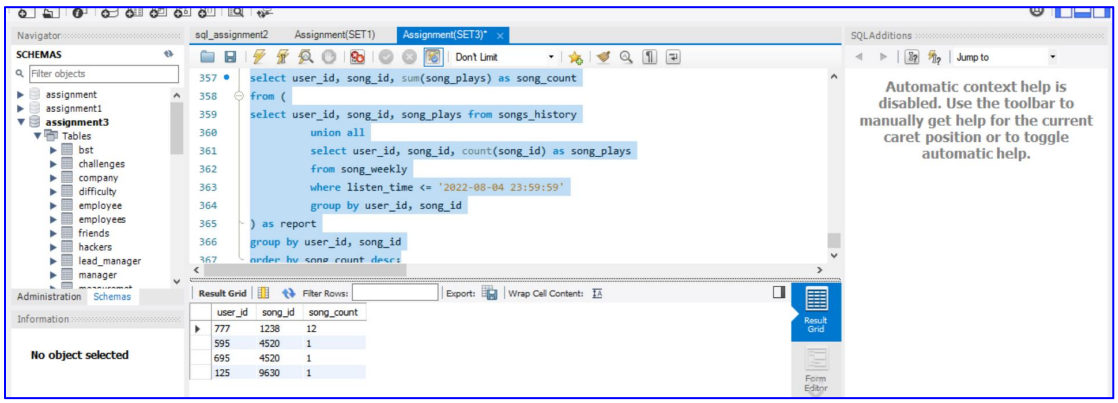
Sol)

Q168) You're given two tables on Spotify users' streaming data. songs_history table contains the historical streaming data and songs_weekly table contains the current week's streaming data. Write a query to output the user id, song id, and cumulative count of song plays as of 4 August 2022 sorted in descending order.
Hint- Use group by

Sol) At first create 2 empty tables 'Songs_weekly ' & 'songs_history' and insert values in it.

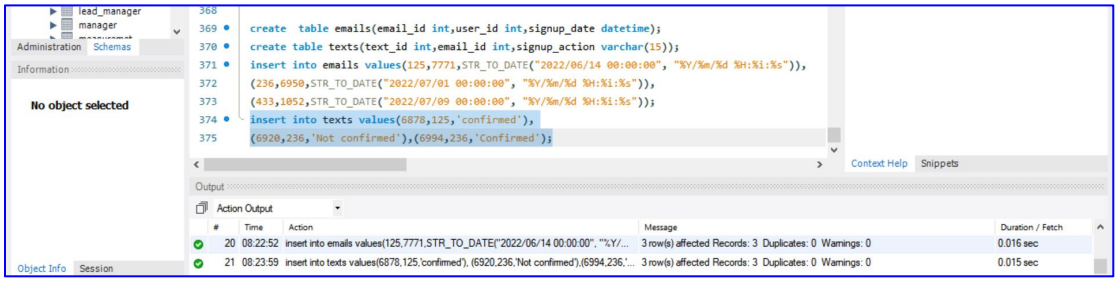



```
select user_id, song_id, sum(song_plays) as song_count
from (
select user_id, song_id, song_plays from songs_history
        union all
        select user_id, song_id, count(song_id) as song_plays
        from song_weekly
        where listen_time <= '2022-08-04 23:59:59'
        group by user_id, song_id
) as report
group by user_id, song_id
order by song_count desc;
```

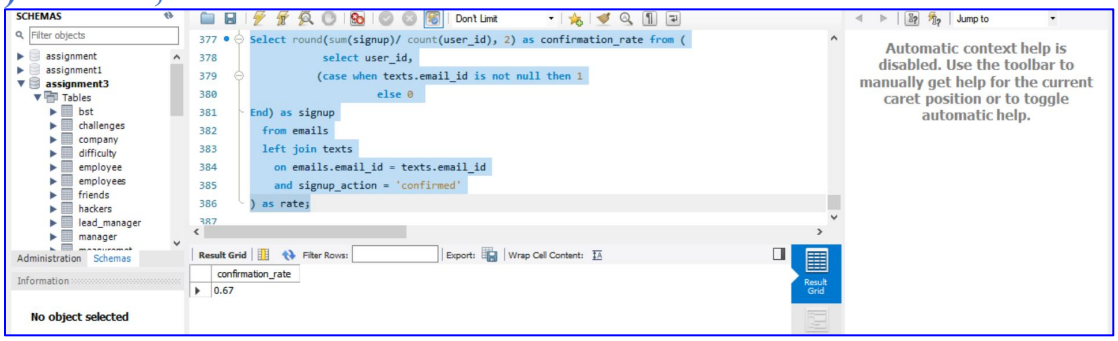


Q169) New TikTok users sign up with their emails, so each signup requires a text confirmation to activate the new user's account. Write a query to find the confirmation rate of users who confirmed their signups with text messages. Round the result to 2 decimal places.
Hint- Use Joins

Sol) At first create 2 empty tables ‘Emails’ & ’Texts’ and insert values in it.




```
Select round(sum(signup) / count(user_id), 2) as
confirmation_rate from (
          select user_id,
         (case when texts.email_id is not null then 1
                    else 0
              End) as signup
  from emails
  left join texts
    on emails.email_id = texts.email_id
    and signup_action = 'confirmed'
) as rate;
```



Q170) The table below contains information about tweets over a given period of time. Calculate the 3-day rolling average of tweets published by each user for each date that a tweet was posted. Output the user id, tweet date, and rolling averages rounded to 2 decimal places.

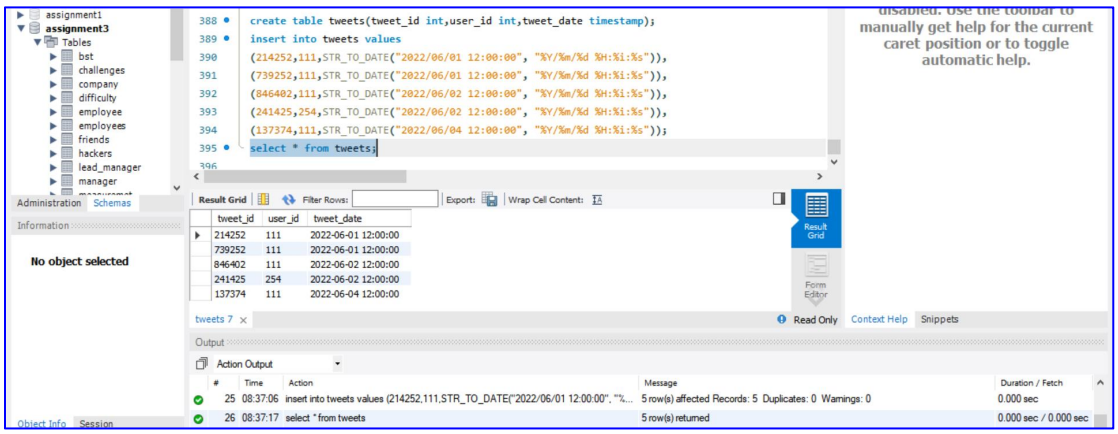
Hint- Use Count and group by

Important Assumptions:

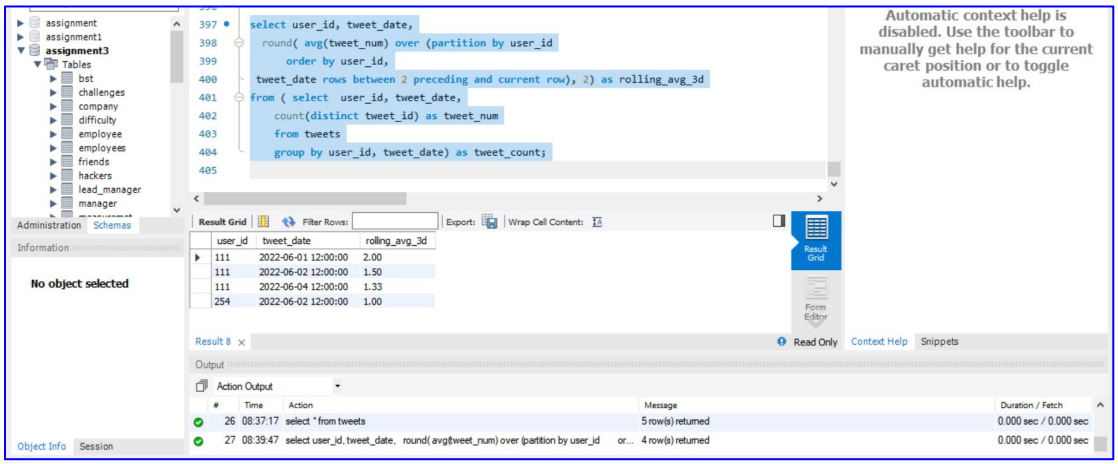
- **Rows in this table are *consecutive* and ordered by date.**
- **Each row represents a different day**
- **A day that does not correspond to a row in this table is not counted.**

The most recent day is the next row above the current row

Sol) At first create an empty table ‘Tweets’ and insert values in it.



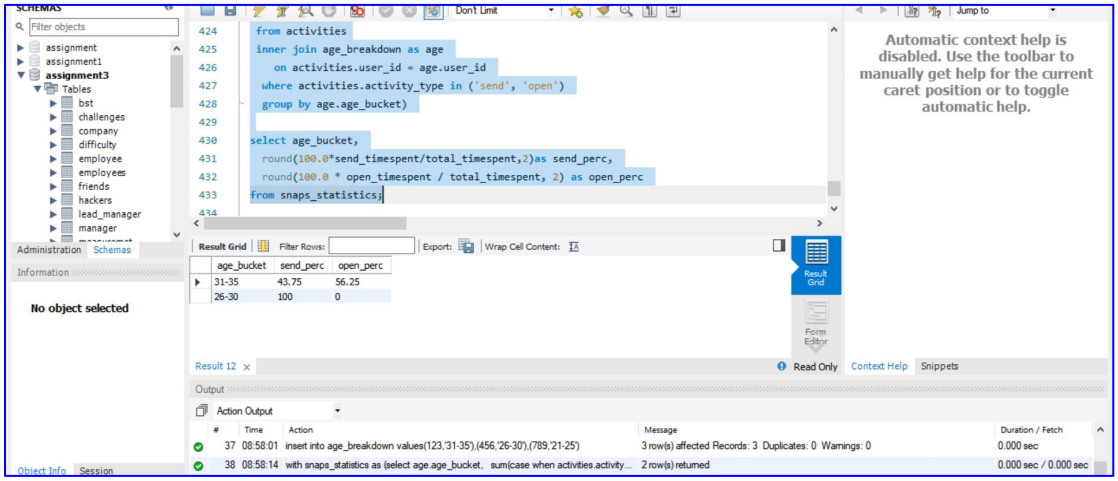

```
select user_id, tweet_date,
  round( avg(tweet_num) over (partition by user_id
      order by user_id,
 tweet_date rows between 2 preceding and current row), 2)
as rolling_avg_3d
from ( select  user_id, tweet_date,
    count(distinct tweet_id) as tweet_num
    from tweets
    group by user_id, tweet_date) as tweet_count;
```



Q171) Assume you are given the tables below containing information on Snapchat users, their ages, and their time spent sending and opening snaps. Write a query to obtain a breakdown of the time spent sending vs. opening snaps (as a percentage of total time spent on these activities) for each age group.
Hint- Use join and case Output the age bucket and percentage of sending and opening snaps. Round the percentage to 2 decimal places.

Sol) At first create 2 empty tables 'Activities ' & 'Age_breakdown' and insert values in it.

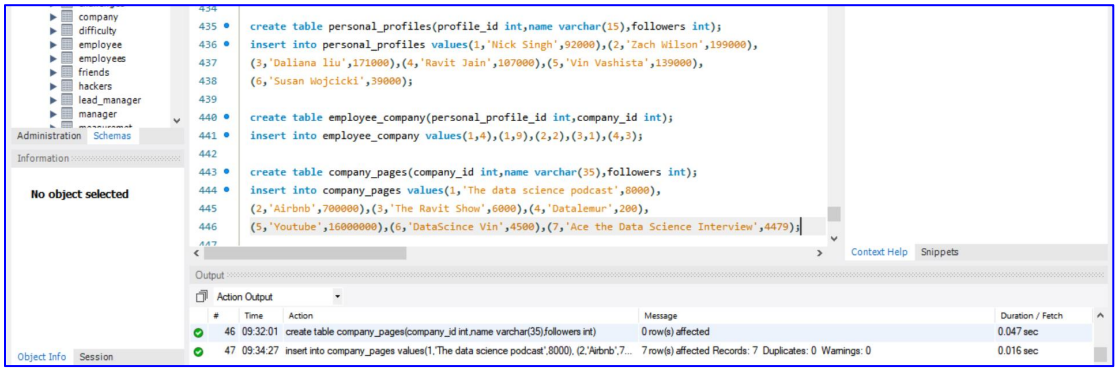

```
with snaps_statistics as (select age.age_bucket,
 sum(case when activities.activity_type = 'send'
        then activities.time_spent else 0 end) as send_timespent,
 sum(case when activities.activity_type = 'open'
     then activities.time_spent else 0 end) as open_timespent,
 sum(activities.time_spent) as total_timespent
 from activities
 inner join age_breakdown as age
   on activities.user_id = age.user_id
  where activities.activity_type in ('send', 'open')
  group by age.age_bucket)

select age_bucket,
  round(100.0*send_timespent/total_timespent,2)as
send_perc,
  round(100.0 * open_timespent / total_timespent, 2) as
open_perc
from snaps_statistics;
```



Q172) The LinkedIn Creator team is looking for power creators who use their personal profile as a company or influencer page. This means that if someone's Linkedin page has more followers than all the companies they work for, we can safely assume that person is a Power Creator. Keep in mind that if a person works at multiple companies, we should take into account the company with the most followers.

Write a query to return the IDs of these LinkedIn power creators in ascending order

Sol) At first create 3 empty tables 'personal_profile','employee_company' & 'company_pages' and insert values in it using multi insert command.




```
select distinct p.profile_id from personal_profiles p inner
join employee_company e
on p.profile_id=e.personal_profile_id inner join
company_pages c
on e.company_id=c.company_id
where p.followers>c.followers and p.profile_id not in
```

```
(    select p.profile_id from personal_profiles p inner join
employee_company e
     on p.profile_id=e.personal_profile_id
     inner join company_pages c on
e.company_id=c.company_id
     where p.followers<c.followers)
   order by p.profile_id;
```