

SQL PRACTISE QUESTIONS

1.Top Travellers

Table: Users

Column name	Type
id	Int
name	varchar

id is the column with unique values for this table.

name is the name of the user.

Table: Rides

Column name	Type
id	Int
user_id	Int
distance	Int

id is the column with unique values for this table.user_id is the id of the user who traveled the distance "distance".

Write a solution to report the distance traveled by each user?Return the result table ordered by travelled_distance in **descending order**, if two or more users traveled the same distance, order them by their name in **ascending order**.

Ans:

```
SELECT name,COALESCE(sum(distance),0) as travelled_distance FROM Users u
```

```
LEFT JOIN Rides r on u.id=r.user_id
```

```
GROUP BY u.id
```

```
ORDER BY travelled_distance DESC,name ASC;
```

2.Patients With a Condition

Table: Patients

Column name	Type
patient_id	Int
Patient_name	varchar
conditions	varchar

patient_id is the primary key (column with unique values) for this table.

'conditions' contains 0 or more code separated by spaces.

This table contains information of the patients in the hospital.

Write a solution to find the patient_id, patient_name, and conditions of the patients who have Type I Diabetes. Type I Diabetes always starts with DIAB1 prefix.Has to handle both starting with DIAB and DIAB in middle

SQL PRACTISE QUESTIONS

Ans:

```
SELECT patient_id,patient_name ,conditions FROM patients
```

```
WHERE conditions LIKE '% DIAB1%'OR conditions LIKE 'DIAB1%';
```

3.Fix Names in a Table

Table:Users

Column name	Type
Id	Int
name	varchar

user_id is the primary key (column with unique values) for this table.

This table contains the ID and the name of the user. The name consists of only lowercase and uppercase characters. Write a solution to fix the names so that only the first character is uppercase and the rest are lowercase. Return the result table ordered by user_id.

Ans:

```
SELECT user_id,
```

```
CONCAT(UPPER(SUBSTRING(name,1,1)),LOWER(SUBSTRING(name,2))) as name FROM Users
```

```
ORDER BY user_id
```

4.Reformat Department Table

Table:Department

Column name	Type
Id	Int
revenue	Int
month	varchar

In SQL, (id, month) is the primary key of this table. The table has information about the revenue of each department per month. The month has values in ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"].

Reformat the table such that there is a department id column and a revenue column **for each month**.

Ans:

```
SELECT
```

```
id,
```

```
MAX(CASE WHEN month = 'Jan' THEN revenue END) AS Jan_Revenue,
```

```
MAX(CASE WHEN month = 'Feb' THEN revenue END) AS Feb_Revenue,
```

```
MAX(CASE WHEN month = 'Mar' THEN revenue END) AS Mar_Revenue,
```

SQL PRACTISE QUESTIONS

```
MAX(CASE WHEN month = 'Apr' THEN revenue END) AS Apr_Revenue,  
MAX(CASE WHEN month = 'May' THEN revenue END) AS May_Revenue,  
MAX(CASE WHEN month = 'Jun' THEN revenue END) AS Jun_Revenue,  
MAX(CASE WHEN month = 'Jul' THEN revenue END) AS Jul_Revenue,  
MAX(CASE WHEN month = 'Aug' THEN revenue END) AS Aug_Revenue,  
MAX(CASE WHEN month = 'Sep' THEN revenue END) AS Sep_Revenue,  
MAX(CASE WHEN month = 'Oct' THEN revenue END) AS Oct_Revenue,  
MAX(CASE WHEN month = 'Nov' THEN revenue END) AS Nov_Revenue,  
MAX(CASE WHEN month = 'Dec' THEN revenue END) AS Dec_Revenue  
FROM  
  
Department  
  
GROUP BY id
```

5.Products Sold In Only One Quarter

Table:Product

Column name	Type
product_id	Int
product_name	varchar
unit_price	Int

product_id is the primary key (column with unique values) of this table.

Each row of this table indicates the name and the price of each product.

Table: Sales

Column name	Type
seller_id	Int
product_id	Int
buyer_id	Int
Sale_date	date
quantity	Int
price	Int

This table can have duplicate rows.

product_id is a foreign key (reference column) to the Product table.

Each row of this table contains some information about one sale.

Write a solution to report the **products** that were **only** sold in the first quarter of 2019. That is, between 2019-01-01 and 2019-03-31 inclusive.

SQL PRACTISE QUESTIONS

Ans:

```
SELECT product_id,product_name FROM product p  
JOIN sales s using(product_id)  
GROUP BY product_id  
HAVING MIN(sale_date)>='2019-01-01' and MAX(sale_date)<='2019-03-31'
```

6.GAME PLAN ANALYSIS

Table: Activity

Column name	Type
player_id	Int
device_id	Int
event_id	date
games_played	Int

(player_id, event_date) is the primary key (combination of columns with unique values) 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 a solution to find the **first login date** for each player.

Ans:

```
SELECT player_id,MIN(event_date) as first_login FROM Activity  
GROUP BY player_id
```

7.Second Highest Salary

Table:Employee

Column name	Type
Id	Int
salary	Int

id is the primary key (column with unique values) for this table.

Each row of this table contains information about the salary of an employee. Write a solution to find the second highest **distinct** salary from the Employee table. If there is no second highest salary, return null

```
SELECT COALESCE((SELECT MAX(salary) FROM Employee  
WHERE salary <(SELECT MAX(salary) FROM Employee)),null)  
AS SecondHighestSalary;
```

SQL PRACTISE QUESTIONS

8.Department wise Highest Salary

Table:Employee

Column name	Type
id	Int
name	varchar
salary	Int
departmentId	Int

id is the primary key (column with unique values) for this table.

departmentId is a foreign key (reference columns) of the ID from the Department table.

Table: Department

Column name	Type
id	Int
name	varchar

id is the primary key (column with unique values) for this table. It is guaranteed that department name is not NULL.Each row of this table indicates the ID of a department and its name.

Write a solution to find employees who have the highest salary in each of the departments?

Ans:

```
SELECT Department,employee,salary FROM (
```

```
SELECT d.name AS Department,
```

```
       e.name as employee,salary,
```

```
       RANK() OVER(PARTITION BY departmentId ORDER BY salary desc) as salary_rnk
```

```
FROM employee e
```

```
JOIN Department d ON e.departmentId=d.id) s
```

```
where salary_rnk=1;
```

9.Cancellation Rate

Table: Trips

Column name	Type
id	Int
client_id	Int
driver_id	Int
city_id	Int
status	enum
request_at	varchar

SQL PRACTISE QUESTIONS

id is the primary key (column with unique values) for this table.

The table holds all taxi trips. Each trip has a unique id, while client_id and driver_id are foreign keys to the users_id at the Users table.

Status is an ENUM (category) type of ('completed', 'cancelled_by_driver', 'cancelled_by_client').

Table:Users

Column name	Type
users_id	Int
banned	enum
role	enum

users_id is the primary key (column with unique values) for this table.

The table holds all users. Each user has a unique users_id, and role is an ENUM type of ('client', 'driver', 'partner').

banned is an ENUM (category) type of ('Yes', 'No').

The **cancellation rate** is computed by dividing the number of canceled (by client or driver) requests with unbanned users by the total number of requests with unbanned users on that day.

Write a solution to find the **cancellation rate** of requests with unbanned users (**both client and driver must not be banned**) each day between "2013-10-01" and "2013-10-03". Round Cancellation Rate to **two decimal** points.

Ans:

```
SELECT request_at AS Day,
       ROUND(SUM(CASE
                WHEN status IN ('cancelled_by_client', 'cancelled_by_driver')
                THEN 1
                ELSE 0
              END)/count(*),2) AS 'Cancellation Rate'
from trips t
LEFT JOIN users u1 on t.client_id=u1.users_id
LEFT JOIN users u2 on t.driver_id=u2.users_id
WHERE u1.banned='no' and u2.banned='no'
AND t.request_at BETWEEN "2013-10-01" AND "2013-10-03"
GROUP BY request_at
ORDER BY request_at
```

SQL PRACTISE QUESTIONS

10. Actors and Directors Who Cooperated At Least Three Times

Table: ActorDirector

Column name	Type
actor_id	Int
director_id	varchar
timestamp	Int

timestamp is the primary key (column with unique values) for this table.

Write a solution to find all the pairs (actor_id, director_id) where the actor has cooperated with the director at least three times.

Ans:

```
SELECT actor_id,director_id FROM ActorDirector
```

```
GROUP BY actor_id,director_id
```

```
HAVING count(timestamp)>=3
```

11. Employees Earning More Than Their Managers

Table: Employee

Column name	Type
Id	Int
name	varchar
salary	Int
managerId	Int

id is the primary key (column with unique values) for this table.

Each row of this table indicates the ID of an employee, their name, salary, and the ID of their manager. Write a solution to find the employees who earn more than their managers?

Ans:

```
SELECT e1.name AS employee FROM Employee e1
```

```
JOIN Employee e2 ON e1.managerid=e2.id
```

```
WHERE e1.salary > e2.salary
```

12. Customers Who Never Order

Table: Customers

Column name	Type
Id	Int
name	varchar

SQL PRACTISE QUESTIONS

id is the primary key (column with unique values) for this table.

Each row of this table indicates the ID and name of a customer.

Table: Orders

Column name	Type
id	Int
customerId	Int

id is the primary key (column with unique values) for this table.

customerId is a foreign key (reference columns) of the ID from the Customers table.

Each row of this table indicates the ID of an order and the ID of the customer who ordered it.

Write a solution to find all customers who never order anything?

Ans:

```
SELECT name AS Customers FROM Customers c
```

```
LEFT JOIN Orders o on c.id=o.customerid
```

```
WHERE o.customerid IS NULL;
```

13.Department Top Three Salaries

Table: Employee

Column name	Type
id	Int
name	varchar
salary	Int
departmentId	Int

id is the primary key (column with unique values) for this table.

departmentId is a foreign key (reference column) of the ID from the Department table.

Each row of this table indicates the ID, name, and salary of an employee. It also contains the ID of their department.

Table: Department

Column name	Type
id	Int
name	varchar

id is the primary key (column with unique values) for this table.

Each row of this table indicates the ID of a department and its name.

SQL PRACTISE QUESTIONS

A company's executives are interested in seeing who earns the most money in each of the company's departments. A **high earner** in a department is an employee who has a salary in the **top three unique** salaries for that department. Write a solution to find the employees who are **high earners** in each of the departments?

Ans:

```
WITH sal_rnk AS(
SELECT
    d.name AS Department,
    e.name AS Employee,
    Salary,
    DENSE_RANK() OVER (PARTITION BY d.name ORDER BY e.salary DESC) AS Salary_rnk
FROM Employee e
JOIN Department d ON e.departmentid = d.id
)
SELECT Department,Employee,Salary FROM sal_rnk
WHERE Salary_rnk<=3;
```

14.Customer Who Visited but Did Not Make Any Transactions

Table: Visits

Column name	Type
visit_id	Int
customer_id	Int

visit_id is the column with unique values for this table.

This table contains information about the customers who visited the mall.

Table: Transactions

Column name	Type
transaction_id	Int
visit_id	Int
amount	Int

transaction_id is column with unique values for this table.

This table contains information about the transactions made during the visit_id.

Write a solution to find the IDs of the users who visited without making any transactions and the number of times they made these types of visits?

SQL PRACTISE QUESTIONS

Ans:

```
SELECT customer_id,count(v.visit_id) AS count_no_trans
FROM visits v
LEFT JOIN Transactions t ON v.visit_id=t.visit_id
WHERE t.transaction_id IS NULL
GROUP BY customer_id
```

15.Customer Placing the Largest Number of Orders

Table: Orders

Column name	Type
order_number	Int
customer_number	Int

order_number is the primary key (column with unique values) for this table.

Write a solution 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.

Ans:

```
SELECT customer_number FROM orders
GROUP BY customer_number
Order BY count(order_number) DESC
LIMIT 1;
```

16.Classes More Than 5 Students

Table: Courses

Column name	Type
student	varchar
class	varchar

(student, class) is the primary key (combination of columns with unique values) for this table.Each row of this table indicates the name of a student and the class in which they are enrolled. Write a solution to find all the classes that have **at least five students**.

Ans:

```
SELECT class from Courses
GROUP BY class
HAVING count(student)>=5;
```

SQL PRACTISE QUESTIONS

17. Consecutive Numbers

Table: Logs

Column name	Type
Id	Int
num	varchar

Find all numbers that appear at least three times consecutively?

Ans:

```
SELECT DISTINCT num AS ConsecutiveNums
FROM (
    SELECT num,
           LAG(num, 1) OVER (ORDER BY id) AS prev_num_1,
           LAG(num, 2) OVER (ORDER BY id) AS prev_num_2
    FROM Logs
) lagged_no
WHERE num = prev_num_1 AND num = prev_num_2;
```

18. Duplicate Emails

Table: Person

Column name	Type
id	Int
email	varchar

id is the primary key (column with unique values) for this table.

Each row of this table contains an email. The emails will not contain uppercase letters.

Write a solution to report all the duplicate emails. Note that it's guaranteed that the email field is not NULL.

Ans:

```
SELECT email from Person
GROUP BY email HAVING count(email)>1;
```

SQL PRACTISE QUESTIONS

19.The Latest Login in 2020

Table: Logins

Column name	Type
user_id	Int
timestamp	datetime

(user_id, time_stamp) is the primary key (combination of columns with unique values) for this table.

Each row contains information about the login time for the user with ID user_id.

Write a solution to report the **latest** login for all users in the year 2020. Do **not** include the users who did not login in 2020?

Ans:

```
SELECT user_id,max(time_stamp) AS last_stamp FROM logins
```

```
WHERE year(time_stamp)='2020'
```

```
GROUP BY user_id
```

20.Not Boring Movies

Table: Cinema

Column name	Type
id	Int
movie	varchar
description	varchar
rating	float

id is the primary key (column with unique values) for this table.

Each row contains information about the name of a movie, its genre, and its rating.

rating is a 2 decimal places float in the range [0, 10]

Write a solution to report the movies with an odd-numbered ID and a description that is not "boring".Return the result table ordered by rating **in descending order**.

Ans:

```
SELECT id,movie,description,rating FROM cinema
```

```
WHERE description!='boring'
```

```
AND id%2!=0
```

```
ORDER BY rating DESC;
```