**SQL PRACTICE PROBLEMS**

**Q1 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**

**ANS.**

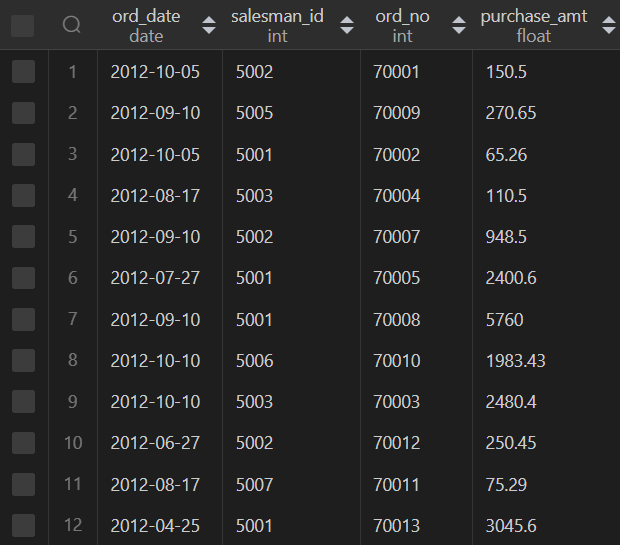
select ord\_date,

salesman\_id,

ord\_no,

purchase\_amt

from order\_details;

****

**Q2. 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**

**Ans.**

select name,

city

from salesperson\_details

where city like 'paris';

****

**Q3a. 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**

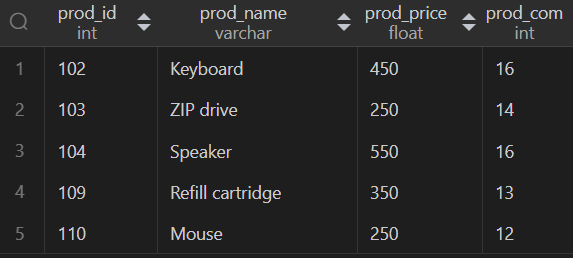
**110 Mouse 250.00 12**

**ANS.**

select \*

from product\_details

 where prod\_price between 200 and 600;

****

**Q3b.From the above 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.**

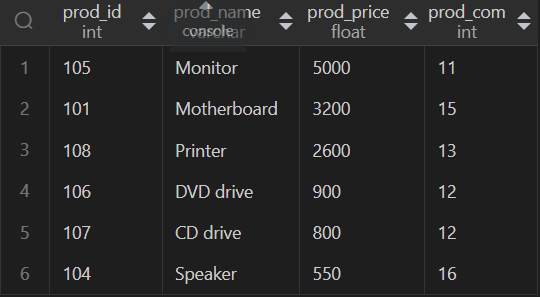
**ANS.**

select \*

from product\_details

 where prod\_price >=550

order by prod\_price DESC, prod\_name ASC;



**Q4. 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**

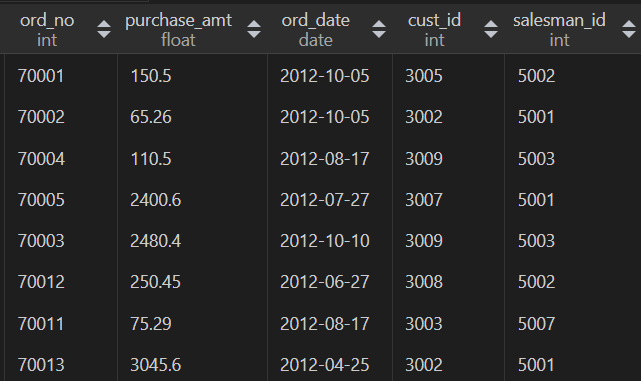
**70013 3045.6 2012-04-25 3002 5001**

**ANS.**

select \*

from order\_details

where ord\_date  not in ('2012-09-10') and( not salesman\_id >5005 or not purchase\_amt >1000) ;

****

**Q5. 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**

**Q5a) Write a query to fetch which country has the highest population?**

**Ans.**

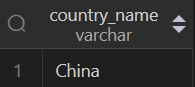
select country\_name

from country\_details

where population\_country= (select

       max(population\_country)

from country\_details);



**Q5b) write a query to fetch the name of the country which has the least gdp?**

**ANS.**

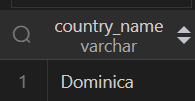
select country\_name

from country\_details

where gdp= (select

       MIN(gdp)

from country\_details);

****

**Q5c) Write a query to fetch the name of the country which ends with letter C?**

**Ans.**

select \*

from country\_details

where country\_name like '%c';

****

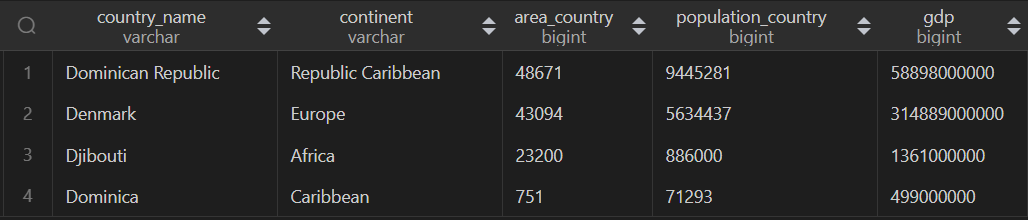
**Q5d) write a query to fetch the name of the country which starts with letter D?**

**Ans.**

select \*

from country\_details

where country\_name like 'd%';

****

**Q5e) write query to fetch which continent has highest gdp?**

**Ans.**

SELECT

        continent,

        sum(gdp) as total\_continent\_gdp

from country\_details

group by continent

having total\_continent\_gdp =

        (select max(tmp.total\_gdp)

        from

                (select continent,

                    sum(gdp) as total\_gdp

                from country\_details

                group by continent) tmp

        );

**2nd way of solving—**

select continent,

        sum(gdp) as total\_gdp

from country\_details

group by continent

order by sum(gdp) desc

limit 1;

****

**Q5f) Give the total GDP of Africa?**

**Ans.**

select 'Africa GDP' as Continent,

        sum(gdp) as total\_gdp\_africa

from country\_details

where continent like 'africa';

****

**Q5g) write a query to fetch the total population for each continent?**

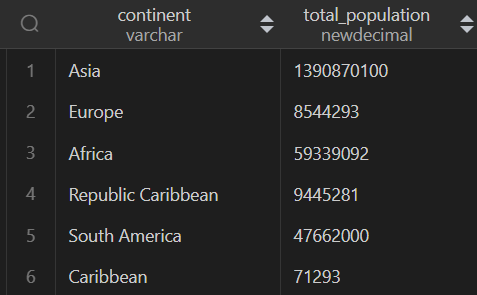
**Ans.**

select continent,

       sum(population\_country) as total\_population

from country\_details

group by continent;

****

**Q5h) For each relevant continent show the number of countries that has a population of at least 200000000?**

**Ans.**

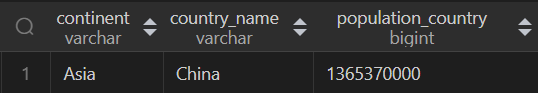
select continent,

       country\_name,

       population\_country

from country\_details

where population\_country >=200000000;

****

**Q6 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)**

**Q6a) write a query to fetch the names of the students having maximum marks in each course?**

**Ans.**

SELECT  s.student\_name,

        c.course\_name,

        c.marks

from students s

inner join course c

on s.student\_id = c.student\_id

and (c.course\_name,c.marks) in

(

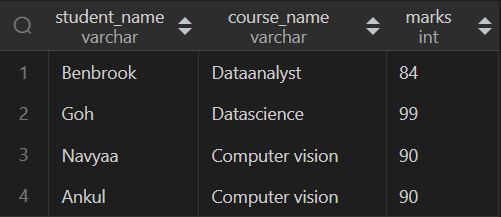
        select course\_name,

            max(marks)

        from course

        group by course\_name)

;

****

**Q6b) write a query to fetch the names of the students having 3th highest marks from each course?**

**Ans.**

with ranking as(

    select \*,

           dense\_rank() over(partition by course\_name order by marks desc) as rnk from course

)

select s.student\_name,

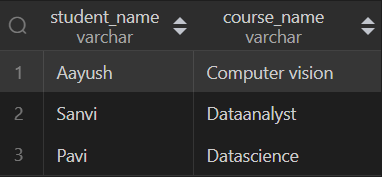
       r.course\_name

from ranking r

join students s

ON r.student\_id = s.student\_id

where rnk=3;

****

**Q6c) write a query to fetch the names of the students having minimum marks in each course?**

**Ans.**

select s.student\_name

from students s

inner join course c

on s.student\_id = c.student\_id

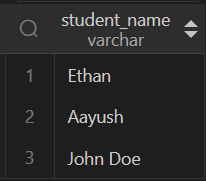
where (c.course\_name, c.marks) in

(select course\_name ,

       min(marks) as min\_marks

from course

group by course\_name);

****

**Q6d) write a query to fetch the names of the students having 4th least marks from each course?**

**Ans.**

with 4th\_least as (

    select \*,

           dense\_rank() over(partition by course\_name order by marks asc) as Rnk

    from course

)

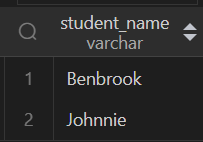
select s.student\_name

from 4th\_least l

inner join students s

on l.student\_id = s.student\_id

where Rnk = 4;

****

**Q6e) write a query to fetch the city name of the students who have 2nd highest marks in each course?**

**Ans.**

**Q6f) write a query to fetch the count of each city?**

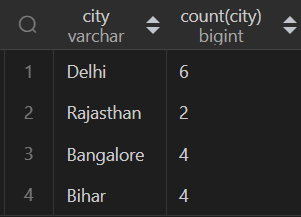
**Ans.**

**select  city,**

**count(city)**

**from students**

**group by city;**

****

**Q6g) write a query to fetch the names of the students who are from the same city?**

**Ans.**

select student\_id,

       student\_name

from students

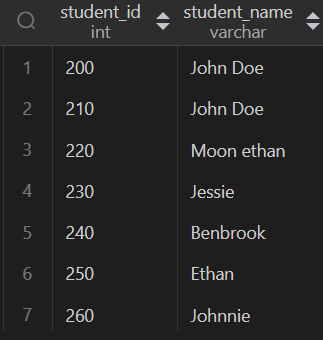
where city IN

(select city

from students

group by city

having count(\*)>1);

****

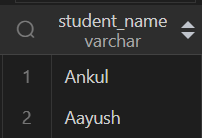
**Q6h) write a query to fetch the names of students starting with 'A'?**

**Ans.**

select student\_name

from students

where student\_name like 'A%';

****

**Q6i) write a query to fetch the count of student's names having the same marks in each course?**

**Ans.**

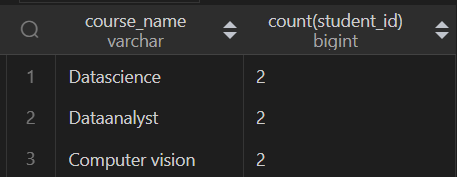
select course\_name,

count(student\_id)

from course

group by course\_name, marks

having count(marks) >=2;

****

**Q6j) write a query to fetch the count of students from each city?**

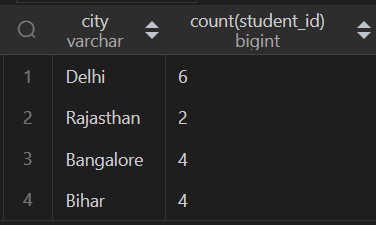
**Ans.**

**select city,**

**count(student\_id)**

**from students**

**group by city;**

****

**Q7) 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 |**

**+-----------+-------------+**

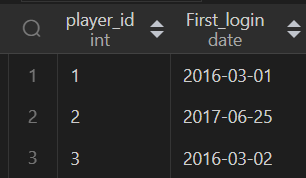
**Ans.**

select player\_id,

       min(event\_date) as First\_login

from gaming\_details

group by player\_id;

****

**Q8) 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 |**

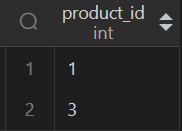
**| 3 |**

**+-------------+Ans.**

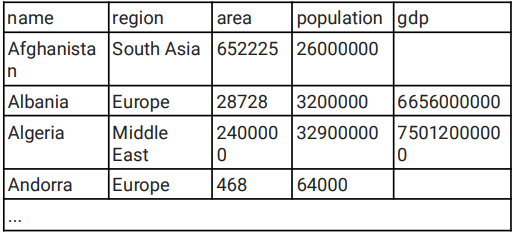
select product\_id

from products

where low\_fats ='Y' and recyclable = 'Y';

****

**Q9) Create a table below.**

**...**

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

select sum(population\_country) as Total\_Population

from country\_details;

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

**than 150000**

select count(name) as Countries

from country\_details

where population\_country <150000;

**3. Select the list of core SQL aggregate functions**

**Max(), Min(), Sum(), Avg(), Count()**

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

**'Denmark'**

select avg(population\_country) as Total\_Population

from country\_details

where country in ('Poland', 'Germany','Denmark');

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

with population\_density as(

    select \*,

    1.0\*population\_country/area as population\_density

    from country\_details

),

median\_population\_density as (

    select \*,

            row\_number() over(partition by region order by population\_density asc) as rnk\_asc,

            row\_number() over(partition by region order by population\_density desc) as rnk\_desc

    from population\_density

)

select  region,

        avg(population\_desnsity) as median\_population\_density

from median\_population\_density

where abs(rnk\_asc-rnk\_desc)<=1

group by region;

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

**with the largest population**

select country,

       1.0\*population\_country/area as population\_density

from country\_details

where population\_country in

(select max(population\_country)

from country\_details);