

```
create database sql_assignment; #creating database
use sql_assignment; #using database
SET SQL_SAFE_UPDATES = 0; #removing safe mode as I am using SQL workbench
```

### **#Solution 1**

```
#creating table for question 1
```

```
create table if not exists salesorder(
    ord_no int,
    purch_amt DOUBLE,
    ord_date date,
    customer_id int,
    salesman_id int );
```

```
#Inserting data into salesorder table
```

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

```
#this is the solution query :
```

```
select ord_date as order_date, salesman_id as salesman_ID, ord_no as order_number, purch_amt as
purchase_amount
from salesorder
```

```
order by ord_date; # we can order it by anything but it was not mentioned in the question,
# so I am doing the sort based on order date
```

order_date	salesman_ID	order_number	purchase_amount
2012-04-25	5001	70013	3045.6
2012-06-27	5002	70012	250.45
2012-07-27	5001	70005	2400.6
2012-08-17	5003	70004	110.5
2012-08-17	5007	70011	75.29
2012-09-10	5005	70009	270.65
2012-09-10	5002	70007	948.5
2012-09-10	5001	70008	5760
2012-10-05	5002	70001	150.5
2012-10-05	5001	70002	65.26
2012-10-10	5006	70010	1983.43
2012-10-10	5003	70003	2480.4

### #Solution 2:

```

create table if not exists salespeople(
    salesman_id int,
    name VARCHAR(100),
    city VARCHAR(100),
    commission DOUBLE);

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

select name as Sales_person_name , city
from salespeople
where city = 'Paris';

```

	Sales_person_name	city
▶	Nail Knite	Paris
	Mc Lyon	Paris

### #Solution 3

```
create table if not exists product(
```

```
    PRO_ID int,
```

```
    PRO_NAME VARCHAR(100),
```

```
    PRO_PRICE DOUBLE,
```

```
    PRO_COM int );
```

```
insert into product values(101, 'Motherboard', 3200.00, 15);
```

```
insert into product values(102, 'Keyboard', 450.00, 16);
```

```
insert into product values(103, 'ZIP drive', 250.00, 14);
```

```
insert into product values(104, 'Speaker', 550.00, 16);
```

```
insert into product values(105, 'Monitor', 5000.00, 11);
```

```
insert into product values(106, 'DVD drive', 900.00, 12);
```

```
insert into product values(107, 'CD drive', 800.00, 12);
```

```
insert into product values(108, 'Printer', 2600.00, 13);
```

```
insert into product values(109, 'Refill cartridge', 350.00, 13);
```

```
insert into product values(110, 'Mouse', 250.00, 12);
```

```
select PRO_ID, PRO_NAME, PRO_PRICE, PRO_COM
```

```
from product
```

```
where PRO_PRICE BETWEEN 200 AND 600 ;
```

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
102	Keyboard	450	16
103	ZIP drive	250	14
104	Speaker	550	16
109	Refill cartridge	350	13
110	Mouse	250	12

#### #SOLUTION 4

```
select PRO_NAME,PRO_PRICE  
from product  
where PRO_PRICE >= 550  
order by PRO_PRICE DESC,PRO_NAME ;
```

PRO_NAME	PRO_PRICE
Monitor	5000
Motherboard	3200
Printer	2600
DVD drive	900
CD drive	800
Speaker	550

#### #SOLUTION 5

```
SELECT ord_no, purch_amt, ord_date, customer_id ,salesman_id  
FROM salesorder  
where ord_date != '2012-09-10'  
and ( salesman_id > 5005 or purch_amt >1000)  
order by salesman_id;
```

ord_no	purch_amt	ord_date	customer_id	salesman_id
70005	2400.6	2012-07-27	3007	5001
70013	3045.6	2012-04-25	3002	5001
70003	2480.4	2012-10-10	3009	5003
70010	1983.43	2012-10-10	3004	5006
70011	75.29	2012-08-17	3003	5007

#### #Solution 6

```
create table if not exists world(  
name VARCHAR(100),  
continent VARCHAR(100),  
area DOUBLE,  
population DOUBLE,  
gdp DOUBLE  
);
```

```

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

```

1. Select name from world order by population desc limit 1;

name
China

2. Select name from world order by gdp limit 1;

name
Dominica

3. Select name from world

WHERE name LIKE '%C' ;

name
Dominican Republic

4. Select name from world

WHERE name LIKE 'D%' ;

name
Dominican Republic
Denmark
Djibouti
Dominica

5. Select continent from world order by gdp desc limit 1;

continent
Asia

6. Select SUM(gdp) as Total\_grp\_of\_Africa

FROM world

WHERE continent = 'Africa';

Total_grp_of_Africa
291648000000

7. Select continent,sum(population) as total\_population\_by\_continent

FROM world

group by continent;

continent	total_population_by_continent
Asia	1390870100
Europe	8544293
Africa	59339092
Caribbean	9516574
South America	47662000

8. Select continent, COUNT(name) as  
number\_of\_countries\_having\_more\_than\_2\_billion\_population

FROM world

where population >=2000000000

group by continent;

continent	number_of_countries_having_more_than_2_billion
Asia	1

### #Soulution 7

1.

2.

select s.student\_name,c.course\_name,c.marks as third\_highest\_mark

from

(select student\_id,course\_name,marks, row\_number() over(partition by course\_name order by marks desc) as third\_highest\_mark from course group by course\_name,marks)c

inner join (select student\_id,student\_name,row\_number() over(partition by student\_id) as st from students group by student\_id)s

on c.student\_id=s.student\_id where c.third\_highest\_mark=3;

	student_name	course_name	third_highest_mark
▶	Aayush	Computer vision	39
	Sanvi	Dataanalyst	78
	Pavi	Datascience	78

3.

select s.student\_name,c.course\_name,c.marks as minimum\_mark

from

(select student\_id,course\_name,marks, row\_number() over(partition by course\_name order by marks) as minimum\_mark from course group by course\_name,marks)c

inner join (select student\_id,student\_name,row\_number() over(partition by student\_id) as st from students group by student\_id)s

on c.student\_id=s.student\_id where c.minimum\_mark=1;

	student_name	course_name	minimum_mark
▶	Aayush	Computer vision	39
	Ethan	Dataanalyst	50
	John Doe	Datascience	45

4.

select s.student\_name,c.course\_name,c.marks as fourth\_least\_mark

from

(select student\_id,course\_name,marks, row\_number() over(partition by course\_name order by marks desc) as fourth\_least\_mark from course group by course\_name,marks)c

inner join (select student\_id,student\_name,row\_number() over(partition by student\_id) as st from students group by student\_id)s

on c.student\_id=s.student\_id where c.fourth\_least\_mark=4;

	student_name	course_name	fourth_least_mark
▶	Ethan	Dataanalyst	50
	John Doe	Datascience	75

5.

6.

select city,count(city) as count\_of\_each\_city from students group by city;

	city	count_of_each_city
▶	Delhi	6
	Rajasthan	2
	Bangalore	4
	Bihar	4

7.

select city, GROUP\_CONCAT(student\_name) as students\_from\_same\_city from students group by city;

	city	students_from_same_city
▶	Bangalore	Jessie,Johnnie,Navyaa,Aayush
	Bihar	Benbrook,Ethan,Ankul,Hitanshi
	Delhi	John Doe,John Doe,Goh,Pavi,Rian,John Doe
	Rajasthan	Moon ethan,Sarvi

8.

select student\_name from students where student\_name like'A%';

	student_name
▶	Ankul
	Aayush

9.

10.

select count(student\_name),city from students group by city;

	count(student_name)	city
▶	6	Delhi
	2	Rajasthan
	4	Bangalore
	4	Bihar

### #Soulution 8

create table activity(player\_id int,device\_id int,event\_date date,games\_played int ,PRIMARY KEY (player\_id, event\_date));

insert into activity 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);

select a.player\_id,a.event\_date as first\_login



```

from(select *, row_number()over(partition by player_id order by event_date) as first_login
from activity) a
where a.first_login=1;

```

	player_id	first_login
▶	1	2016-03-01
	2	2017-06-25
	3	2016-03-02

### #Soulution 9

```

create table products(product_id int NOT NULL,
low_fats enum('Y','N'),recyclable enum('Y','N'),PRIMARY KEY (product_id));

```

```

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

```

```

select product_id from products where low_fats='Y' and recyclable='Y';

```

	product_id
▶	1
	3

### #Soulution 10,

1. select sum(population) as Total\_Population from world1;

	Total_Population
▶	62164000

2. select count(name) from world1 where population<=150000 group by name;

	count(name)
▶	1

3. ,4. ,5. ,6., and 7 has been skipped due to insufficient data and question is not clear at all (looks like mcq type questions with missing options).