1. 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

Solution:

select * from orders order by ord_no, purchase_amount, order_date, customer_id, salesman_id

2. 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

Solution:

select name,city from salesman where city='Paris';

3. 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

Solution:

select * from products where pro_price between 200.00 and 600.00

4. 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

Solution:

select pro_name,pro_price from products where pro_price>=500.00 order by pro_price desc,pro_name;

5. 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

Solution:

```
select * from orders where order_date<>'2012-09-10' and salesman_id>5005 and
purchase_amount>1000;
```

6. 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

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?

Solution:select country from world order by population desc limit 1;

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

Solution:select country from world order by gdp limit 1;

3. Write a query to fetch the name of the country which ends with letter C?

Solution:select country from world where country like '%c'

4.write a query to fetch the name of the country which starts with letter D?

Solution:select country from world where country like 'd%'

5.write query to fetch which continent has highest gdp?

Solution:select continent from world order by gdp desc limit 1;

6.Give the total GDP of Africa?

Solution:select sum(gdp) as Total_GDP from world where continent='Africa';

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

Solution:select continent,sum(population) from world group by continent;

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

Solution:select continent,count(country) 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?

Solution:

```
select s.*,c.*
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks desc) as marks_rank
from course) c on
s.student_id=c.student_id and c.marks_rank=1
```

;

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

Solution:

```
select s.*,c.*
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks desc) as marks_rank
from course) c on
s.student_id=c.student_id and c.marks_rank=3
```

;

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

Solution:

```
select s.*,c.*
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks) as min_marks
from course) c on
s.student_id=c.student_id and c.min_marks=1
```

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

Solution:

```
select s.*,c.*
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks) as min_marks
from course) c on
s.student_id=c.student_id and c.min_marks>=4
```

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

Solution:

```
select s.*,c.*
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks) as min_marks
from course) c on
s.student_id=c.student_id and c.min_marks=2;
```

q6. write a query to fetch the count of each city?

Soltuion:

```
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?

Solution:

```
select city, group_concat(distinct student_name) as students_from_respective_city from students group by city;
```

q8.write a query to fetch the names of students starting with 'A'?

Solution:

```
select student_name from students where student_name like 'A%';
```

q9.write a query to fetch the count of students' names having the same marks in each course?

Solution:

```
select s.student_id,c.course_name,c.marks,count(*)
from
students s
inner join
(select *,
        dense_rank() over(partition by course_name order by Marks desc) as min_marks
from course) c on
s.student_id=c.student_id
group by c.course_name,c.marks
having count(c.marks)>1;
```

q10.write a query to fetch the count of students from each city?

Solution:

```
select distinct city,count(distinct student_name) from students group by city;
```

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 |

Solution:

select player_id,min(event_date) as first_login from player group by player_id;

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 |

+-------------+

Solution:

select product_id from food_products

where low_fats='Y' and recyclable='Y';

10. Create a table below.

name region area population gdp

Afghanista

n

South Asia 652225 26000000

Albania Europe 28728 3200000 6656000000

Algeria Middle

East

240000

0

32900000 7501200000

0

Andorra Europe 468 64000

...

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

Solution:

select country_name,sum(population) from countries group by country_name order by country_name;

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

than 150000

Solution:

select count(c.country_counts) from

(select count(country_name) country_counts from countries where population>150000 group by country_name) c

3. Select the list of core SQL aggregate functions

Solution:

#List of aggregate functions in SQL

count()

sum()

min()

max()

avg()

4. Select the result that would be obtained from the following code:

#NOT GIVEN

5. Select the statement that shows the average population of 'Poland', 'Germany' and

'Denmark'

#NOT GIVEN

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

select region,avg(population) from countries group by region

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

with the largest population

Solution:

```
select c.country_name,c.region,c.population from
(select *,rank() over(partition by region order by population desc) as population_index
from countries) c
where c.population_index=1;
```