**SQL Exercise**

1. Recyclable and Low Fat Products

SELECT product\_id

FROM Products

WHERE low\_fats = 'Y' AND recyclable = 'Y';

1. Find Customer Referee

SELECT name

FROM Customer

WHERE referee\_id !=2;

1. Big Countries

SELECT name, population, area

FROM World

WHERE area >= 3000000 OR population >=25000000;

1. Article View I

SELECT DISTINCT author\_id

FROM Views

WHERE author\_id = viewer\_id

ORDER BY author\_id ASC;

1. Invalid Tweets

SELECT tweet\_id

FROM Tweets

WHERE content.length > 15;

1. Replace Employee ID with Unique Identifier

SELECT Employees.name, EmployeeUNI.unique\_id

FROM Employees, EmployeeUNI

WHERE Employees.id=EmployeeUNI.id;

1. Product Sales Analysis I

SELECT Product.produce\_name, Sales.year, Sales.price

From Sales, Product

WHERE Sales.product\_id = Product.product\_id;

1. Customer who Visited But Did Not Make Any Transactions

SELECT customer\_id, COUNT(Visits.visit\_id) As count\_no\_trans

From Visits

LEFT JOIN Transactions ON Visits.visit\_id = Transaction.visit\_id

WHERE Transications.transactions\_id IS NULL

GROUP BY Visits.customer\_id;

1. Rising Temperature

SELECT w1.id

FROM Weather w1, Weather w2

WHERE DATEDIFF(w1.recordDate, w2.recordDate) = 1 AND w1.temperature > w2.temperature;

1. Averagae Time Of Process Per Machine

select

a.machine\_id,

round(

(select avg(a1.timestamp) from Activity a1 where a1.activity\_type = 'end' and a1.machine\_id = a.machine\_id) -

(select avg(a1.timestamp) from Activity a1 where a1.activity\_type = 'start' and a1.machine\_id = a.machine\_id),3) as processing\_time

from Activity a

group by a.machine\_id

1. Employee Bonous

select Employee.name, Bonus.bonus

from Employee,Bonus

where Employee.empId= Bonus.empId and Bonus.bonus<1000;

1. Students and Examination

select Students.student\_id, Students.student\_name, Subjects.subject\_name count(student\_id) as attended\_exams

from Students, Subjects, Examinations

where Students.student\_id=Examinations.student\_id and Subjects.subject\_name=Examinations.subject\_name group by Students.student\_id, Students.student\_name,Subjects.subject\_name order by Students.student\_id, Subjects.subject\_name;

1. Managers With At Least 5 Direct Reports

SELECT name

FROM Employee

WHERE id IN (

    SELECT managerId

    FROM Employee

    GROUP BY managerId

    HAVING COUNT(\*) >= 5)

1. Confirmation Rate

SELECT s.user\_id,

ROUND(AVG(IF(c.action='confirmed',1,0)),2) as confirmation\_rate

FROM Signups s

LEFT JOIN Confirmations c using (user\_id)

GROUP BY s.user\_id

1. Not Boring Movies

select id,movie,description,rating

from Cinema

where mod(id,2)= 1 and description != 'boring' order  by rating DESC;

1. Average Selling Price

select

p.product\_id,

ifnull(round(sum(units \* price) / sum(units),2),0) as average\_price

from

Prices p left join UnitsSold u

on p.product\_id = u.product\_id and u.purchase\_date between start\_date and end\_date

group by product\_id;

1. Project Employee I

SELECT p.project\_id, ROUND(AVG(e.experience\_years),2) AS average\_years

FROM Project p

LEFT JOIN Employee e

ON p.employee\_id = e.employee\_id

GROUP BY p.project\_id

1. Percentage Of Users Attended A Contest

select

contest\_id,

round(count(distinct user\_id) \* 100 /(select count(user\_id) from Users) ,2) as percentage

from  Register

group by contest\_id

order by percentage desc,contest\_id;

1. Queries Quanity And Percentage

SELECT QUERY\_NAME, ROUND(AVG(RATING/POSITION), 2) AS QUALITY,

ROUND(AVG(IF(RATING<3,1,0))\*100, 2) AS POOR\_QUERY\_PERCENTAGE

FROM QUERIES

GROUP BY QUERY\_NAME

1. Monthly Transactions I

SELECT SUBSTR(trans\_date,1,7) as month, country, count(id) as trans\_count, SUM(CASE WHEN state = 'approved' then 1 else 0 END) as approved\_count, SUM(amount) as trans\_total\_amount, SUM(CASE WHEN state = 'approved' then amount else 0 END) as approved\_total\_amount

FROM Transactions

GROUP BY month, country

1. Immediate Food Delivery

Select

round(avg(order\_date = customer\_pref\_delivery\_date)\*100, 2) as immediate\_percentage

from Delivery

where (customer\_id, order\_date) in (

Select customer\_id, min(order\_date)

from Delivery

group by customer\_id);

1. Game Play Analysis IV

SELECT

ROUND(COUNT(DISTINCT player\_id) / (SELECT COUNT(DISTINCT player\_id)

FROM Activity), 2) AS fraction

FROM Activity

WHERE (player\_id, DATE\_SUB(event\_date, INTERVAL 1 DAY))

IN (SELECT player\_id, MIN(event\_date) AS first\_login FROM Activity GROUP BY player\_id)';

1. Number Of Unique Subjects Taught By Each Teacher

select teacher\_id,COUNT(distinct subject\_id) as cnt

from teacher

group by teacher\_id

1. User Activity For the Past 30 Days I

SELECT activity\_date AS day, COUNT(DISTINCT user\_id) AS active\_users

FROM Activity

WHERE (activity\_date > "2019-06-27" AND activity\_date <= "2019-07-27")

GROUP BY activity\_date;

1. Products Sales Analysis III

SELECT product\_id, year AS first\_year, quantity, price

FROM sales s

WHERE year = (SELECT MIN(YEAR)

FROM sales

WHERE s.product\_id = product\_id

GROUP BY product\_id);

1. Classes More Than 5 Students

SELECT class

FROM Courses

GROUP BY class

HAVING COUNT(student) >= 5;

1. Find Followers Count

select user\_id, count(follower\_id) as followers\_count

from Followers

group by user\_id

order by user\_id asc;

1. Biggest Single Number

SELECT MAX(num) AS num

FROM (

SELECT num

FROM MyNumbers

GROUP BY num

HAVING COUNT(num) = 1

) AS unique\_numbers;

1. Customers Who Bought All Products

SELECT order\_num, COUNT(\*) number, AVG (total\_price) average

FROM items

GROUP BY order\_num

HAVING COUNT(DISTINCT \*) > 2;

1. The Number Of Employees Which Report To Each Employee

select mgr.employee\_id, mgr.name, COUNT(emp.employee\_id) as reports\_count, ROUND(AVG(emp.age)) as average\_age

from employees emp join employees mgr

on emp.reports\_to = mgr.employee\_id

group by employee\_id

order by employee\_id;

1. Primary Department For Each Employee

select mgr.employee\_id, mgr.name, COUNT(emp.employee\_id) as reports\_count, ROUND(AVG(emp.age)) as average\_age

from employees emp join employees mgr

on emp.reports\_to = mgr.employee\_id

group by employee\_id

order by employee\_id;

1. Triangle Judgement

SELECT \*, IF(x+y>z and y+z>x and z+x>y, "Yes", "No") as triangle FROM Triangle

1. Consecutive Number
2. Products Price At A Given Date

select product\_id, new\_price as price from Products where (product\_id,change\_date) in (select product\_id , max(change\_date) as date from Products where change\_date <='2019-08-16' group by product\_id);

1. Last Person To Fit in The Bus

SELECT

q1.person\_name

FROM Queue q1 JOIN Queue q2 ON q1.turn >= q2.turn

GROUP BY q1.turn

HAVING SUM(q2.weight) <= 1000

ORDER BY SUM(q2.weight) DESC

LIMIT 1

1. Count Salary Categories
2. Employee Whose Manager Left The Company

SELECT employee\_id

FROM employees

WHERE salary < 30000 AND manager\_id NOT IN (SELECT employee\_id FROM employees)

ORDER BY employee\_id;

1. Exchange Seats

SELECT CASE

WHEN s.id % 2 <> 0 AND s.id = (SELECT COUNT(\*) FROM Seat) THEN s.id

WHEN s.id % 2 = 0 THEN s.id - 1

ELSE

s.id + 1

END AS id,

student

FROM Seat AS s

ORDER BY id

1. Movie Rating
2. Restaurant Growth

SELECT

visited\_on,

(SELECT SUM(amount)

FROM customer

WHERE visited\_on BETWEEN DATE\_SUB(c.visited\_on, INTERVAL 6 DAY) AND

c.visited\_on ) AS amount,

ROUND(

(SELECT SUM(amount) / 7

FROM customer

WHERE visited\_on BETWEEN DATE\_SUB(c.visited\_on, INTERVAL 6 DAY) AND

c.visited\_on ), 2) AS average\_amount

FROM customer c

WHERE visited\_on >= (

SELECT DATE\_ADD(MIN(visited\_on), INTERVAL 6 DAY)

FROM customer )

GROUP BY visited\_on;

1. Friend Request II : Who Has The Most Friends
2. Investments in 2016

SELECT ROUND(SUM(tiv\_2016), 2) AS tiv\_2016

FROM Insurance

WHERE tiv\_2015 IN (

SELECT tiv\_2015

FROM Insurance

GROUP BY tiv\_2015

HAVING COUNT(\*) > 1)

AND (lat, lon) IN (

SELECT lat, lon

FROM Insurance

GROUP BY lat, lon

HAVING COUNT(\*) = 1);

1. Department Top Three Salaries

WITH allInfo AS (SELECT name AS Employee,

salary,departmentID,name FROM(

SELECT id,name,

salary,departmentID,

DENSE\_RANK() OVER(PARTITION BY departmentId ORDER BY salary DESC)

AS count

FROM Employee)

WHERE count BETWEEN 1 AND 3)

SELECT d.name AS Department,

aa.name AS Employee,

salary FROM allInfo aa

JOIN Department d

ON d.id =aa.departmentId ;

1. Fix Names in a Table

SELECT Users.user\_id , CONCAT(UPPER(SUBSTR(Users.name,1,1)),LOWER(SUBSTR(Users.name,2))) AS name

FROM Users

ORDER BY

Users.user\_id ASC;

1. Patients With a Condition

SELECT patient\_id, patient\_name, conditions

FROM Patients

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

1. Delete Duplicate Email

delete p1 from person p1,person p2

where p1.email=p2.email and p1.id>p2.id;

1. Second Highest Salary

select

(select distinct Salary

from Employee order by salary desc

limit 1 offset 1)

as SecondHighestSalary;

1. Group Sold Products By the Date

SELECT sell\_date,

COUNT(DISTINCT product) AS num\_sold,

GROUP\_CONCAT(DISTINCT product ORDER BY product)

AS products

FROM Activities

GROUP BY sell\_date

ORDER BY sell\_date, product;

1. List The Product Ordered in a Period

SELECT

p.product\_name,

SUM(o.unit) AS unit

FROM

Products p

JOIN

Orders o ON p.product\_id = o.product\_id

WHERE

o.order\_date BETWEEN '2020-02-01' AND '2020-02-29'

GROUP BY

p.product\_id,

p.product\_name

HAVING

SUM(o.unit) >= 100;

1. Find Users With Valid Emails

SELECT \*

FROM Users

WHERE mail REGEXP '^[A-Za-z][A-Za-z0-9\_\.\-]\*@leetcode(\\?com)?\\.com$';