Chinmay Chawathe

Write a query to display the columns in a specific order, such as order date, salesman ID, order number, and purchase amount for all orders.

```
ord_no purch_amt ord_date customer_id salesman_id

-----------  -----------  -----------  ------------  ------------  ------------ ------------ -----

70001      150.5      2012-10-05 3005       5002
70009      270.65      2012-09-10 3001       5005
70002      65.26      2012-10-05 3002       5001
70004      110.5      2012-08-17 3009       5003
70007      948.5      2012-09-10 3005       5002
70005      2400.6      2012-07-27 3007       5001
70008      5760       2012-09-10 3002       5001
70010      1983.43     2012-10-10 3004       5006
70003      2480.4      2012-10-10 3009       5003
70012      250.45      2012-06-27 3008       5002
70011      75.29      2012-08-17 3003       5007
70013      3045.6      2012-04-25 3002       5001
```

```
Sql> create table orders(
ord_no int, purch_amt
int, ord_date DATE,
customer_id int, salesman_id
int
);

Sql> insert into  orders values
(70001,150.5,'2012-10-05',3005,5002),
(70009,270.65,'2012-09-10',3001,5005),
(70002,65.26,'2012-10-05',3002,5001),
(70004,110.5,'2012-08-17',3009,5003),
(70007,948.5,'2012-09-10',3005,5002),
(70005,2400.6,'2012-07-27', 3007,5001),
(70008,5760,'2012-09-10',3002,5001),
(70010,1983.43,'2012-10-10',3004,5006),
(70003,2480.4,'2012-10-10',3009,5003),
(70012,250.45,'2012-06-27',3008,5002), (70011,75.29,'2012-08-17',3003,5007),
(70013,3045.6,'2012-04-25',3002,5001);
```

Sql> select ord_date, salesman_id,ord_no, purch_amt from orders; **From the following table, write a SQL query to locate salespeople who live in the city of 'Paris'. Return salesperson's name, city.**

```
 salesman_id | name | city | commission
                  ...............         +______+_______+______
                          5001 | James Hoog | New York |      0.15
                          5002 | Nail Knite | Paris |      0.13
                          5005 | Pit Alex | London |   0.11
                          5006 | Mc Lyon | Paris          |  0.14
                          5007 | Paul Adam | Rome      |   0.13
                          5003 | Lauson Hen | San Jose |   0.12
```

```
Sql> create table sales_data(
salesman_id int primary key,
name     varchar(20),     city
varchar(20),
commission float
);

Sql> insert into sales_data values
(5001,'James Hoog','New York',0.15),
(5002,'Nail Knite','Paris',0.13),
(5005,'Pit Alex','London',0.11),
(5006,'Mc Lyon','Paris',0.14),
(5007,'Paul Adam','Rome',0.13) ,
(5003,'Lauson Hen','San Jose',0.12);

sql> select * from sales_data;

sql> select name,city from sales_data where city = 'Paris';
```

From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com.

```
PRO_ID PRO_NAME PRO_PRICE PRO_COM

-------- ---------------------------- ----------------------
                                         ------------------------

   101 Motherboard              3200.00        15
   102 Keyboard                  450.00        16
   103 ZIP drive                 250.00        14
   104 Speaker                   550.00        16
   105 Monitor                  5000.00        11
   106 DVD drive                 900.00        12
   107 CD drive                  800.00        12
   108 Printer                  2600.00        13
   109 Refill cartridge          350.00        13
   110 Mouse                     250.00        12
```

```
Sql> create table programing(
PRO_ID int primary key,
PRO_NAME not null,
PRO_PRICE float,
PRO_COM int
);

insert into programing values  (101,'Motherboard'3200.00,15),
(102,'Keyboard'450.00,16)
(103,'ZIP drive'250.00,14),
(104,'Speaker'550.00,16),
(105,'Monitor'5000.00,11),
(106,'DVD drive'900.00,12),
(107,'CD drive'800.00,12),
(108,'Printer'2600.00,13),
(109,'Refill cartridge',350.00,13),
(110,'Mouse',250.00,12);

Sql> select * from programing where Pro_price between 200 and 600;
```

From the following table, write a SQL query to find the items whose prices are higher than or equal to $550. Order the result by product price in descending, then product name in ascending.

Return pro_name and pro_price.

```
PRO_ID PRO_NAME PRO_PRICE PRO_COM
-------- ---------------------------- ---------------- ------------

   101 Motherboard          3200.00     15
   102 Keyboard             450.00      16
   103 ZIP drive            250.00      14
   104 Speaker              550.00      16
   105 Monitor              5000.00     11
   106 DVD drive            900.00      12
   107 CD drive             800.00      12
   108 Printer              2600.00     13
   109 Refill cartridge     350.00      13
   110 Mouse                250.00      12
```

```
Sql> select pro_name,pro_price from programing where Pro_price >= 550 order by pro_name
    asc,pro_price desc;
```

From the following table, write a SQL query to find details of all orders excluding those with ord_date equal to '2012-09-10' and salesman_id higher than 5005 or purch_amt greater than 1000.Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

```
Sql > select * from orders where ord_no <> '2012-09-10'  and salesman_id > 5005 or purch_amt >
1000;
```

Create the table world with your schema and find the below queries !

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	188681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000
Dominican Republic	Caribbean	48671	9445281	58898000000
China	Asia	9596961	1365370000	8358400000000

Chinmay Chawathe

Colombia	South America	1141748	47662000	369813000000
Comoros	Africa	1862	743798	616000000
Denmark	Europe	43094	5634437	314889000000
Djibouti	Africa	23200	886000	1361000000
Dominica	Caribbean	751	71293	499000000

```
create table world(
name varchar(20),
continent varchar(20),
area bigint,
population bigint,
gdp bigint);

insert into world values
('Afghanistan','Asia',652230,25500100,20343000000),
('Albania','Europe',28748,2831741,12960000000),
('Algeria','Africa',2381741,37100000,188681000000),
('Andorra','Europe',468,78115,3712000000),
('Angola','Africa',1246700,20609294,100990000000),
('Dominican Republic','Caribbean',48671,9445281,58898000000),
('China','Asia',9596961,1365370000,8358400000000),
('Colombia','South America',1141748,47662000,369813000000),
('Comoros','Africa',1862,743798,616000000),
('Denmark','Europe',43094,5634437,314889000000),
('Djibouti','Africa',23200,886000,1361000000),
('Dominica','Caribbean',751,71293,499000000);
```

1. Write a query to fetch which country has the highest population?

Sql> select name from world where population = (select max(population) from world) ;

2.write a query to fetch the name of the country which has the least gdp?

Sql> select name from world where gdp = (select min(gdp) from world) ; 3. Write a query to fetch the name of the country which ends with letter C?

Sql> select name from world where name REGEXP 'C$';

4.write a query to fetch the name of the country which starts with letter D? Sql> select name from world where name REGEXP '^d';

5.write query to fetch which continent has highest gdp?
Sql> select continent from world where gdp = (select max(gdp) from world) ;

6.Give the total GDP of Africa?
Sql> select distinct sum(gdp) over(partition by continent) from world where continent = 'Africa';

7.write a query to fetch the total population for each continent?

Sql> select continent,sum(population) over(partition by continent) population_by_continent from world group by continent;

8.For each relevant continent show the number of countries that has a population of at least 200000000?

Sql> select continent, count(name)no_of_countries from world where population >= 200000000 group by continent;

7. Problem statement: Suppose we have two table students and course

```
create table students(student_id int,
student_name varchar(60) not null,
city varchar(60) not null,
primary key(student_id));

create table course(student_id int,
course_name varchar(60) not null,
Marks int not null, primary
key(student_id),
foreign key(student_id) references students(student_id));


insert into students values(200,'John Doe','Delhi'), (210,'John
Doe','Delhi'),
(220,'Moon ethan','Rajasthan'),
(230,'Jessie','Bangalore'),
(240,'Benbrook','Bihar'),
(250,'Ethan','Bihar'),
(260,'Johnnie','Bangalore'),
(270,'Goh','Delhi'),(380,'John Doe','Delhi'),
(280,'Pavi','Delhi'),
(290,'Sanvi','Rajasthan'),
(300,'Navyaa','Bangalore'),
(310,'Ankul','Bihar'),
(311,'Hitanshi','Bihar'),
(312,'Aayush','Bangalore'),
(313,'Rian','Delhi');


insert into course values(200,'Datascience',75),
(210,'Datascience',75),
(220,'Dataanalyst',80),
(230,'Dataanalyst',80),
(240,'Dataanalyst',84),
(250,'Dataanalyst',50),
(260,'Datascience',80),
(270,'Datascience',99),
```

(380,'Datascience',45),
(280,'Datascience',78),
(290,'Dataanalyst',78),
(300,'Computer vision',90),
(310,'Computer vision',90),
(311,'Computer vision',75),
(312,'Computer vision',39)

Questions :

q1. write a query to fetch the names of the students having maximum marks in each course?

Sql> select student_name from students where student_id in (select student_id from (select student_id,Marks,course_name,dense_rank() over(partition by course_name order by Marks desc) as rk from course) temp where temp.rk = 1);

q2. write a query to fetch the names of the students having 3th highest marks from each course?

Sql>

q3. write a query to fetch the names of the students having minimum marks in each course?

select student_name from students where student_id in(select student_id from course where marks in (select distinct min(Marks) over(partition by course_name) as rk from course))

q4. write a query to fetch the names of the students having 4th least marks from each course?

Sql> select student_name from students where student_id in (
select student_id from
(select student_id,Marks,course_name,dense_rank() over(partition by course_name order by Marks) as rk from course) temp where temp.rk = 4);

or

Sql> with cte as (SELECT
S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks)marks_rank
FROM
students S JOIN course c on S.student_id = c.student_id)

select student_name from cte where marks_rank=4;

q5. write a query to fetch the city name of the students who have 2nd highest marks?

Sql> with cte as (SELECT
S.student_name,S.city,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks)marks_rank
FROM
students S JOIN course c on S.student_id = c.student_id)
select student_name,city from cte where marks_rank=2; q6.

write a query to fetch the count of each city? Sql> select city,count(*) from students group by city;

q7. write a query to fetch the names of the students who are from the same city?

```
Sql> select city,group_concat(DISTINCT student_name
        ORDER BY student_name
        SEPARATOR ';')  from students group by city order by city;
```

q8.write a query to fetch the names of students starting with 'A'?
Sql> select student_name from students where student_name REGEXP '^A';

q9.write a query to fetch the count of students' names having the same marks in each course?
q10.write a query to fetch the count of students from each city?
Sql> select city,count(student_id) from students group by city;

Hint : You must use Joins, Windows functions and CTE

8. Create a table below.

```
+ ................  ...........+           +
| Column Name | Type |
+ ................  ...........+           +
 | player_id | int      |
 | device_id | int      |
 | event_date | date      |
| games_played | int      |
+ ................  ...........+           +
```

(player_id, event_date) is the primary key of this table.
This table shows the activity of players of some games.
Each row is a record of a player who logged in and played a number of games (possibly 0) before logging out on someday using some device.

Write an SQL query to report the first login date for each player.
Return the result table in any order.
The query result format is in the following example.

Input:
Activity table:

```
+...........+...........+............+..............+
| 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          |
| 3      | 4      | 2018-07-03 | 5          |
+----------+----------+------------+--------------+
```

Output:

```
+-----------+-------------+
| player_id | first_login |
+-----------+-------------+
| 1         | 2016-03-01  |
| 2         | 2017-06-25  |
| 3         | 2016-03-02  |
+-----------+-------------+
```

```
Sql> create table Activity(
device_id         int,
event_date date,
games_played  int, player_id
int,
primary key (player_id, event_date)
);
```

```
Sql> insert into Activity
    (player_id,device_id,event_date,games_played)
    values
    (1,2,'2016-03-01',5),
    (1,2,'2016-05-02',6),
    (2,3,'2017-06-25',1),
    (3,1,'2016-03-02',0),
    (3,4,'2018-07-03',5);
```

```
Sql> select player_id,event_date as first_login from( select player_id,event_date,dense_rank()
    over(partition by player_id order by event_date) rk from Activity) a where a.rk = 1;
```

9. Create a table below.

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| product_id  | int     |
| low_fats    | enum    |
| recyclable  | enum    |
+-------------+---------+
```

product_id is the primary key for this table. low_fats is an ENUM of type ('Y', 'N') where 'Y' means this product is low fat and 'N' means it is not. recyclable is an ENUM of types ('Y', 'N') where 'Y' means this product is recyclable and 'N' means it is not.

Write an SQL query to find the ids of products that are both low fat and recyclable.
Return the result table in any order.
The query result format is in the following example.

```
Input:
Products table:
+ ............   ...........   .............+          +          +
| product_id | low_fats | recyclable |
+            .+..........+........-----+
.............
| 0          | Y        | N          |
| 1          | Y        | Y          |
| 2          | N        | Y          |
| 3          | Y        | Y          |
| 4          | N        | N          |
+            +          +
  .............   ...........
.................+ Output:
+.............+
| product_id |
+ .............+
| 1          |
| 3          |
+________+
```

```
Sql>  create table products
        (
        product_id int primary key,
        low_fats enum('Y', 'N') ,
        recyclable enum('Y', 'N')
        );
```

```
Sql> insert into products values
        (0,'Y','N'),
        (1,'Y','Y'),
        (2,'N','Y'),
        (3,'Y','Y'),
        (4,'N','N');
```

```
Sql> select product_id from (
select product_id, case
when low_fats = 'Y' and recyclable =
'Y' then 1 else 0 end as ans
from products)a where a.ans = 1;
```

10. Create a table below.

name	region	area	population	gdp
Afghanistan	South Asia	652225	26000000	

Chinmay Chawathe

Albania	Europe	28728	3200000	6656000000
Algeria	Middl e East	240000 0	32900000	7501200000 0
Andorra	Europe	468	64000	
...				

1. Select the statement that shows the sum of population of all countries i Sql>

   select sum(population) from world;

2. Select the statement that shows the number of countries with population smaller than 150000

   Sql> select (name) from world where population < 150000;

3. Select the list of core SQL aggregate functions

   Sql> select count(*),sum(population) total_population,avg(population) avg_population,min(population) min_population,max(population) max_population from world;

4. Select the result that would be obtained from the following code:
5. Select the statement that shows the average population of 'Poland', 'Germany' and 'Denmark'

   Sql> select avg(population) from world where name in ('Poland','Germany','Denmark');

6. Select the statement that shows the medium population density of each region

Sql> with cte as
     (select name,region,sum(population)over(partition by region order by name) total_population,   sum(area)over(partition by region order by name) total_area from world),
     cte2 as
    (select *,cte.total_population/cte.total_area as density from cte) select distinct cte2.region,round(sum(cte2.density)over(partition by cte2.region)/count(cte2.name)over(partition by cte2.region),2) as region_wise_population_density from cte2;

7. Select the statement that shows the name and population density of the country with the largest population

   Sql> with cte as
    (select name,region,population,sum(population)over(partition by region order by name) total_population,
    sum(area)over(partition by region order by name) total_area from world),
    cte2 as

Chinmay Chawathe

```
 (select *,cte.total_population/cte.total_area as density from cte)  select
cte2.name,cte2.density from cte2 where cte2.population = (select
max(cte2.population) from cte2);
```