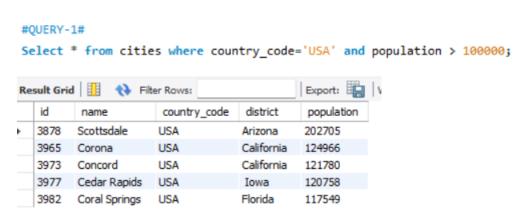
SQL CHALLENGE

[Suru Shashank]

Q1. Query all columns for all American cities in the CITY table with populations larger than 100000. The CountryCode for America is USA.

The CITY table is described as follows:

ANS:



Q2. Query the NAME field for all American cities in the CITY table with populations larger than 120000. The CountryCode for America is USA.

The CITY table is described as follows:



Q3. Query all columns (attributes) for every row in the CITY table.

The CITY table is described as follows:

ANS:

31 #QUERY-3#
32 • select * from cities;



Q4. Query all columns for a city in CITY with the ID 1661.

The CITY table is described as follows:

ANS:

```
#QUERY-4#
select * from cities where id='1661'
```

Q5. Query all attributes of every Japanese city in the CITY table. The COUNTRYCODE for Japan is JPN.

ANS:

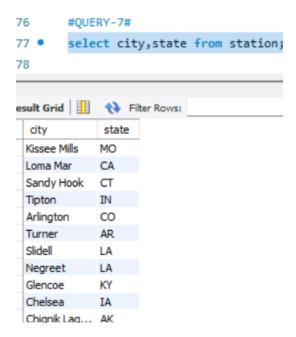
```
#QUERY-5#
select * from cities where country_code='JPN'
```

Q6. Query the names of all the Japanese cities in the CITY table. The COUNTRYCODE for Japan is JPN.

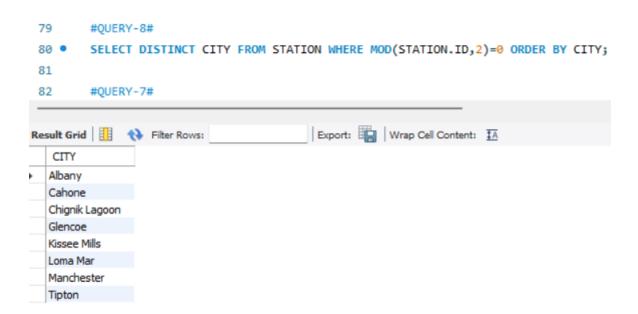
```
#QUERY-6#
select name from cities where country_code='JPN'
```

Q7. Query a list of CITY and STATE from the STATION table.

ANS:



Q8. Query a list of CITY names from STATION for cities that have an even ID number. Print the results in any order, but exclude duplicates from the answer.



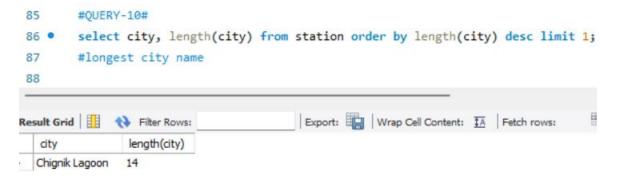
Q9. Find the difference between the total number of CITY entries in the table and the number of distinct CITY entries in the table.

ANS:

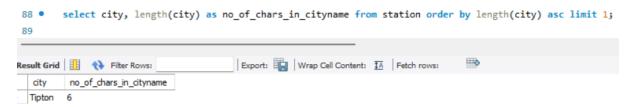
Q10. Query the two cities in STATION with the shortest and longest CITY names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically.

ANS:

Longest characters:



Shortest characters:



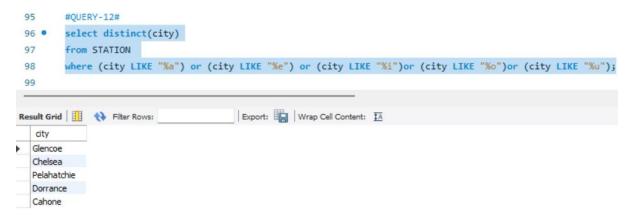
Q11. Query the list of CITY names starting with vowels (i.e., a, e, i, o, or u) from STATION. Your result cannot contain duplicates.

ANS:



Q12. Query the list of CITY names ending with vowels (a, e, i, o, u) from STATION. Your result cannot contain duplicates.

ANS:

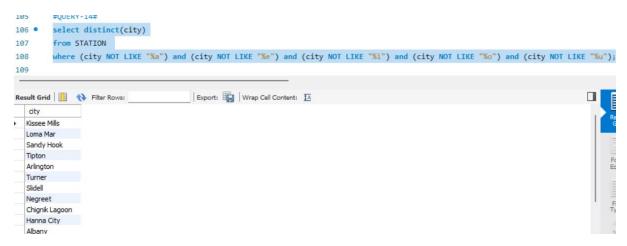


Q13. Query the list of CITY names from STATION that do not start with vowels. Your result cannot contain duplicates.

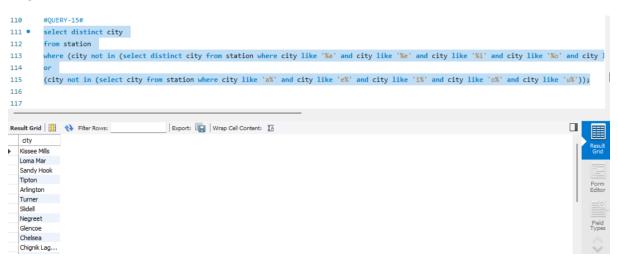
```
101 • select distinct(city)
        from STATION
        where (city NOT LIKE "a%") and (city NOT LIKE "e%") and (city NOT LIKE "i%") and (city NOT LIKE "o%") and (city NOT LIKE "u%");
103
Export: Wrap Cell Content: IA
  city
 Kissee Mills
  Loma Mar
  Sandy Hook
  Tipton
  Turner
  Slidell
  Negreet
  Glencoe
  Chignik Lagoon
```

Q14. Query the list of CITY names from STATION that do not end with vowels. Your result cannot contain duplicates.

ANS:



Q16. Query the list of CITY names from STATION that do not start with vowels and do not end with vowels. Your result cannot contain duplicates.



Q15. Query the list of CITY names from STATION that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates.

ANS:

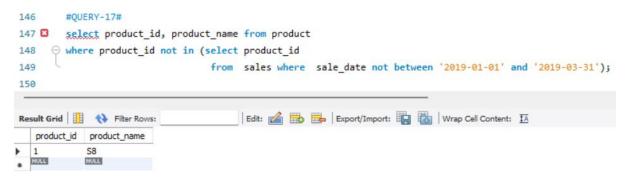
```
117
        #0UERY-16#
118 •
       select distinct city
       where (city not in (select distinct city from station where city like '%a' or city like '%e' or city like '%i' or city like '%o' or city like
120
121
122
        (city not in (select city from station where city like 'a%' or city like 'e%' or city like 'i%' or city like 'o%' or city like 'u%'));
124
125
Export: Wrap Cell Content: 1A
   city
Kissee Mills
  Sandy Hook
  Tipton
  Arlington
  Turner
  Negreet
  Glencoe
```

17)

Write an SQL query that reports the products that were only sold in the first quarter of 2019. That is, between 2019-01-01 and 2019-03-31 inclusive.

Return the result table in any order.

ANS:



18)

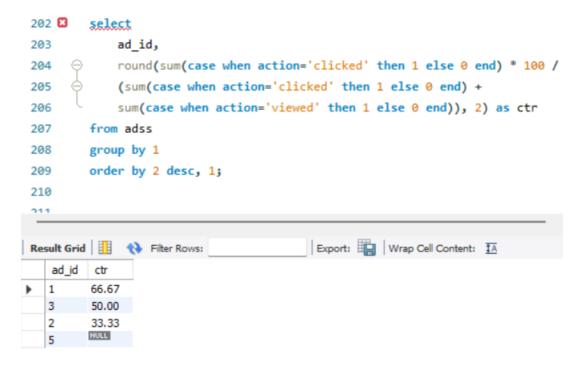
Write an SQL query to find all the authors that viewed at least one of their own articles. Return the result table sorted by id in ascending order.

Write an SQL query to find the percentage of immediate orders in the table, rounded to 2 decimal places.

Write an SQL query to find the ctr of each Ad. Round ctr to two decimal points.

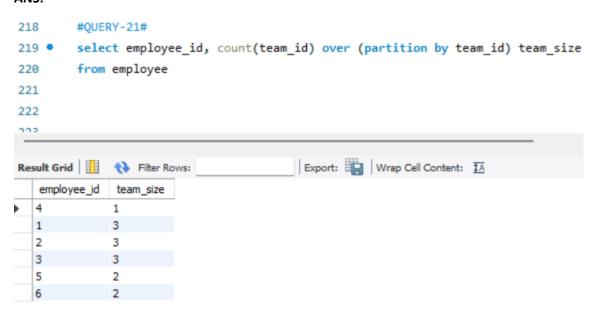
Return the result table ordered by ctr in descending order and by ad_id in ascending order in case of a

ANS:



21)

Write an SQL query to find the team size of each of the employees.



Write an SQL query to find the type of weather in each country for November 2019. The type of weather is:

- Cold if the average weather_state is less than or equal 15,
- Hot if the average weather_state is greater than or equal to 25, and
- Warm otherwise.

ANS:

```
251
         #QUERY-22#
252
         select c.country_name,
253
                case
254
                     when avg(w.weather_state) <= 15.0 then 'Cold'
                    when avg(w.weather_state) >= 25.0 then 'Hot'
255
256
                     else 'Warm'
257
                end as weather type
         from countries as c
258
         inner join weather as w
259
260
         on c.country id = w.country id
        where w.day between '2019-11-01' and '2019-11-30'
261
262
         group by c.country_id;
                                           Export: Wrap Cell Content:
Result Grid
              Filter Rows:
   country_name
                weather_type
  USA
                Cold
  Australia
               Cold
  China
               Warm
  Peru
               Hot
  Morocco
               Hot
```

23)

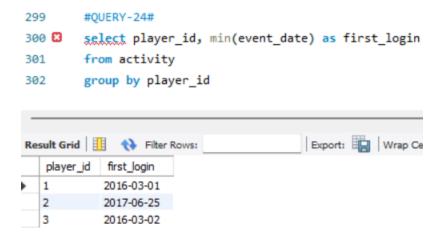
Write an SQL query to find the average selling price for each product. average_price should be rounded to 2 decimal places.

Return the result table in any order.

```
#QUERY-23#
282
283
        select units_sold.product_id, round(sum(units*price)/sum(units), 2) as average_price
        from units_sold inner join prices
284
        on units_sold.product_id = prices.product_id
        and units_sold.purchase_date between prices.start_date and prices.end_date
286
287
        group by units_sold.product_id
                                      Export: Wrap Cell Content: IA
product id
           average price
            6.96
  1
            16.96
```

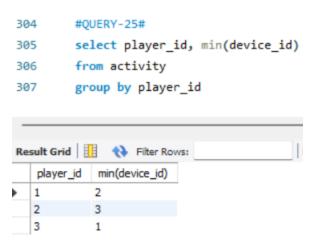
24) Write an SQL query to report the first login date for each player.

ANS:



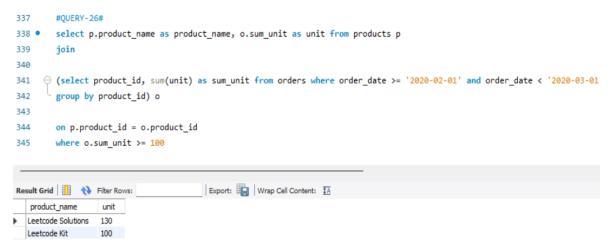
25)

Write an SQL query to report the device that is first logged in for each player.



Write an SQL query to get the names of products that have at least 100 units ordered in February 2020 and their amount.

ANS:



27)

Write an SQL query to find the users who have valid emails.

A valid e-mail has a prefix name and a domain where:

- The prefix name is a string that may contain letters (upper or lower case), digits, underscore
 '_', period '.', and/or dash '-'. The prefix name must start with a letter.
- The domain is '@leetcode.com'.

ANS:

```
#QUERY-27#
SELECT *
FROM Users
WHERE REGEXP_LIKE(mail, '^[a-zA-Z][a-zA-Z0-9\_\.\-]*@leetcode.com')
28)
```

Write an SQL query to report the customer_id and customer_name of customers who have spent at least \$100 in each month of June and July 2020.

```
#QUERY-28#
select o.customer_id, c.name
from customers as c, product p, orders o
where c.customer_id = o.customer_id and p.product_id = o.product_id
group by o.customer_id
having

(
    sum(case when o.order_date like '2020-06%' then o.quantity*p.price else 0 end) >= 100
    and
    sum(case when o.order_date like '2020-07%' then o.quantity*p.price else 0 end) >= 100
)
```

Write an SQL query to report the distinct titles of the kid-friendly movies streamed in June 2020. Return the result table in any order.

ANS:

```
#QUERY-30#

366     select distinct title

367     from content

368     join tv_program using(content_id)

369     where kids_content = 'Y'

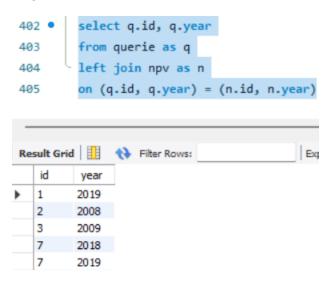
370          and content_type = 'Movies'

371          and (month(program_date)=6 and year(program_date) = 2020

30)
```

Write an SQL query to find the npv of each query of the Queries table.

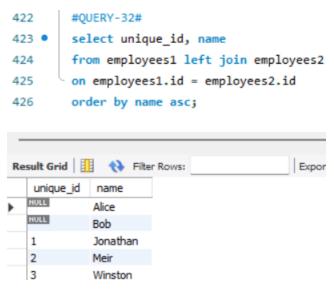
ANS:



31) REPEATED previously

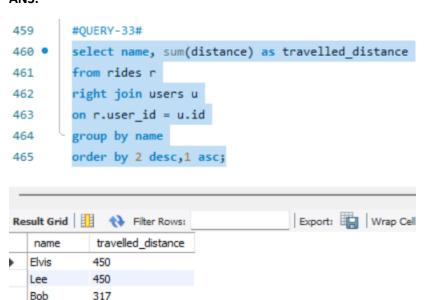
32)

Write an SQL query to show the unique ID of each user, If a user does not have a unique ID replace just show null.



Write an SQL query to report the distance travelled by each user.

ANS:



34)

REPEATED PREVIOUSLY

312

222

Jonathan

Alex

Write an SQL query to:

- Find the name of the user who has rated the greatest number of movies. In case of a tie, return the lexicographically smaller user name.
- Find the movie name with the highest average rating in February 2020. In case of a tie, return
 the lexicographically smaller movie name.

ANS:

```
from movie_rating natural join users1
  group by users1.user_id
  order by count(*) desc, name asc
  limit 1

)
union
(
  select movies.title results
  from movie_rating natural join movies
  where month(created_at)='2'
  group by movies.movie_id
  order by avg(rating) desc,title asc
);
```

36) REPEATED PREVIOUSLY

37) REPEATED PREVIOUSLY

38)

....

Write an SQL query to find the id and the name of all students who are enrolled in departments that no longer exist.

Return the result table in any order.

```
529
       #QUERY-38#
530 🖾
       select id, name
531
       from students
532
       where department_id not in (select id from departments)
Edit: 🕍 🖶 Export/Im
       name
  2
       John
  3
      Steve
       Jasmine
      Daiana
NULL
```

Write an SQL query to report the number of calls and the total call duration between each pair of distinct persons (person1, person2) where person1 < person2.

Return the result table in any order.

ANS:

```
select from_id as person1, to_id as person2,
          count(duration) as call_count, sum(duration) as total_duration
9
   0
      union all
1
      select to_id, from_id, duration
2
    from calls) as t1
3
      where from id < to id
4
5
      group by person1, person2
7 🖾
      drop table calls;
8
      select * from calls;
                                     Export: Wrap Cell Content: IA
sult Grid 🔢 🔷 Filter Rows:
person1
        person2
                call_count
                         total_duration
        2
                2
                         70
1
        3
                1
                         20
```

40)

3

REPEATED AGAIN-Q23

4

999

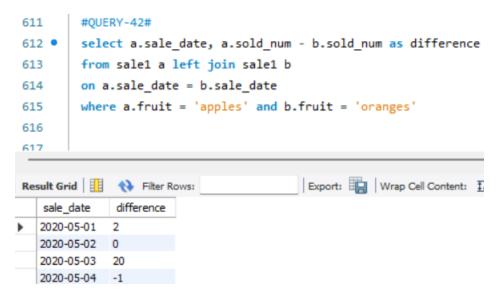
41)

Write an SQL query to report the number of cubic feet of volume the inventory occupies in each warehouse.

```
select warehouse_name, sum(volume) as volume from (
    select w.name as warehouse_name
```

Write an SQL query to report the difference between the number of apples and oranges sold each day. Return the result table ordered by sale_date.

ANS:



43) pending

44)

Write an SQL query to report the managers with at least five direct reports. Return the result table in any order.

ANS:

```
select e2.name
from employee2 e1
inner join employee2 e2 on e1.managerid = e2.id
group by e1.managerid
having count(e1.id) >= 5;

Result Grid

Result Grid

Name
John
```

45)

Write an SQL query to report the respective department name and number of students majoring in each department for all departments in the Department table (even ones with no current students). Return the result table ordered by student_number in descending order. In case of a tie, order them by

```
select
673
674
             a.dept_name,
675
             count(student_id) student_count
676
         from
677
             department2 a
678
         left join
             student2 b
679
680
         on
             (a.dept_id = b.dept_id)
681
682
         group by a.dept_name
         order by student_count desc, a.dept_name asc;
683
              Filter Rows:
Result Grid
                                           Export: Wrap
   dept_name
              student_count
  Engineering
             2
  Science
             1
  Law
             0
```

Write an SQL query to report the customer ids from the Customer table that bought all the products in the Product table.

ANS:

```
#QUERY-46#
select a.customer_id from

(select customer_id, count(distinct product_key) as num
from customer4
group by customer_id) a
where a.num = (select count(distinct product_key) from product4);
```

47)

Write an SQL query that reports the most experienced employees in each project. In case of a tie, report all employees with the maximum number of experience years.

```
select project_id, project47.employee_id
from project47 inner join Employee
on project47.employee_id = employee47.employee_id
where (project_id, experience_years) in
ANS: (select project_id, max(experience_years))
```

Write an SQL query that reports the books that have sold less than 10 copies in the last year, excluding books that have been available for less than one month from today. Assume today is 2019-06-23.

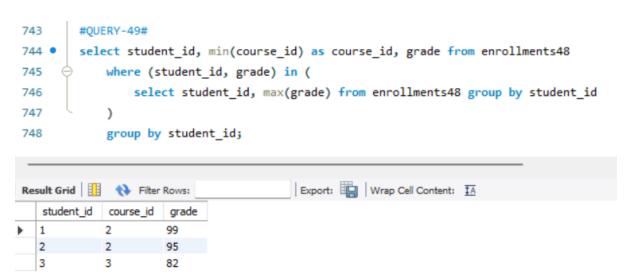
48)

```
#QUERY-48#
select book_id, name
from Books
where book_id not in (
    select book_id
    from Orders
    where dispatch_date >= '2018-06-23' and dispatch_date <= '2019-06-22'
    group by book_id
    having sum(quantity) >= 10)
and available_from < '2019-05-23'</pre>
```

49)

Write a SQL query to find the highest grade with its corresponding course for each student. In case of a tie, you should find the course with the smallest course_id.

ANS:

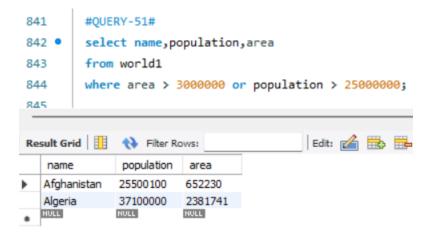


50) QUESTION is wrong because the given input tables to create say "teams" and matches but the expected output shows "players". Confusing...

Write an SQL query to report the name, population, and area of the big countries.

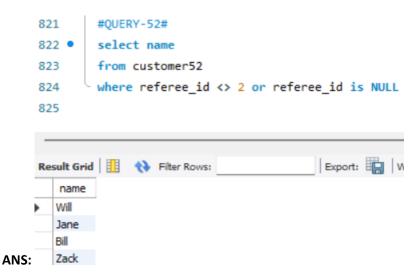
51) Return the result table in any order.

ANS:



52)

Write an SQL query to report the names of the customer that are not referred by the customer with id = 2.



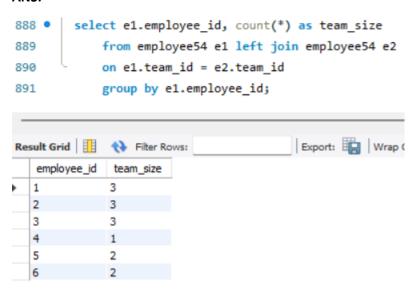
Write an SQL query to report all customers who never order anything.

53) Return the result table in any order.

```
select c.name as customers
from customers53 as c
where c.id not in (
select o.customerId from orders53 as o
);
```

Write an SQL query to find the team size of each of the employees. Return result table in any order.

ANS:



55)

Write an SQL query to find the countries where this company can invest.

```
1059
         #QUERY-55#
1060 •
         select c.name as country
         from person55 p
1061
         inner join country55 c
1062
         on left (p.phone number,3) = c.country code
1063
1064
      inner join (select caller id as id, duration
                     from calls55
1065
1066
         union
1067
         select callee_id as id, duration
                     from calls55) phn
1068
1069
         on p.id = phn.id
         group by country
1070
         having avg(duration) > (select avg(duration) from calls55);
1071
```

ANS: REPEATED PREVIOUSLY→ANS->25

57)

Write an SQL query to find the customer_number for the customer who has placed the largest number of orders.

The test cases are generated so that exactly one customer will have placed more orders than any other customer.

ANS:

Write an SQL query to report all the consecutive available seats in the cinema.

58) Return the result table ordered by seat_id in ascending order.

```
#QUERY-58#

SELECT c1.seat_id

FROM Cinema c1,

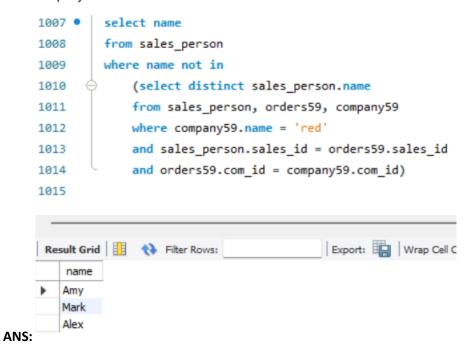
Cinema c2

WHERE ( ( c1.seat_id = c2.seat_id + 1 )

OR ( c1.seat_id = c2.seat_id - 1 ) )

ANS:
```

Write an SQL query to report the names of all the salespersons who did not have any orders related to the company with the name "RED".



60)

Write an SQL query to report for every three line segments whether they can form a triangle. Return the result table in any order.

```
928
         #QUERY-60#
929 •
         select x, y, z,
930
             case when x + y > z and y + z > x then 'Yes'
             when y + z > x and z + x > y then 'No'
931
             else 'wrong'
932
             end as triangle
933
         from triangle;
934
935
936
                                           Export: Wrap Ce
Result Grid
              Filter Rows:
                     triangle
   х
         у
               Z
  10
        20
               15
                     Yes
        15
  13
               30
                     No
```

Write an SQL query to report the shortest distance between any two points from the Point table.

ANS:

```
select min(p2.x - p1.x) as least
from point as p1 inner join point as p2
on p1.x <> p2.x;
```

62)

Write a SQL query for a report that provides the pairs (actor_id, director_id) where the actor has cooperated with the director at least three times.

```
#QUERY-62#
select actor_id, director_id from actordirector
group by actor_id, director_id
having count(actor_id) >= 3;

Result Grid  Filter Rows:

| Export: | Wrap C
```

Write an SQL query that reports the product_name, year, and price for each sale_id in the Sales table.

ANS:

```
#QUERY-63#
select product_name, year, price from sales63
left join product63
on sales63.product_id = product63.product_id;
```

64)

Write an SQL query that reports the average experience years of all the employees for each project, rounded to 2 digits.

```
#QUERY-64#
select project_id, round(avg(experience_years), 2) as avg_years
from project64 inner join employee64
on project64.employee_id = employee64.employee_id
group by project_id
65)
```

Write an SQL query that reports the best seller by total sales price, If there is a tie, report them all.

ANS:

```
select a.seller_id

from

(select seller_id, sum(price) as sum

from sales65

group by seller_id) as a1

where a.sum = (select max(b.sum)

from(select seller_id, sum(price) as sum

from sales65

group by seller_id as b1 )
```

66)

Write an SQL query to compute the moving average of how much the customer paid in a seven days window (i.e., current day + 6 days before). average_amount should be rounded to two decimal places.

ANS:

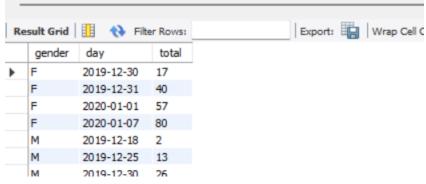
```
select
    visited_on,
    sum(amount) over(order by visited_on rows between 6 preceding and current row) as amount,
    round(avg(amount) over(order by visited_on rows between 6 preceding and current row),2) as average_amount,
    dense_rank() over(order by visited_on) as rnk
from result
```

68)

Write an SQL query to find the total score for each gender on each day. Return the result table ordered by gender and day in ascending order.

```
#QUERY-68#
select s.gender, s.day,

(select sum(score_points) from scores68
where gender = s.gender and day <= s.day) as total
from scores68 as s
group by gender, day
order by gender, day;
```



Write an SQL query to find the start and end number of continuous ranges in the table Logs. Return the result table ordered by start_id.

```
ANS:
```

```
with cte as(
select log_id,
log_id
70)
ANS:
```

71)

Write an SQL query to find employee_id of all employees that directly or indirectly report their work to the head of the company.

```
#QUERY-71#
select t1.employee_id
from Employees as t1 inner join employees71 as t2
on t1.manager_id = t2.employee_id
inner join employees71 as t3
on t2.manager_id = t3.employee_id
where t3.manager_id = 1 and t1.employee_id <> 1
```

Write an SQL query to find for each month and country, the number of transactions and their total amount, the number of approved transactions and their total amount.

ANS:

1241	#QUERY-72#
1242 •	<pre>select trans_date as month, country,</pre>
1243	<pre>count(id) as trans_count,</pre>
1244	<pre>sum(case when state='approved' then 1 else 0 end) as approved_count,</pre>
1245	<pre>sum(amount) as trans_total_amount,</pre>
1246	<pre>sum(case when state='approved' then amount else 0 end) as approved_total_amount</pre>
1247	from transactions72
1248	group by month, country;
1249	

Result Grid 1							
	month	country	trans_count	approved_count	trans_total_amount	approved_total_amount	
•	2018-12-18	US	1	1	1000	1000	
	2018-12-19	US	1	0	2000	0	
	2019-01-01	US	1	1	2000	2000	
	2019-01-07	DE	1	1	2000	2000	

73)

Write an SQL query to find the average daily percentage of posts that got removed after being reported as spam, rounded to 2 decimal places.

Write an SQL query to find the salaries of the employees after applying taxes. Round the salary to the nearest integer.

The tax rate is calculated for each company based on the following criteria:

- 0% If the max salary of any employee in the company is less than \$1000.
- 24% If the max salary of any employee in the company is in the range [1000, 10000] inclusive.
- 49% If the max salary of any employee in the company is greater than \$10000.

ANS:

77)

Write an SQL query to evaluate the boolean expressions in Expressions table. Return the result table in any order.

ANS:

78) Repeated previously

79)

Write a query that prints a list of employee names (i.e.: the name attribute) from the Employee table in alphabetical order.

```
select name from employee73 order by name;
```

Write a query to update the Facebook advertiser's status using the daily_pay table. Advertiser is a two-column table containing the user id and their payment status based on the last payment and daily_pay table has current information about their payment. Only advertisers who paid will show up in this table.

ANS:

```
#QUERY-84#
select advertiser.user_id, advertiser.status,payment.paid
from advertiser
left join daily_pay as pay
on advertiser.user_id = pay.user_id
union
select pay.user_id,advertiser.status,pay.paid
from daily_pay as pay
left join advertiser
    on advertiser.user_id = pay.user_id
)
```

88) Previously Repeated

89) Previously Repeated

91)

Write an SQL query to report the comparison result (higher/lower/same) of the average salary of employees in a department to the company's average salary.

ANS:

Write an SQL query to report for each install date, the number of players that installed the game on that day, and the day one retention.

```
#QUERY-92#
select t1.install_date as install_dt, count(t1.install_date) as installs,
count(t2.event_date) / count(*) as first_retention

from (
    select player_id, min(event_date) as install_date
    from activity92
    group by 1

) as team1
left join activity92 as team2
on t1.install_date = t2.event_date
    and t1.player_id = t2.player_id
group by 1
order by 1;
```

The winner in each group is the player who scored the maximum total points within the group. In the case of a tie, the lowest player_id wins.

Write an SQL query to find the winner in each group.

ANS:

94)

95) Repeated, same as 94)

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.

ANS:

```
#QUERY-96#
select user_id, song_id, sum(song_plays) as total_songs
from (
select user_id, song_id, song_plays
from songs_history
union all
select user_id, song_id, count(song_id) as songs_played
from songs_weekly
where listen_time <= '08/04/2022 23:59:59'
group by user_id, song_id
) as full
group by user_id, song_id
order by total_songs desc;</pre>
```

99)

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.

```
select
age.age_bucket,
sum(case when activities99.activity_type = 'sending_snaps'
    then activities99.time_spent else 0 end) as sending_time,
sum(case when activities99.activity_type = 'opening_snaps'
    then activities99.time_spent else 0 end) as opening_time,
sum(activities99.time_spent) as total_time
    from activities99
inner join age_breakdown as age_br
    on activities99.user_id = age_br.user_id
    where activities99.activity_type in ('sending_snaps', 'opening_snaps')
    group by age_br.age_bucket)
```