

SQL ASSIGNMENT SET 2

Q51) Write an SQL query to report the name, population, and area of the big countries. Return the result table in any order. The query result format is in the following example.

Sol) At first , create an empty table and then insert records in it using multi insert command.

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with various database objects. The main editor contains the following SQL script:

```
1 create database assignment1;
2 use assignment1;
3
4 create table world(
5     name varchar(15) primary key,
6     continent varchar(15), area int, population bigint, gdp bigint);
7
8 insert into world values('Afghanistan','Asia',652230,25500100,20343000000),
9 ('Albania','Europe',28748,2831741,12960000000),
10 ('Algeria','Africa',2381741,37100000,186681000000),
11 ('Andorra','Europe',468,78115,3712000000),
12 ('Angola','Africa',1246700,20609294,100990000000);
13
```

The right pane shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

The bottom pane shows the 'Output' window with the following log:

#	Time	Action	Message	Duration / Fetch
7	07:24:30	create table world (name varchar(15) primary key, continent varchar(15), area int, pop...	0 row(s) affected	0.016 sec
8	07:24:35	insert into world values('Afghanistan','Asia',652230,25500100,20343000000), ('Albani...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec

**select name,population,area from world
where area>=3000000 or population >= 25000000;**

The screenshot shows the SQL Developer interface. The main editor contains the following SQL query:

```
14
15 select name,population,area from world
16 where area>=3000000 or population >= 25000000;
```

The bottom pane shows the 'Output' window with the following log:

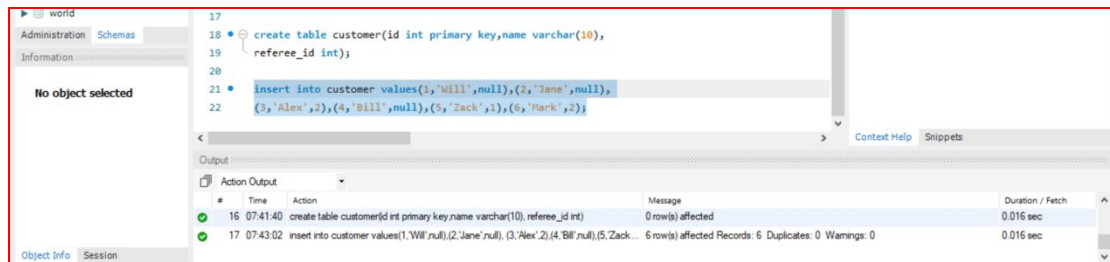
#	Time	Action	Message	Duration / Fetch
10	07:27:04	select * from world	5 row(s) returned	0.000 sec / 0.000 sec
11	07:33:24	select name,population,area from world where area>=3000000 or population >= 250...	2 row(s) returned	0.000 sec / 0.000 sec

The 'Result Grid' shows the following data:

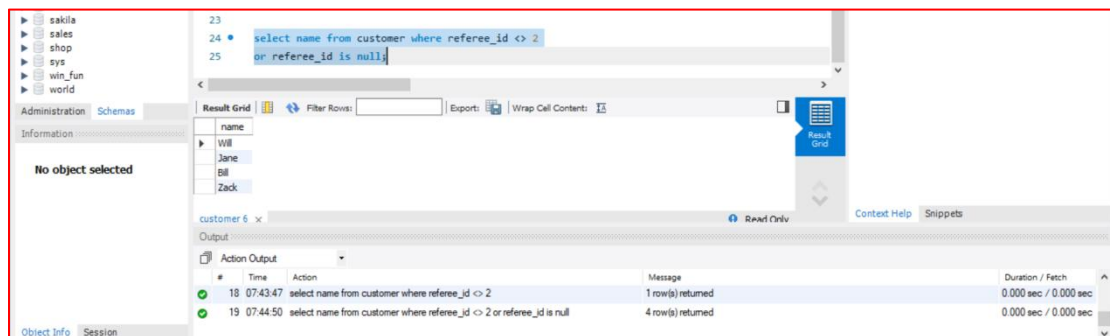
name	population	area
Afghanistan	25500100	652230
Algeria	37100000	2381741

Q52) Write an SQL query to report the names of the customer that are not referred by the customer with id = 2. Return the result table in any order. The query result format is in the following example.

Sol) At first , create an empty table named 'Customer' and then insert records in it using multi insert command.

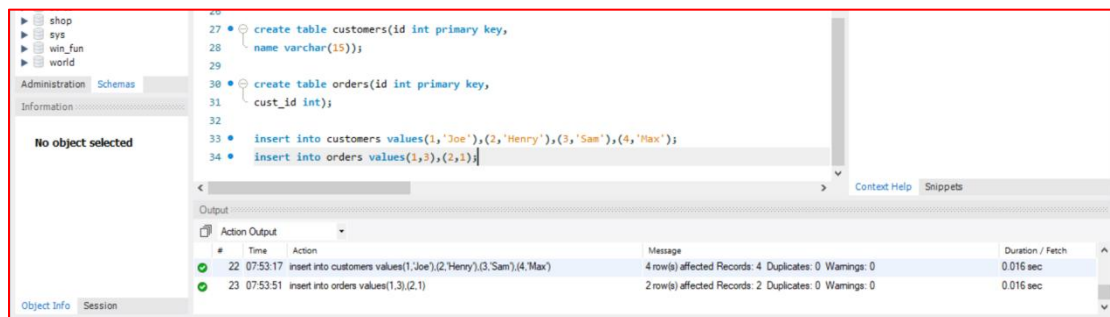


**select name from customer where referee_id <> 2
or referee_id is null;**

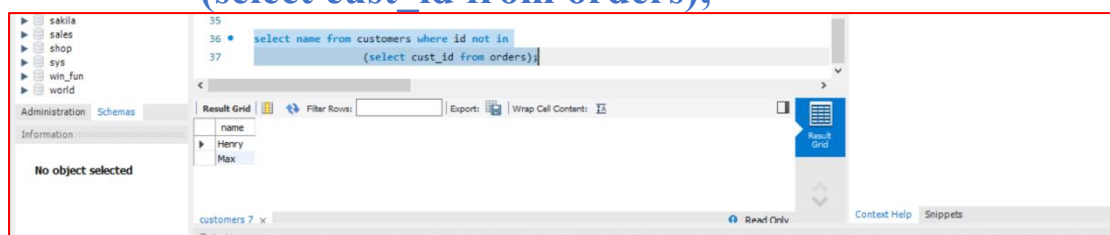


Q53) Write an SQL query to report all customers who never order anything. Return the result table in any order. The query result format is in the following example.

Sol) At first , create two empty table named 'Customers' and 'Orders' And then insert records in it using multi insert command.



**select name from customers where id not in
(select cust_id from orders);**

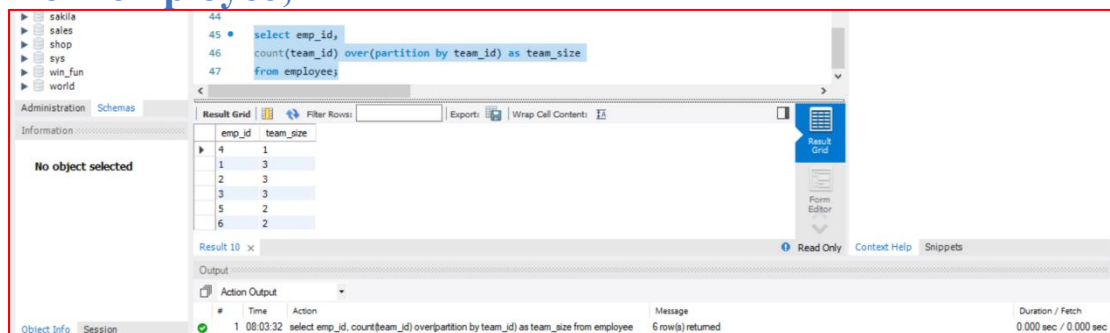


Q54) Write an SQL query to find the team size of each of the employees. Return result table in any order. The query result format is in the following example.

Sol) At first , create an empty table named 'Employee'. And then insert records in it using multi insert command.

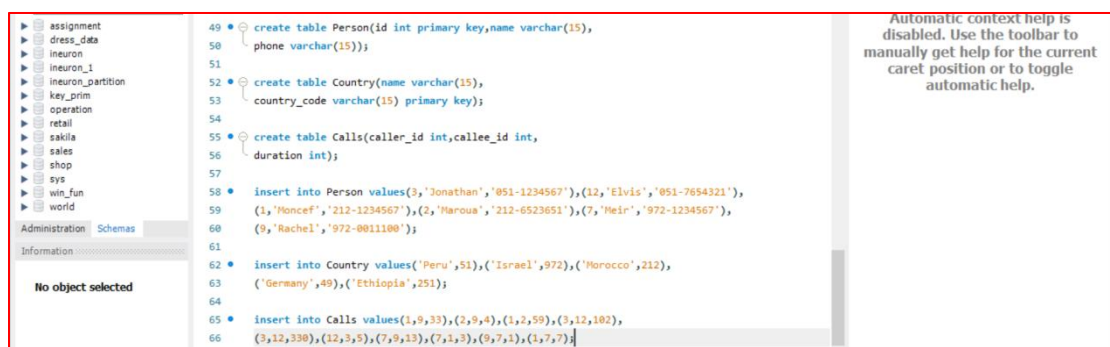


**select emp_id,
count(team_id) over(partition by team_id) as team_size
from employee;**



Q55) Write an SQL query to find the countries where this company can invest. Return the result table in any order. The query result format is in the following example

Sol) At first , create 3 empty table naming 'Person', 'Country' & 'Calls'. And then insert records in it using multi insert command.



```

with phn as(select caller_id as id,duration from calls
union all
select callee_id as id,duration from calls)

```

```

select c.name as country from phn
join person p on phn.id = p.id
join country c on left(p.phone,3) = c.country_code
group by c.name
having avg(duration)>(select avg(duration) from Calls);

```

Q56) Write an SQL query to report the device that is first logged in for each player. Return the result table in any order. The query result format is in the following example.

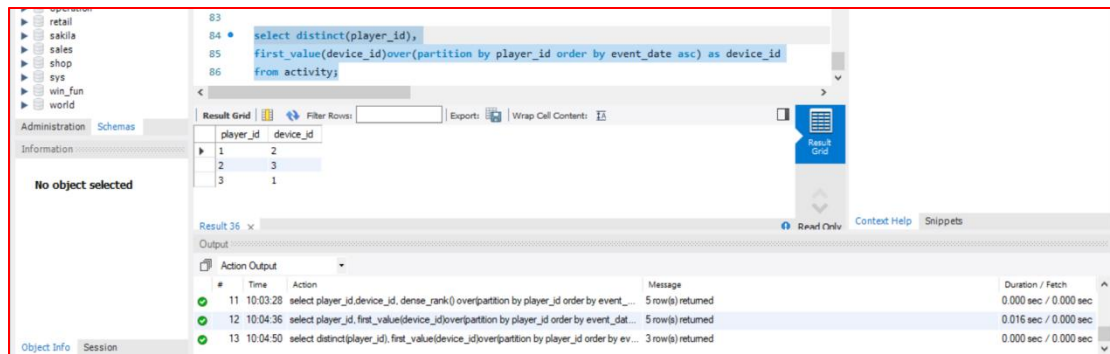
Sol) At first , create an empty table named ‘Activity’ and then insert records in it using multi insert command.



```

select distinct(player_id),
first_value(device_id)over(partition by player_id order by
event_date asc) as device_id
from activity;

```



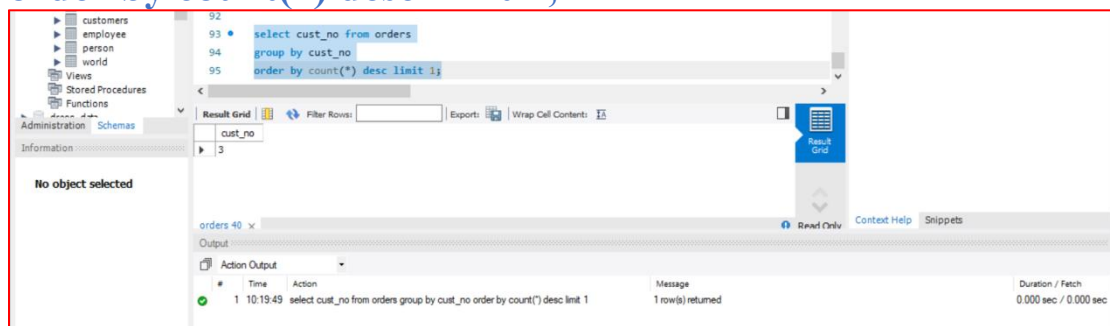
Q57) Write an SQL query to find the customer_number for the customer who has placed the largest number of orders. The test cases are generated so that exactly one customer will have placed more orders than any other customer. The query result format is in the following example.

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



After inserting records, now the table can be used to perform desired query over it.

**select cust_no from orders
group by cust_no
order by count(*) desc limit 1;**



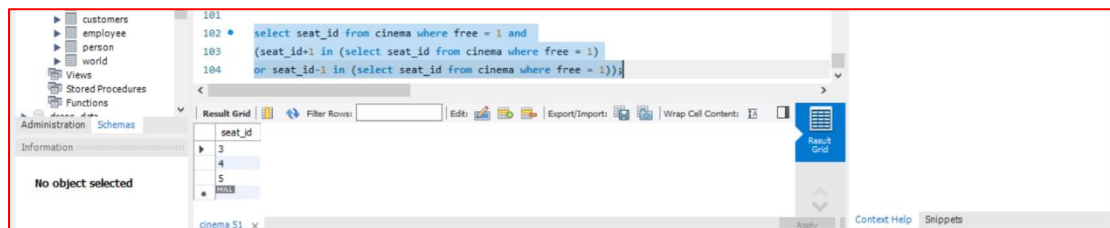
Q58) Write an SQL query to report all the consecutive available seats in the cinema. Return the result table ordered by seat_id in ascending order. The test cases are generated so that more than two

seats are consecutively available. The query result format is in the following example.

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

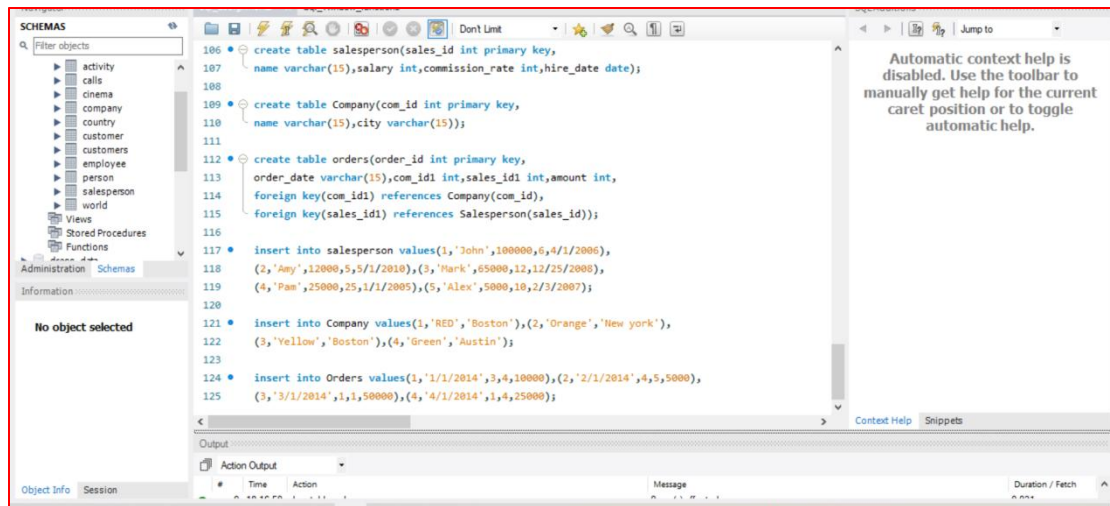


**select seat_id from cinema where free = 1 and
(seat_id+1 in (select seat_id from cinema where free = 1)
or seat_id-1 in (select seat_id from cinema where free = 1));**

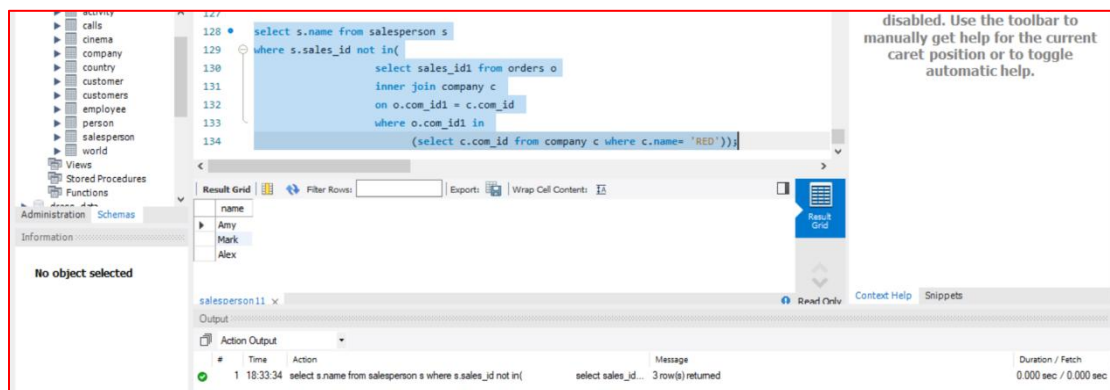


Q59) Write an SQL query to report the names of all the salespersons who did not have any orders related to the company with the name "RED". Return the result table in any order. The query result format is in the following example.

Sol) At first , create 3 empty table named ‘Salesperson’, ‘Company’ & ‘Orders’ and then insert records in it using multi insert command.

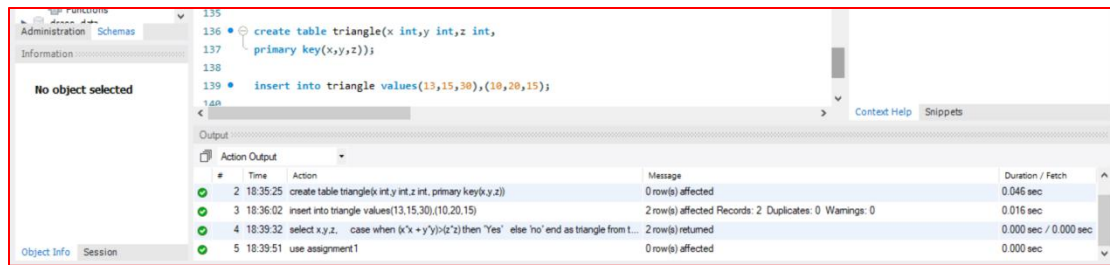


Select s.name from salesperson s
 where s.sales_id not in(
 select sales_id1 from orders o
 inner join company c
 on o.com_id1 = c.com_id
 where o.com_id1 in
 (select c.com_id from company c where
 c.name= 'RED'));



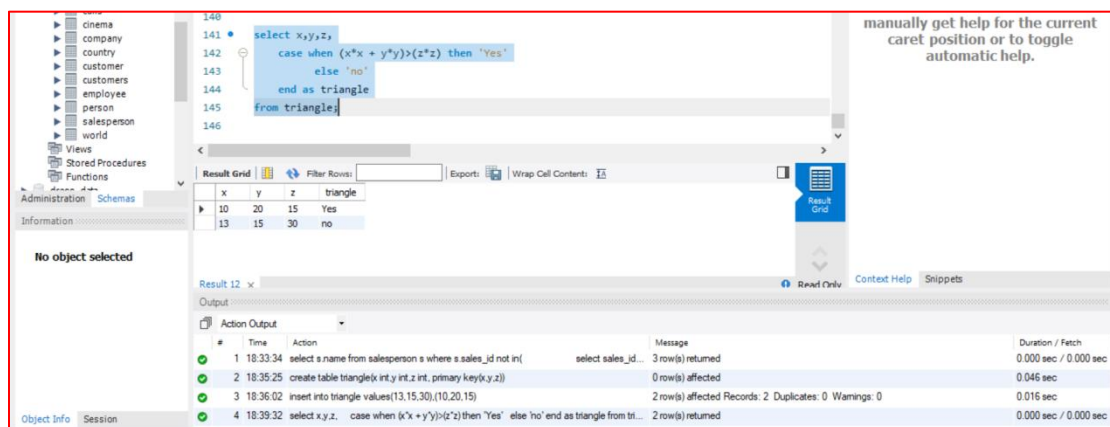
Q60) Write an SQL query to report for every three line segments whether they can form a triangle. Return the result table in any order. The query result format is in the following example

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



Given dimensions can be checked whether it will form a triangle or not using simple theorem of $a^2 + b^2 > c^2$.

```
select x,y,z,
       case when (x*x + y*y)>(z*z) then 'Yes'
              else 'no'
       end as triangle
from triangle;
```



Q61) Write an SQL query to report the shortest distance between any two points from the Point table. The query result format is in the following example.

Sol) At first , create an empty table named 'Point' and then insert records in it using multi insert command.

146
147 • create table point(x int primary key);
148 • insert into point values(-1),(0),(2);
149
150 • Select min(abs(p2.x - p1.x)) as shortest
151 from point p1 JOIN point p2
152 ON p1.x != p2.x;

Result Grid
shortest
1

Result 13 ×
Read Only Context Help Snippets

Output
Action Output

#	Time	Action	Message	Duration / Fetch
6	18:54:56	create table point(x int primary key)	0 row(s) affected	0.063 sec
7	18:55:38	insert into point values(-1,0,2)	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
8	18:56:18	insert into point values(-1,0,2)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
9	18:58:21	Select min(abs(p2.x - p1.x)) as shortest from point p1 JOIN point p2 ON p1.x != p2.x	1 row(s) returned	0.000 sec / 0.000 sec

Select min(abs(p2.x - p1.x)) as shortest
from point p1 JOIN point p2
ON p1.x != p2.x;

Q62) 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. Return the result table in any order. The query result format is in the following example.

Sol) At first , create an empty table named 'Actordirector' and then insert records in it using multi insert command.

153
154 • create table Actordirector(actor_id int,director_id int,
155 'timestamp' int primary key);
156
157 • insert into Actordirector values(1,1,0),(1,1,1),
158 (1,1,2),(1,2,3),(1,2,4),(2,1,5),(2,1,6);
159
160 • select distinct(actor_id),director_id from Actordirector where actor_id = director_id and
161 (actor_id + 1 = director_id + 1
162 or actor_id - 1 = director_id - 1);

Result Grid
actor_id director_id
1 1

Result 16 ×
Read Only Context Help Snippets

Output
Action Output

#	Time	Action	Message	Duration / Fetch
1	19:08:20	select distinct(actor_id),director_id from Actordirector where actor_id = director_id and...	1 row(s) returned	0.000 sec / 0.000 sec

select distinct(actor_id),director_id from Actordirector
where actor_id = director_id and
(actor_id + 1 = director_id + 1
or actor_id - 1 = director_id - 1);

Q63) Write an SQL query that reports the product_name, year, and price for each sale_id in the Sales table. Return the resulting table in any order. The query result format is in the following example.

Sol) At first , create 2 empty tables named 'Sales' & 'Product' and then insert records in it using multi insert command.

```

163
164 • create table Sales(sale_id int,product_id int,
165 year int,quantity int,price int,
166 primary key(sale_id,year));
167
168 • create table product(product_id int primary key,
169 product_name varchar(15));
170
171 • insert into Sales values(1,100,2008,10,5000),(2,100,2009,12,5000),
172 (7,200,2011,15,9000);
173
174 • insert into Product values(100,'Nokia'),(200,'Apple'),(300,'Samsung');
  
```

Output:

#	Time	Action	Message	Duration / Fetch
4	19:16:52	insert into Sales values(1,100,2008,10,5000),(2,100,2009,12,5000), (7,200,2011,15,9000);	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
5	19:17:44	insert into Product values(100,'Nokia'),(200,'Apple'),(300,'Samsung');	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec

**select p.product_name,s.year,s.price
from product p right join sales s
on p.product_id = s.product_id;**

```

175
176 • select p.product_name,s.year,s.price
177 from product p right join sales s
178 on p.product_id = s.product_id;
  
```

Result Grid:

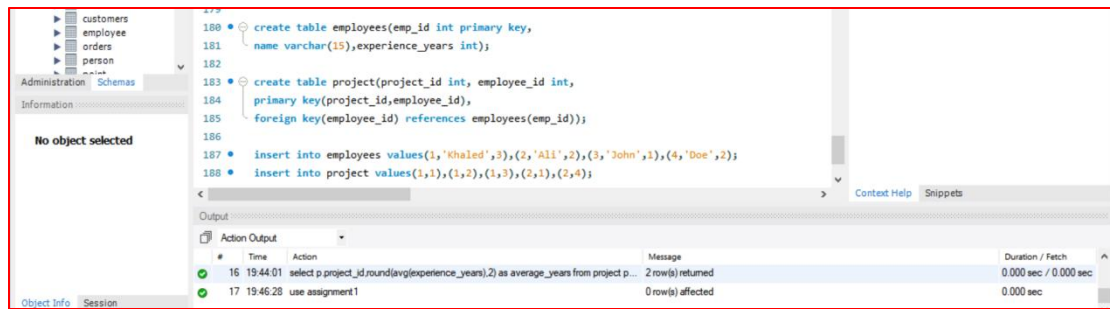
product_name	year	price
Nokia	2008	5000
Nokia	2009	5000
Apple	2011	9000

Output:

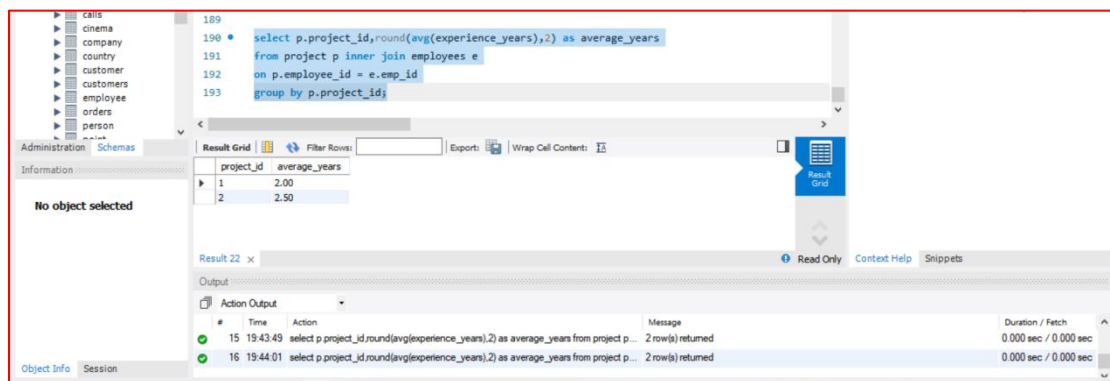
#	Time	Action	Message	Duration / Fetch
6	19:21:41	select p.product_name,s.year,s.price from product p right join sales s on p.product_id = s...	2 row(s) returned	0.000 sec / 0.000 sec
7	19:23:54	select p.product_name,s.year,s.price from product p right join sales s on p.product_id = s...	3 row(s) returned	0.000 sec / 0.000 sec

Q64) Write an SQL query that reports the average experience years of all the employees for each project, rounded to 2 digits. Return the result table in any order. The query result format is in the following example.

Sol) At first , create 2 empty tables named 'Project' & 'Employee' and then insert records in it using multi insert command.



**select p.project_id,round(avg(experience_years),2) as
average_years
from project p inner join employees e
on p.employee_id = e.emp_id
group by p.project_id;**



Q65) Write an SQL query that reports the best seller by total sales price, If there is a tie, report them all. Return the result table in any order. The query result format is in the following example.

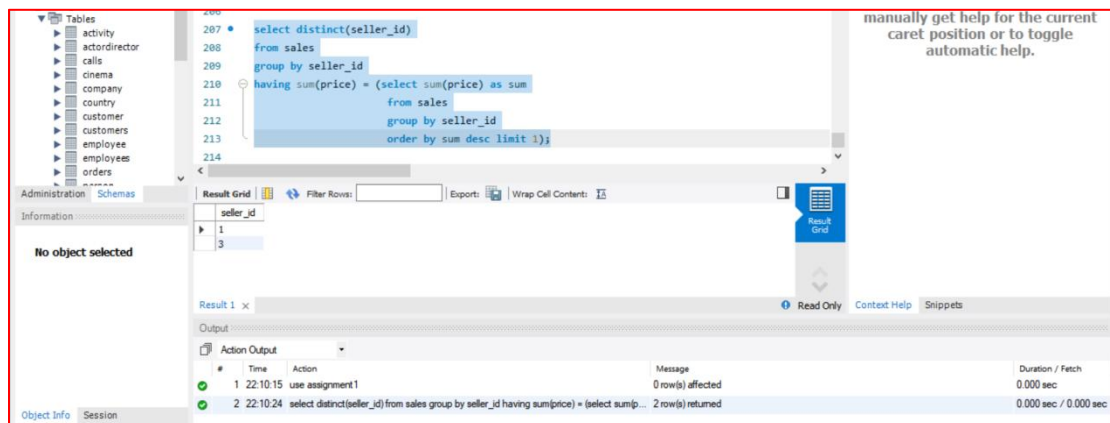
Sol) At first , create 2 empty tables named ‘Sales’ & ‘produkt’ and then insert records in it using multi insert command.



```

select distinct(seller_id)
from sales
group by seller_id
having sum(price) = (select sum(price) as sum
                    from sales
                    group by seller_id
                    order by sum desc limit 1);

```



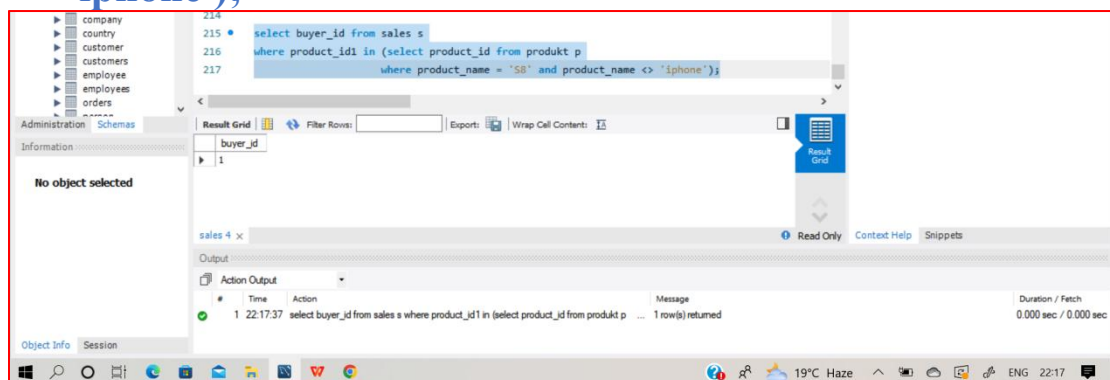
Q66) Write an SQL query that reports the buyers who have bought S8 but not iPhone. Note that S8 and iPhone are products present in the Product table. Return the result table in any order. The query result format is in the following example.

Sol) In this question the two tables used are already created earlier in the last question naming 'produkt' & 'Sales'. Hence there is only need of query analysis on both tables.

```

select buyer_id from sales s
where product_id1 in (select product_id from produkt p
                     where product_name = 'S8' and product_name
<> 'iphone');

```



Q67) 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. Return result table ordered by visited_on in ascending order. The query result format is in the following example.

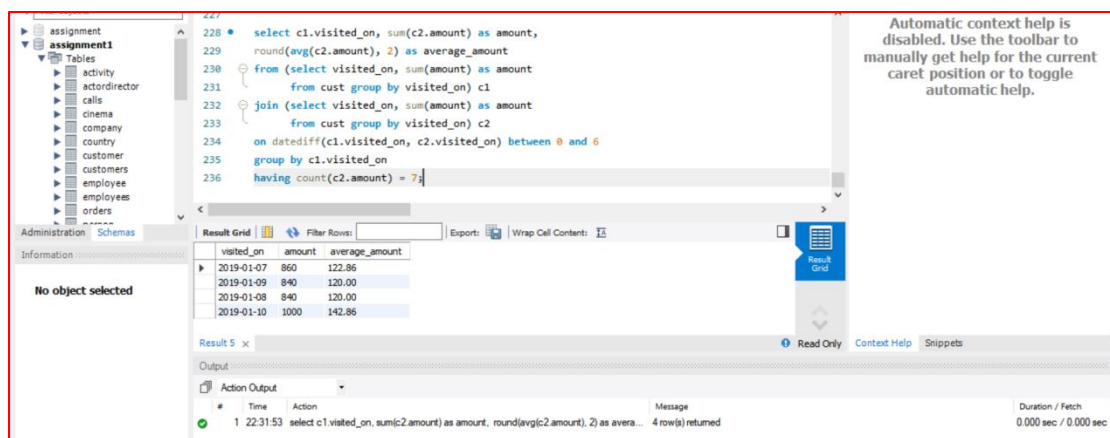
Sol) At first , create an empty table named ‘Cust’ and then insert records in it using multi insert command.



```

select c1.visited_on, sum(c2.amount) as amount,
round(avg(c2.amount), 2) as average_amount
from (select visited_on, sum(amount) as amount
      from cust group by visited_on) c1
join (select visited_on, sum(amount) as amount
      from cust group by visited_on) c2
on datediff(c1.visited_on, c2.visited_on) between 0 and 6
group by c1.visited_on
having count(c2.amount) = 7;

```

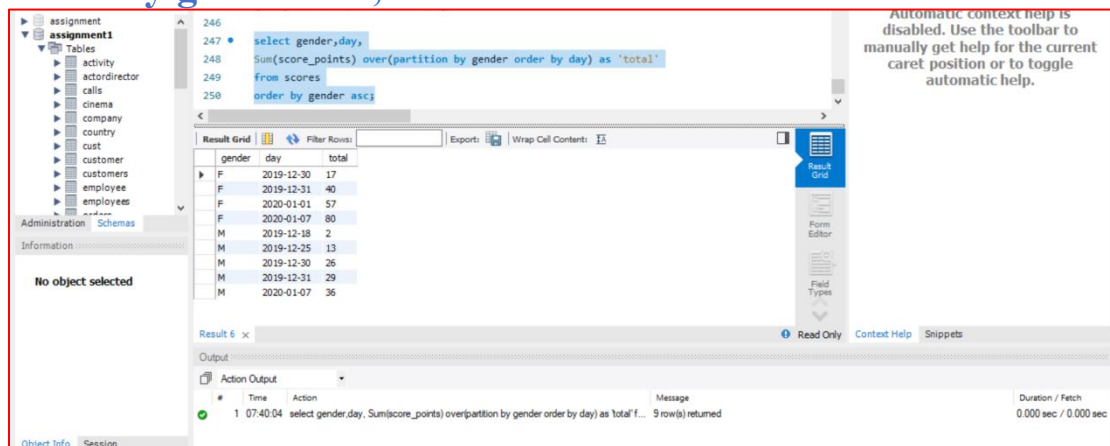


Q68) Write an SQL query to find the total score for each gender on each day. Return the result table ordered by gender and day in ascending order. The query result format is in the following example.

Sol) At first , create an empty table named 'Scores' and then insert records in it using multi insert command.



**Select gender,day,
Sum(score_points) over(partition by gender order by day) as
'total'
from scores
order by gender asc;**



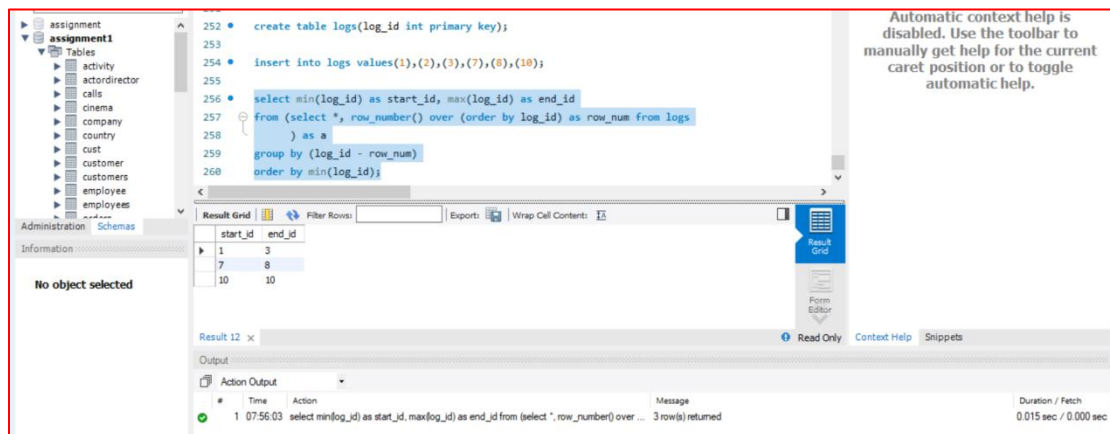
Q69) Write an SQL query to find the start and end number of continuous ranges in the table Logs. Return the result table ordered by start_id. The query result format is in the following example.

Sol) At first , create an empty table named 'Logs' and then insert records in it using multi insert command.


```

select min(log_id) as start_id, max(log_id) as end_id
from (select *, row_number() over (order by log_id) as
row_num from logs
) as a
group by (log_id - row_num)
order by min(log_id);

```



Q70) Write an SQL query to find the number of times each student attended each exam. Return the result table ordered by student_id and subject_name. The query result format is in the following example.

Sol) At first , create 3 empty table named 'Students', 'Subjects' & 'Exams' and then insert records in it using multi insert command.



```

select
a.student_id,a.student_name,b.subject_name1,count(c.stude
nt_id1) as 'attended_exams'
from students a inner join subjects b

```

left join exams c
 on a.student_id = c.student_id1 and b.subject_name1 =
 c.subject_name2
 group by a.student_id,b.subject_name1;

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

276
277 * select a.student_id,a.student_name,b.subject_name1,count(c.student_id1) as 'attended_exams'
278 from students a inner join subjects b
279 left join exams c
280 on a.student_id = c.student_id1 and b.subject_name1 = c.subject_name2
281 group by a.student_id,b.subject_name1;
  
```

student_id	student_name	subject_name1	attended_exams
1	Alice	Programming	1
1	Alice	Physics	2
1	Alice	Math	3
2	Bob	Programming	0
2	Bob	Physics	0
2	Bob	Math	1
6	Alex	Programming	0
6	Alex	Physics	0
6	Alex	Math	0
13	John	Programming	1
13	John	Physics	1
13	John	Math	1

Result 25 x

Output

#	Time	Action	Message	Duration / Fetch
1	09:04:59	select a.student_id,a.student_name,b.subject_name1,count(c.student_id1) as 'attend...	12 row(s) returned	0.000 sec / 0.000 sec

Q71) Write an SQL query to find employee_id of all employees that directly or indirectly report their work to the head of the company. The indirect relation between managers will not exceed three managers as the company is small. Return the result table in any order. The query result format is in the following example.

Sol) At first , create an empty table named 'Employees' and then insert records in it using multi insert command.

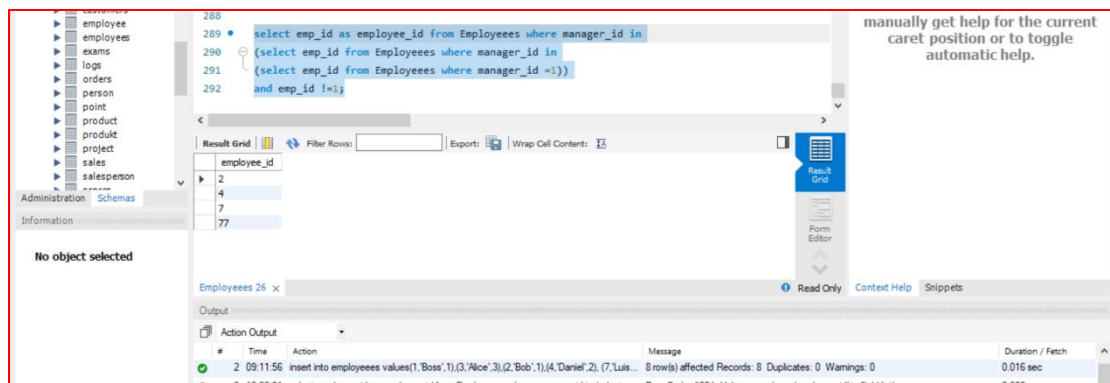
```

283 * create table employees(emp_id int primary key,emp_name varchar(15),
284 manager_id int);
285
286 * insert into employees values(1,'Boss',1),(3,'Alice',3),(2,'Bob',1),(4,'Daniel',2),
287 (7,'Luis',4),(8,'John',3),(9,'Angela',8),(77,'Robert',1);
288
  
```

Output

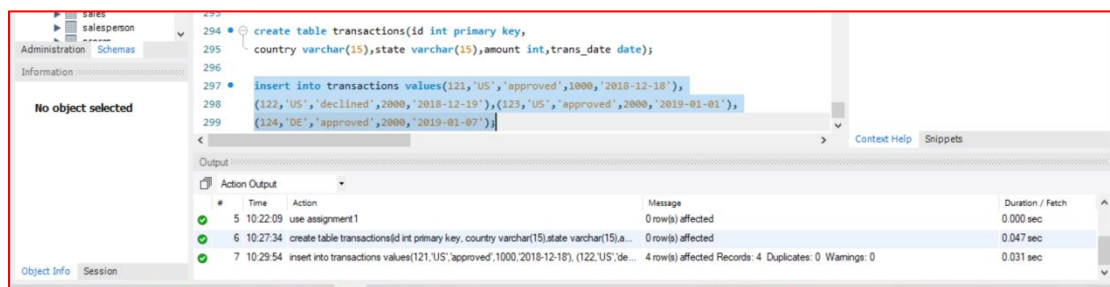
#	Time	Action	Message	Duration / Fetch
1	09:10:05	create table employees(emp_id int primary key,emp_name varchar(15), manager_id...	0 row(s) affected	0.047 sec
2	09:11:56	insert into employees values(1,'Boss',1),(3,'Alice',3),(2,'Bob',1),(4,'Daniel',2),...	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec

select emp_id as employee_id
 from Employees where manager_id in
 (select emp_id from Employees where manager_id in
 (select emp_id from Employees where manager_id =1))
 and emp_id <>1;



Q72) 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. Return the result table in any order. The query result format is in the following example.

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



**select date_format(trans_date,"%y-%m") as month,country,
count(id) as trans_count,**

**sum(case when state='approved' then 1 else 0 end) as
'approved_count',**

sum(amount) as trans_total_amount,

**sum(case when state='approved' then amount else 0 end) as
'approved_total_amount'**

**from transactions
group by month,country;**

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

301 select date_format(trans_date,"%y-%m") as month,country,
302 count(id) as trans_count,
303 sum(case when state='approved' then 1 else 0 end) as 'approved_count',
304 sum(amount) as trans_total_amount,
305 sum(case when state='approved' then amount else 0 end) as 'approved_total_amount'
306 from transactions
307 group by month,country;

```

month	country	trans_count	approved_count	trans_total_amount	approved_total_amount
18-12	US	2	1	3000	1000
19-01	US	1	1	2000	2000
19-01	DE	1	1	2000	2000

Result 29 x

Output

#	Time	Action	Message	Duration / Fetch
1	10:42:22	select date_format(trans_date,"%y-%m") as month,country, count(id) as trans_count, ...	3 row(s) returned	0.000 sec / 0.000 sec

Q73) 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. The query result format is in the following example.

Sol) At first , create an empty table named 'Actions' and then insert records in it using multi insert command.

automatic help.

```

309 create table actions(user_id int,post_id int,
310 action_date date,action varchar(15),extra varchar(15));
311
312 create table removals(post_id int primary key,remove_date date);
313
314 insert into actions values(1,1,'2019-07-01','view','null'),
315 (1,1,'2019-07-01','like','null'),(1,1,'2019-07-01','share','null'),
316 (2,2,'2019-07-04','view','null'),(2,2,'2019-07-04','report','spam'),
317 (3,4,'2019-07-04','view','null'),(3,4,'2019-07-04','report','spam'),
318 (4,3,'2019-07-02','view','null'),(4,3,'2019-07-02','report','spam');
319
320 insert into removals values(2,'2019-07-20'),(3,'2019-07-18');

```

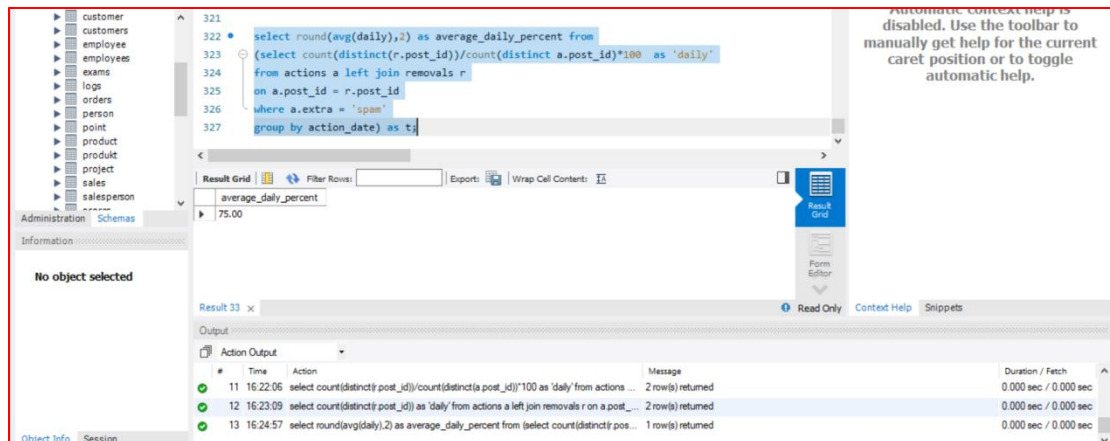
Output

#	Time	Action	Message	Duration / Fetch
4	10:52:07	insert into actions values(1,1,'2019-07-01','view','null'), (1,1,'2019-07-01','like','null'), (...)	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.031 sec
5	10:52:54	insert into removals values(2,'2019-07-20'),(3,'2019-07-18')	Error Code: 1292. Incorrect date value: '1994' for column 'remove_date' at row 2	0.016 sec
6	10:53:13	insert into removals values(2,'2019-07-20'),(3,'2019-07-18')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

```

select round(avg(daily),2)
as 'average_daily_percent' from
(select count(distinct(r.post_id))/count(distinct
a.post_id)*100 as 'daily'
from actions a left join removals r
on a.post_id = r.post_id
where a.extra = 'spam'
group by action_date) as t;

```

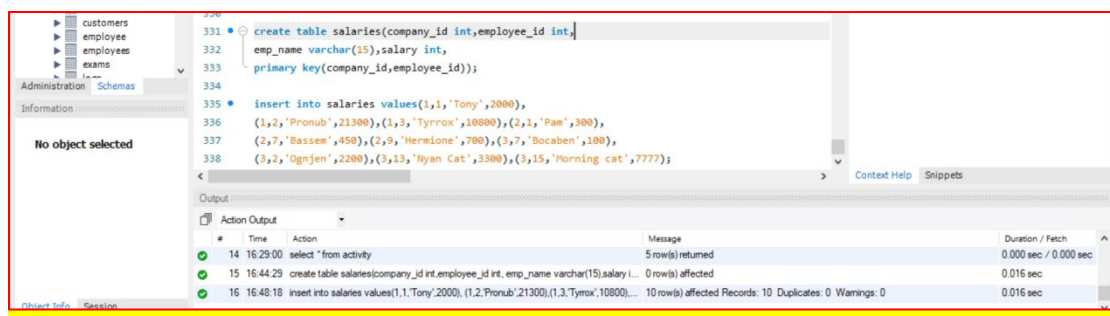


Q74,75) SAME QUESTION AS Q43

**Q76)_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

Sol) At first , create an empty table named 'salaries' and then insert records in it using multi insert command.



with cte as(
 select company_id,
 (case when max(salary) < 1000 then 1.0
 when max(salary) <= 10000 then 0.76


```

else 0.51
end) as tax
from salaries
group by company_id)

```

```

select      s.company_id,      s.employee_id,      s.emp_name,
ROUND(s.salary * t.tax) as salary
from Salaries s join cte t on s.company_id = t.company_id;

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

company_id	employee_id	emp_name	salary
1	1	Tony	1020
1	2	Ponrub	10863
1	3	Tyrox	5508
2	1	Pam	300
2	7	Bassem	450
2	9	Hermione	700
3	2	Ogryen	1672
3	7	Bocaben	76

Q77) 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. The query result format is in the following example.

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

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

create table sales(sale_date date,fruit varchar(15),sold_num int,
primary key(sale_date,fruit));

insert into sales values('2020-05-01','apples',10),('2020-05-01','oranges',8),
('2020-05-02','apples',15),('2020-05-02','oranges',15),('2020-05-03','apples',20),
('2020-05-03','oranges',8),('2020-05-04','apples',15),('2020-05-04','oranges',16);

select s.sale_date,
sum(if(s.fruit = 'apples', s.sold_num, -s.sold_num)) as diff
from Sales s
group by s.sale_date;

```

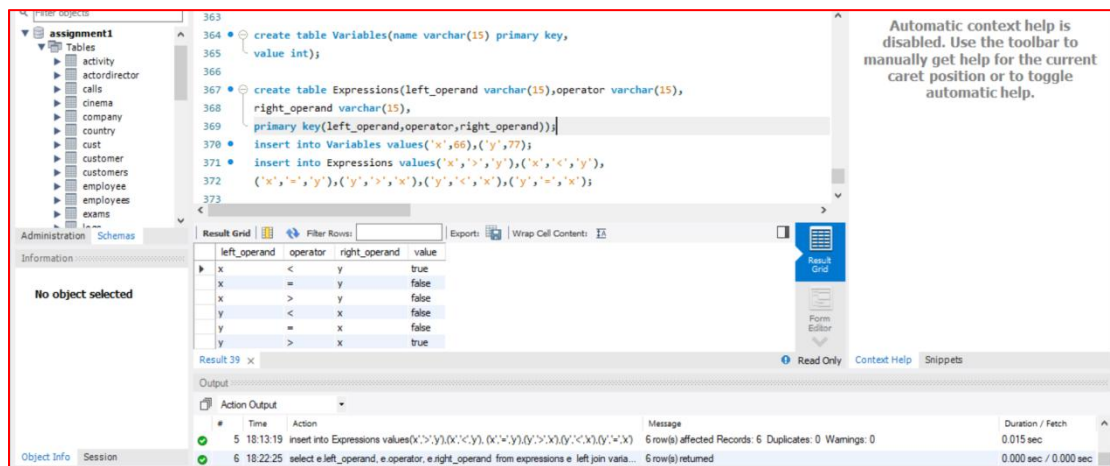
sale_date	diff
2020-05-01	2
2020-05-02	0
2020-05-03	20
2020-05-04	-1

1 18:05:47 select s.sale_date, sum(if(s.fruit = 'apples', s.sold_num, -s.sold_num)) as diff from Sale... 4 row(s) returned 0.000 sec / 0.000 sec


```
select s.sale_date,
sum(if(s.fruit = 'apples', s.sold_num, -s.sold_num)) as diff
from Sales s
group by s.sale_date;
```

Q78) Write an SQL query to evaluate the boolean expressions in Expressions table. Return the result table in any order. The query result format is in the following example.

Sol) At first , create two empty table named 'variables' & 'Expressions' and then insert records in it using multi insert command.



```
select e.left_operand, e.operator, e.right_operand,
(case
  when e.operator = '<' then if(l.value <
r.value,'true','false')
  when e.operator = '>' then if(l.value >
r.value,'true','false')
  else if(l.value = r.value,'true','false')
end) as value
from expressions e
left join variables l on e.left_operand = l.name
left join variables r on e.right_operand = r.name;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

374 select e.left_operand, e.operator, e.right_operand,
375 (
376   case
377     when e.operator = '<' then if(1.value < r.value,'true','false')
378     when e.operator = '>' then if(1.value > r.value,'true','false')
379     else if(1.value = r.value,'true','false')
380   end) as value
381 from expressions e
382 left join variables l on e.left_operand = l.name
383 left join variables r on e.right_operand = r.name;

```

left_operand	operator	right_operand	value
x	<	y	true
x	=	y	false
x	>	y	false
y	<	x	false
y	=	x	false
y	>	x	true

Result 39 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	18:13:19	insert into Expressions values(x<y),(x<y),(x<y),(y>x),(y>x),(y>x)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
6	18:22:25	select e.left_operand, e.operator, e.right_operand from expressions e left join variables l on e.left_operand = l.name left join variables r on e.right_operand = r.name;	6 row(s) returned	0.000 sec / 0.000 sec

Q79) SAME QUESTION AS Q35

Q80) SAME QUESTION AS Q55

Q81) 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 ID.

Sol) At first , create an empty table named 'student' and then insert values in it.

**select name from student
where marks>75
order by SUBSTR(Name, -3), id asc;**

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

383
384 create table student(id int,name varchar(15),marks int);
385
386 insert into student values(1,'Ashley',81),(2,'Samantha',75),(4,'Julia',76),
387 (3,'Belvet',84);
388
389 select name from student
390 where marks>75
391 order by SUBSTR(Name, -3), id asc;

```

name
Ashley
Julia
Belvet

student 40 x

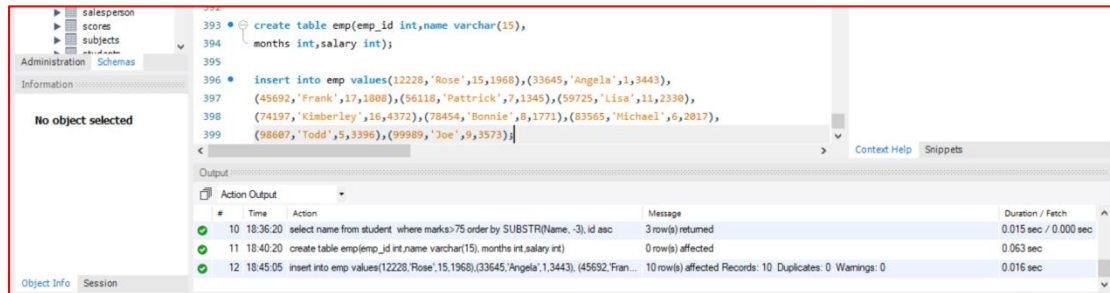
Output

Action Output

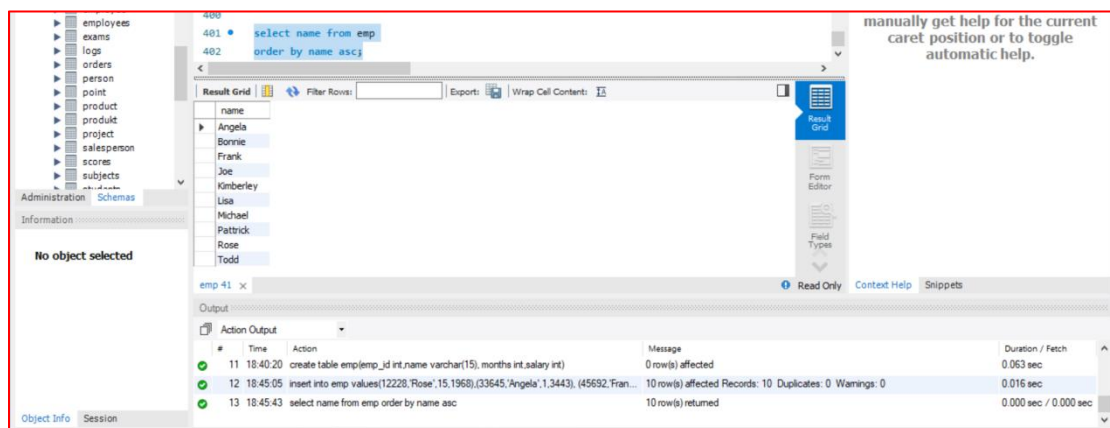
#	Time	Action	Message	Duration / Fetch
8	18:32:01	create table student(id int,name varchar(15),marks int)	0 row(s) affected	0.047 sec
9	18:33:16	insert into student values(1,'Ashley',81),(2,'Samantha',75),(4,'Julia',76),(3,'Belvet',84)	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.000 sec
10	18:36:20	select name from student where marks>75 order by SUBSTR(Name, -3), id asc	3 row(s) returned	0.015 sec / 0.000 sec

Q82) Write a query that prints a list of employee names (i.e.: the name attribute) from the Employee table in alphabetical order.

Sol) At first , create an empty table named 'emp' and then insert values in it.



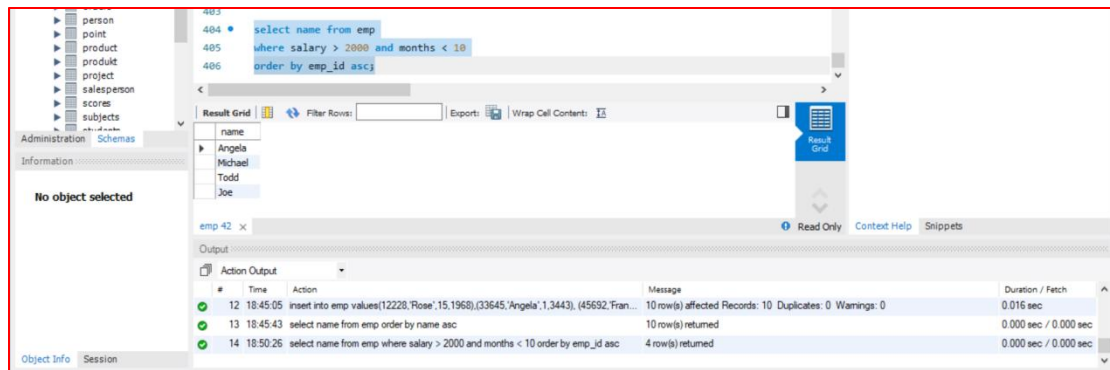
**select name from emp
order by name asc;**



Q83) Write a query that prints a list of employee names (i.e.: the name attribute) for employees in Employee having a salary greater than \$2000 per month who have been employees for less than 10 months. Sort your result by ascending employee_id.

Sol) The same table is used as in the previous question. In this question we have to find out names having salary greater than 2000 dollars with work experience less than 10 months.

**select name from emp
where salary > 2000 and months < 10
order by emp_id asc;**



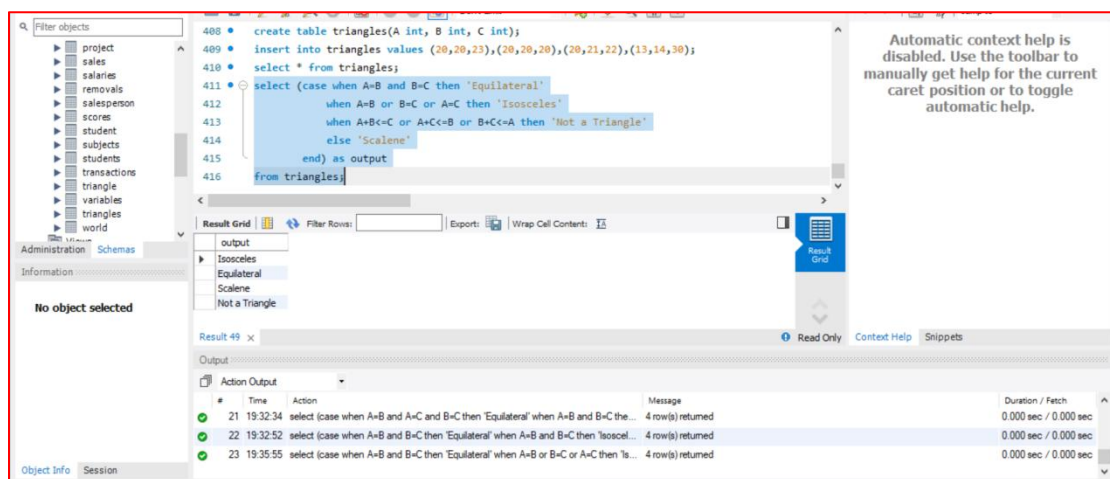
Q84) Write a query identifying the type of each record in the TRIANGLES table using its three side lengths.

Output one of the following statements for each record in the table:

- **Equilateral:** It's a triangle with sides of equal length.
- **Isosceles:** It's a triangle with sides of equal length.
- **Scalene:** It's a triangle with sides of differing lengths.
- **Not A Triangle:** The given values of A, B, and C don't form a triangle.

Sol) At first create an empty table named 'Triangles' and then insert the values in it as per in the question.

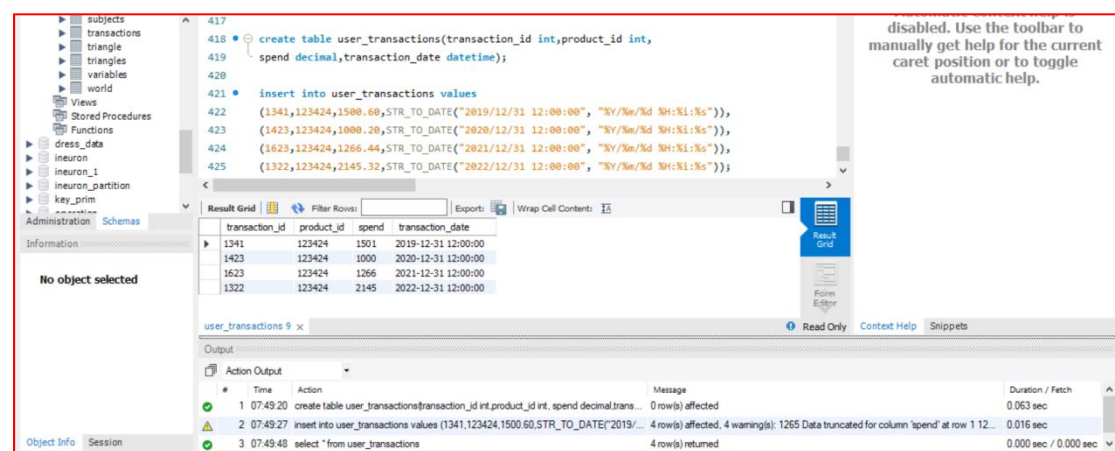
**select (case when A=B and B=C then 'Equilateral'
when A=B or B=C or A=C then 'Isosceles'
when A+B<=C or A+C<=B or B+C<=A then 'Not a Triangle'
else 'Scalene'
end) as output
from triangles;**



Q85) Assume you are given the table below containing information on user transactions for particular products. Write a query to obtain the year-on-year growth rate for the total spend of each product for each year.

Output the year (in ascending order) partitioned by product id, current year's spend, previous year's spend and year-on-year growth rate (percentage rounded to 2 decimal places).

Sol) At first create an empty table 'user_transactions' and then insert value in it. The values of the transaction date column in the table is not in standard MYSQL format , hence I have converted the 4 values to the standard values using STR_TO_DATE function.



After inserting the values and converting the timestamp datatype to the standard format , the table is ready for analysis purpose. Here in the query, I have extracted the year part from 'transaction_date' using extract function And by using 2 common table expressions.

with cte1 as

**(select extract(year from transaction_date) as year,
product_id,spend as 'curr_year_spend'
from user_transactions),**

cte2 as

**(select *,lag(curr_year_spend,1) over(partition by
product_id order by product_id,year)
as 'prev_year_spend' from cte1)**

**select year,product_id,curr_year_spend,prev_year_spend,
round(100*(curr_year_spend-
prev_year_spend)/prev_year_spend,2) as 'yoy_rate'
from cte2;**

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

429 with cte1 as
430 (select extract(year from transaction_date) as year,
431 product_id, spend as 'curr_year_spend'
432 from user_transactions),
433 cte2 as
434 (select *, lag(curr_year_spend, 1) over(partition by product_id order by product_id, year)
435 as 'prev_year_spend' from cte1)
436 select year, product_id, curr_year_spend, prev_year_spend,
437 round(100*(curr_year_spend-prev_year_spend)/prev_year_spend, 2) as 'yoy_rate'
438 from cte2;

```

year	product_id	curr_year_spend	prev_year_spend	yoy_rate
2019	123424	1501		
2020	123424	1000	1501	-33.38
2021	123424	1266	1000	26.60
2022	123424	2145	1266	69.43

Result 15 x

Output

Action Output

Time Action Message Duration / Fetch

1 08:09:22 with cte1 as (select extract(year from transaction_date) as year, product... 4 row(s) returned 0.000 sec / 0.000 sec

Q86) Amazon wants to maximize the number of items it can stock in a 500,000 square feet warehouse. It wants to stock as many prime items as possible, and afterwards use the remaining square footage to stock the most number of non-prime items. Write a SQL query to find the number of prime and non-prime items that can be stored in the 500,000 square feet warehouse. Output the item type and number of items to be stocked.

Sol) At first create an empty table 'Inventory' and then insert value in it Using multi insert statement.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

450 prime_items as (
451 select distinct item_type, total_sqft,
452 truncate(500000/total_sqft, 0) as prime_combo,
453 (truncate(500000/total_sqft, 0) * item_count) as prime_count from summary
454 where item_type = 'prime_eligible'),
455 non_prime_items as (select distinct item_type, total_sqft,
456 truncate((500000 - (select prime_combo * total_sqft from prime_items))
457 / total_sqft, 0) * item_count as non_prime_item_count
458 from summary where item_type = 'not_prime')
459 select item_type, prime_count as item_count from prime_items
460 union all
461 select item_type, non_prime_item_count as item_count from non_prime_items;
462

```

item_type	item_count
prime_eligible	9258
not_prime	2

Result 21 x

Output

Action Output


```

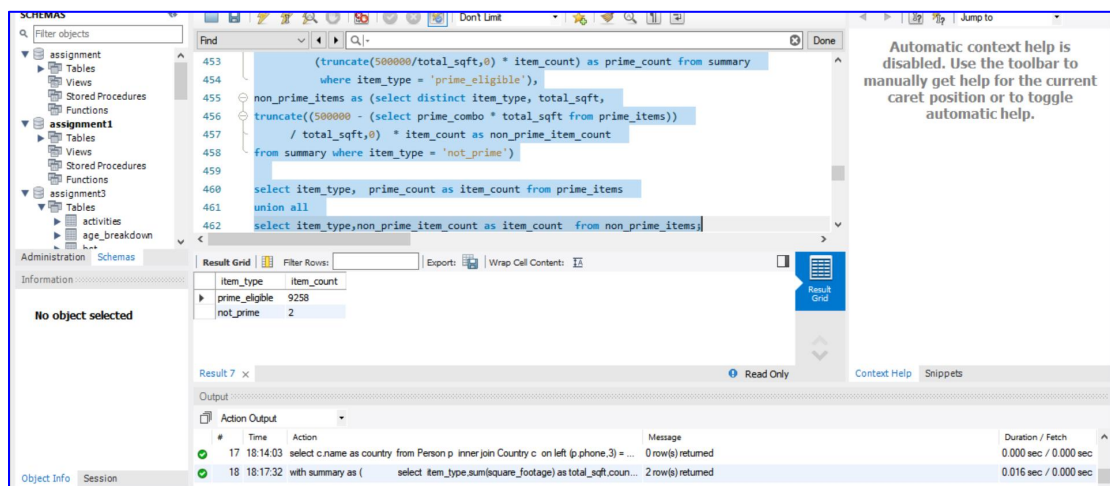
with summary as (
    select item_type,sum(square_footage) as
total_sqft,count(*) as 'item_count'
    from inventory group by item_type),
prime_items as (
    select distinct item_type, total_sqft,
    truncate(500000/total_sqft,0) as prime_combo,
    (truncate(500000/total_sqft,0) * item_count) as
prime_count from summary
    where item_type = 'prime_eligible'),
non_prime_items as (select distinct item_type, total_sqft,
truncate((500000 - (select prime_combo * total_sqft from
prime_items))
    / total_sqft,0) * item_count as non_prime_item_count
from summary where item_type = 'not_prime')

```

```

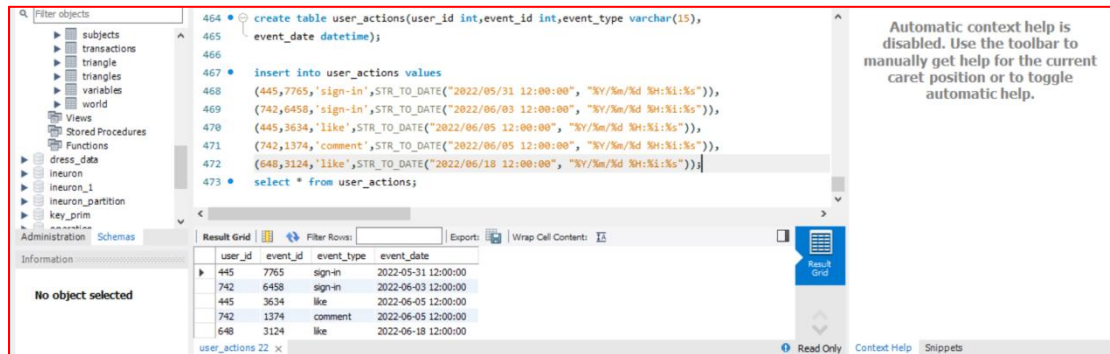
select item_type, prime_count as item_count from
prime_items
union all
select item_type,non_prime_item_count as item_count from
non_prime_items;

```



Q87) Assume you have the table below containing information on Facebook user actions. Write a query to obtain the active user retention in July 2022. Output the month (in numerical format 1, 2, 3) and the number of monthly active users (MAUs).

Sol) At first create an empty table 'user_actions' and then insert value in it. The values of the 'event date' column in the table is not in standard MYSQL format , hence I have converted the 5 values to the standard values using STR_TO_DATE function.



```
select
  extract(month from curr_month.event_date) as mth,
  count(distinct curr_month.user_id) as monthly_active_users
from user_actions as curr_month
where exists (
  select last_month.user_id
  from user_actions as last_month
  where last_month.user_id = curr_month.user_id
    and extract(month from last_month.event_date) =
      extract(month from curr_month.event_date - interval '1 month')
)
and extract(month from curr_month.event_date) = 7
group by extract(month from curr_month.event_date);
```

Q88) Google's marketing team is making a Superbowl commercial and needs a simple statistic to put on their TV ad: the median number of searches a person made last year. However, at Google scale, querying the 2 trillion searches is too costly. Luckily, you have access to the summary table which tells you the number of searches made last year and how many Google users fall into that bucket. Write a query to report the median of searches made by a user. Round the median to one decimal

Sol) At first create an empty table 'search' and then insert value in it Using multi insert statement.

The screenshot shows a SQL IDE interface with a query editor on the left and a result grid on the right. The query editor contains the following SQL statements:

```
487
488 • create table search(searches int,num_users int);
489 • insert into search values(1,2),(2,2),(3,3),(4,1);
490 • select * from search;
```

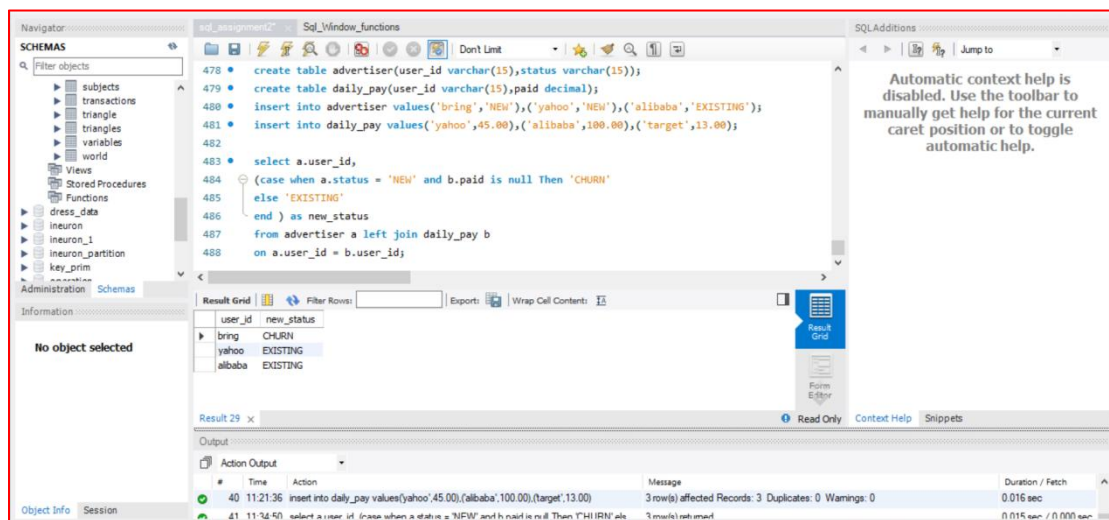
The result grid displays the following data:

searches	num_users
1	2
2	2
3	3
4	1

The bottom panel shows the output of the query, indicating that 4 rows were returned.

Q89) 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. Output the user id and current payment status sorted by the user id.

Sol)At first create an empty table 'advertiser' & 'daily_table' and then insert value in it using multi insert statement.



```

select a.user_id,
(case when a.status = 'NEW' and b.paid is null Then
'CHURN'
else 'EXISTING'
end ) as new_status
from advertiser a left join daily_pay b
on a.user_id = b.user_id;

```

Q90) Amazon Web Services (AWS) is powered by fleets of servers. Senior management has requested data-driven solutions to optimize server usage. Write a query that calculates the total time that the fleet of servers was running. The output should be in units of full days.

Sol)At first create an empty table 'severs' and then insert value in it using multi insert statement.

Filter objects

subjects

transactions

triangle

triangles

variables

world

Views

Stored Procedures

Functions

dress_data

neuron

neuron_1

neuron_partition

key_prim

Administration

Schemas

No object selected

491 session_status varchar(15));

492

493 insert into servers values

494 (1,STR_TO_DATE("2022/08/02 10:00:00", "%Y/%m/%d %H:%i:%s"),'start'),

495 (1,STR_TO_DATE("2022/08/04 10:00:00", "%Y/%m/%d %H:%i:%s"),'stop'),

496 (2,STR_TO_DATE("2022/08/17 10:00:00", "%Y/%m/%d %H:%i:%s"),'start'),

497 (2,STR_TO_DATE("2022/08/24 10:00:00", "%Y/%m/%d %H:%i:%s"),'stop');

498

499 select * from servers;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

server_id	status_time	session_status
1	2022-08-02 10:00:00	start
1	2022-08-04 10:00:00	stop
2	2022-08-17 10:00:00	start
2	2022-08-24 10:00:00	stop

servers 30 x

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
42	16:45:58	create table servers(server_id int,status_time timestamp, session_status varchar(15))	0 row(s) affected	0.047 sec
43	16:49:30	insert into servers values (1,STR_TO_DATE("2022/08/02 10:00:00", "%Y/%m/%d ...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.015 sec
44	16:49:46	select * from servers	4 row(s) returned	0.000 sec / 0.000 sec

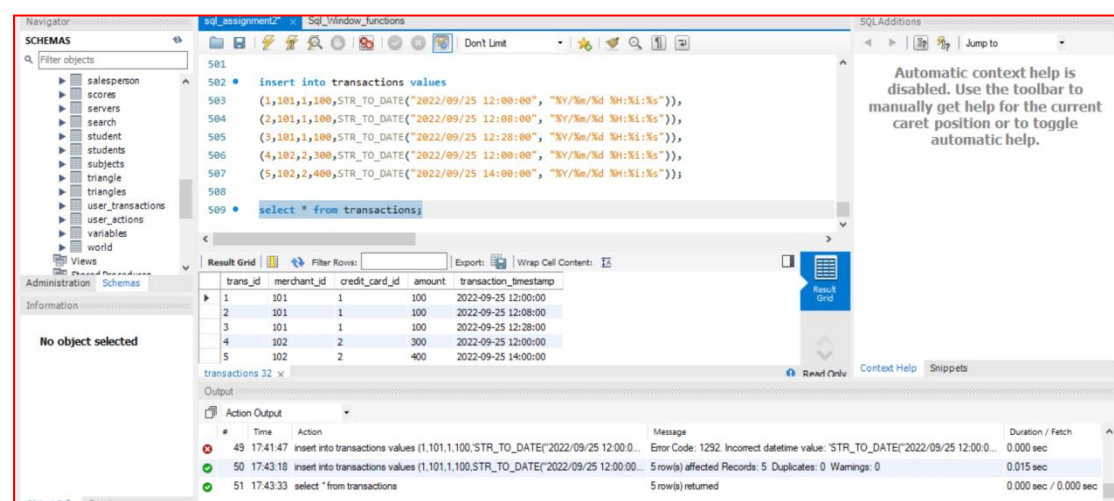
Object Info

Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

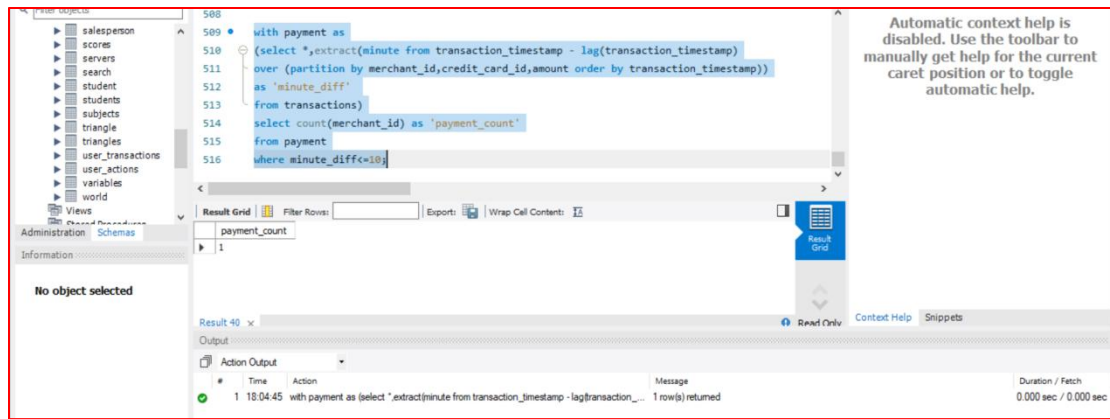
Q91) Sometimes, payment transactions are repeated by accident; it could be due to user error, API failure or a retry error that causes a credit card to be charged twice. Using the transactions table, identify any payments made at the same merchant with the same credit card for the same amount within 10 minutes of each other. Count such repeated payments.

Sol) At first create an empty table 'transactions' and then insert value in it. The values of the 'transaction_timestamp' column in the table is not in standard MYSQL format , hence I have converted the 5 values to the standard values using STR_TO_DATE function.



with payment as

(select *,extract(minute from transaction_timestamp
- lag(transaction_timestamp)
over (partition by merchant_id,credit_card_id,amount order
by transaction_timestamp))
as 'minute_diff'
from transactions)
select count(merchant_id) as 'payment_count'
from payment
where minute_diff<=10;

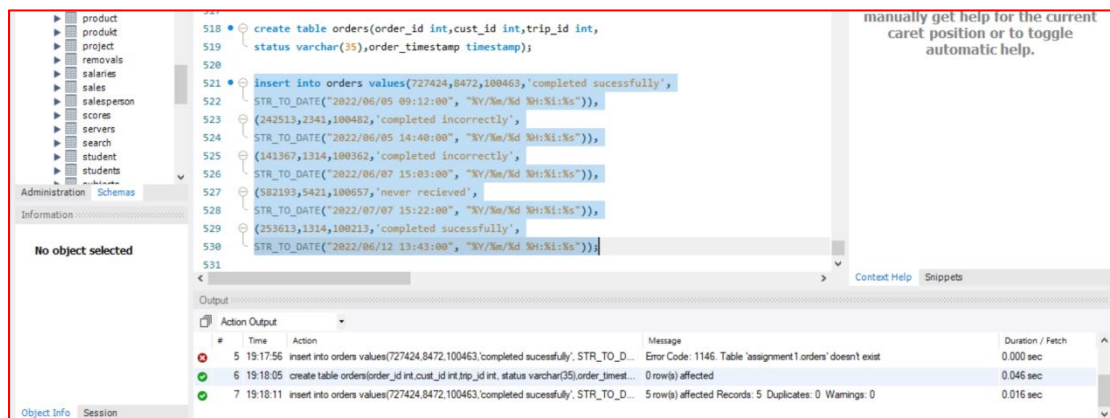


Q92)DoorDash's Growth Team is trying to make sure new users (those who are making orders in their first 14 days) have a great experience on all their orders in their 2 weeks on the platform. Unfortunately, many deliveries are being messed up because:

- the orders are being completed incorrectly (missing items, wrong order, etc.)
- the orders aren't being received (wrong address, wrong drop off spot)
- the orders are being delivered late (the actual delivery time is 30 minutes later than when the order was placed).

Write a query to find the bad experience rate in the first 14 days for new users who signed up in June 2022. Output the percentage of bad experience rounded to 2 decimal places.

Sol)At first create an empty table 'orders', 'trips' & 'customers' and then insert value in it using multi insert statement.

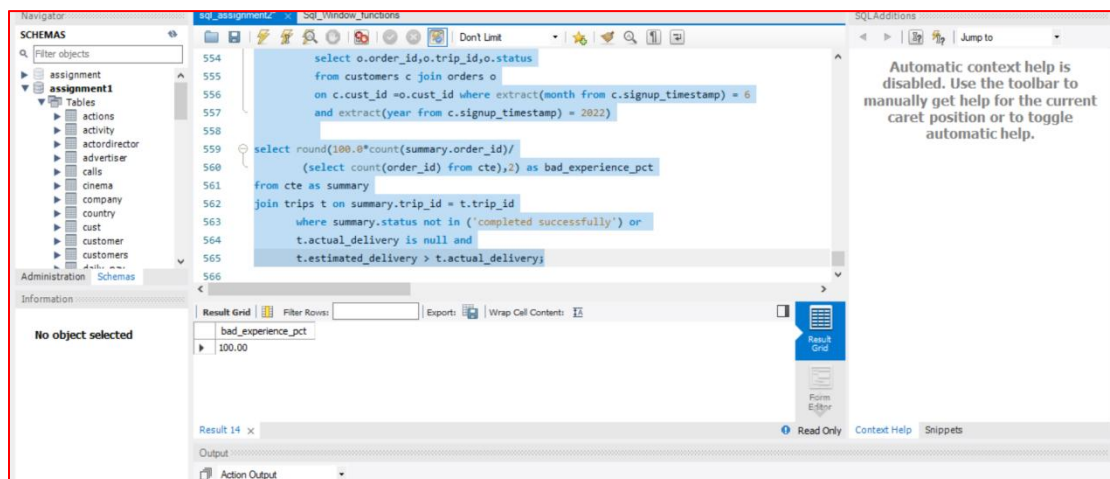


```

with cte as(
    select o.order_id,o.trip_id,o.status
    from customers c join orders o
    on c.cust_id =o.cust_id where extract(month from
c.signup_timestamp) = 6
    and extract(year from c.signup_timestamp) = 2022)

select round(100.0*count(summary.order_id)/
(select count(order_id) from cte),2) as bad_experience_pct
from cte as summary
join trips t on summary.trip_id = t.trip_id
where summary.status not in ('completed successfully') or
t.actual_delivery is null and
t.estimated_delivery > t.actual_delivery;

```



Q93) SAME QUESTION AS Q68

Q94) SAME QUESTION AS Q55

Q95) Write an SQL query to report the median of all the numbers in the database after decompressing the Numbers table. Round the median to one decimal point. The query result format is in the following example.

Sol) At first create an empty table 'Numbers' and then insert value in it using multi insert statement.

Q96) 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. Return the result table in any order. The query result format is in the following example

Sol) At first create two empty tables 'salary ' and 'employee' and then insert records in it using multi insert function.

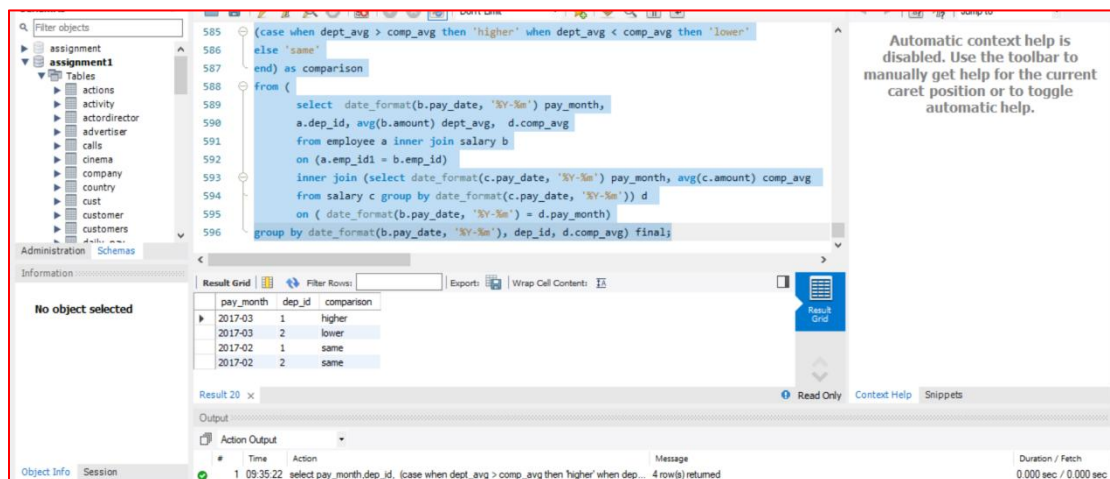


```
573 • create table Salary(id int primary key,emp_id int,
574   amount int,pay_date date);
575
576 • create table employee(emp_id1 int primary key,dep_id int);
577
578 • insert into Salary values(1,1,9000,'2017-03-31'),(2,2,6000,'2017-03-31'),
579   (3,3,10000,'2017-03-31'),(4,1,7000,'2017-02-28'),(5,2,6000,'2017-02-28'),
580   (6,3,6000,'2017-02-28');
581
582 • insert into employee values(1,1),(2,2),(3,2);
```

```

select pay_month,dep_id,
(case when dept_avg > comp_avg then 'higher' when
dept_avg < comp_avg then 'lower'
else 'same'
end) as comparison
from (
    select date_format(b.pay_date, '%Y-%m') pay_month,
    a.dep_id, avg(b.amount) dept_avg, d.comp_avg
    from employee a inner join salary b
    on (a.emp_id1 = b.emp_id)
    inner join (select date_format(c.pay_date, '%Y-%m')
pay_month, avg(c.amount) comp_avg
    from salary c group by date_format(c.pay_date,
'%Y-%m')) d
    on ( date_format(b.pay_date, '%Y-%m') =
d.pay_month)
group by date_format(b.pay_date, '%Y-%m'), dep_id,
d.comp_avg) final;

```



Q97) 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. Return the result table in any order. The query result format is in the following example.

Sol) The table used for this question is already created and ready for analysis purpose.

The screenshot shows a SQL IDE interface. On the left, there's a tree view with 'customer' and 'customers' folders. The main window displays a query: `select * from activity;`. Below the query, a 'Result Grid' shows 4 rows of data with columns: player_id, device_id, event_date, games_played. The data is as follows:

player_id	device_id	event_date	games_played
1	2	2016-03-01	5
1	2	2016-05-02	6
2	3	2017-06-25	1
3	1	2016-03-02	0

Below the result grid, an 'Action Output' table shows the execution details:

#	Time	Action	Message	Duration / Fetch
3	09:38:20	describe activity	4 row(s) returned	0.015 sec / 0.000 sec

```

select a.event_date as install_dt, count(a.player_id) as
installs,
round(count(b.player_id)/count(a.player_id), 2) as
Day1_retention
from (select player_id, MIN(event_date) as event_date from
activity
group by player_id) a
left join activity b on a.player_id = b.player_id and
a.event_date + 1=b.event_date
group by a.event_date;

```

The screenshot shows a SQL IDE interface. The query window contains the following SQL query:

```

select a.event_date as install_dt, count(a.player_id) as installs,
round(count(b.player_id)/count(a.player_id), 2) as Day1_retention
from (select player_id, MIN(event_date) as event_date from activity
group by player_id) a
left join activity b on a.player_id = b.player_id and a.event_date + 1=b.event_date
group by a.event_date;

```

The 'Result Grid' shows 3 rows of data with columns: install_dt, installs, Day1_retention. The data is as follows:

install_dt	installs	Day1_retention
2016-03-01	1	0.00
2017-06-25	1	0.00
2016-03-02	1	0.00

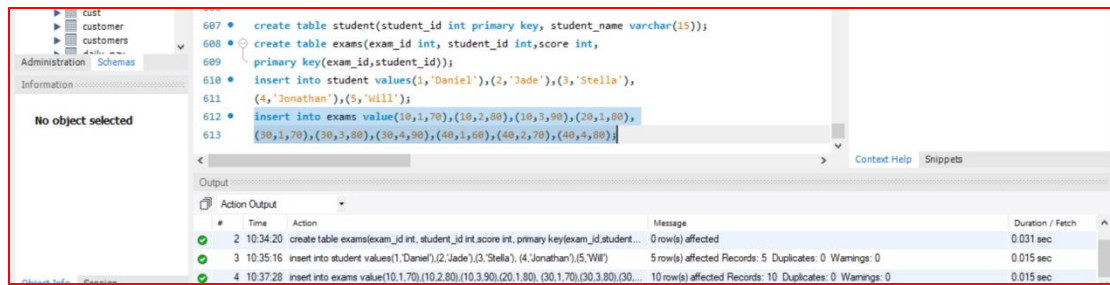
The 'Action Output' table shows the execution details:

#	Time	Action	Message	Duration / Fetch
10	10:17:49	select a.event_date as install_dt, count(a.player_id) as installs, round(count(b.player...	3 row(s) returned	0.015 sec / 0.000 sec
11	10:19:16	select a.event_date as install_dt, count(a.player_id) as installs, round(count(b.player...	3 row(s) returned	0.016 sec / 0.000 sec
12	10:19:39	select a.event_date as install_dt, count(a.player_id) as installs, round(count(b.player...	3 row(s) returned	0.000 sec / 0.000 sec

Q98) SAME QUESTION AS Q50

Q99) Write an SQL query to report the students (student_id, student_name) being quiet in all exams. Do not return the student who has never taken any exam. Return the result table ordered by student_id. The query result format is in the following example.

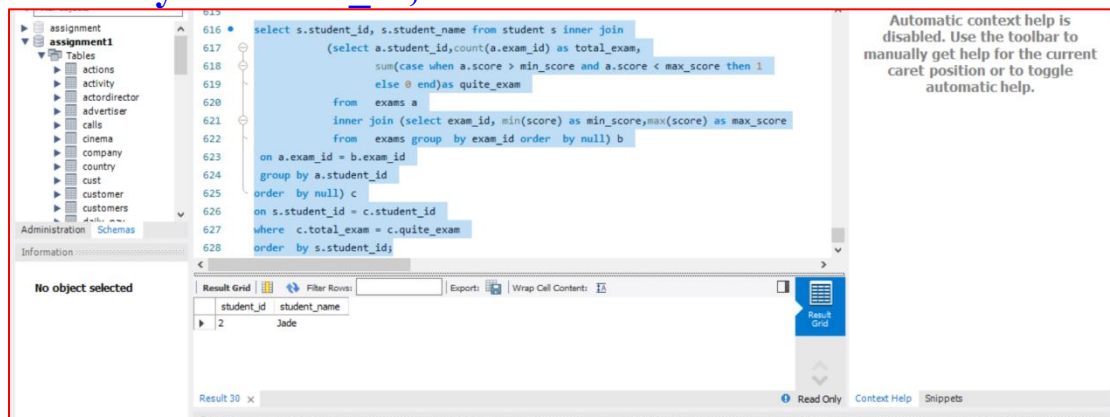
Sol) At first create two empty tables 'student' and 'exam' and then insert records in it using multi insert function.



```

select s.student_id, s.student_name from student s inner join
(select a.student_id,count(a.exam_id) as total_exam,
sum(case when a.score > min_score and a.score
< max_score then 1
else 0 end)as quite_exam
from exams a
inner join (select exam_id, min(score) as
min_score,max(score) as max_score
from exams group by exam_id order by null) b
on a.exam_id = b.exam_id
group by a.student_id
order by null) c
on s.student_id = c.student_id
where c.total_exam = c.quite_exam
order by s.student_id;

```



Q100) SAME QUESTION AS Q99