SQL\_PRACTICE

website\_link: https://www.sql-practice.com/

Chatgpt problem\_solving\_link: <https://chatgpt.com/share/c612f2a9-c9cc-466c-855b-668e97056ead>

Problem1\_solution:

SELECT first\_name, last\_name, gender

FROM patients

WHERE gender = 'M';

Problem2\_solution:

SELECT first\_name, last\_name FROM patients

where allergies IS null;

Problem3\_solution:

SELECT first\_name FROM patients

where first\_name like 'C%';

Problem4\_solution:

SELECT first\_name, last\_name

FROM patients

where weight between 100 AND 120;

Problem4\_explanation:

Between is used to get a range of value.

## The SQL BETWEEN Operator

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

### Example[Get your own SQL Server](https://www.w3schools.com/sql/sql_server.asp" \t "_blank)

Selects all products with a price between 10 and 20:

SELECT \* FROM Products  
WHERE Price BETWEEN 10 AND 20;

## BETWEEN with IN

The following SQL statement selects all products with a price between 10 and 20. In addition, the CategoryID must be either 1,2, or 3:

### Example

SELECT \* FROM Products  
WHERE Price BETWEEN 10 AND 20  
AND CategoryID IN (1,2,3);

## BETWEEN Text Values

The following SQL statement selects all products with a ProductName alphabetically between Carnarvon Tigers and Mozzarella di Giovanni:

### Example

SELECT \* FROM Products  
WHERE ProductName BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'  
ORDER BY ProductName;

## BETWEEN Dates

The following SQL statement selects all orders with an OrderDate between '01-July-1996' and '31-July-1996':

### Example

SELECT \* FROM Orders  
WHERE OrderDate BETWEEN #07/01/1996# AND #07/31/1996#;

[Try it Yourself »](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_between_date" \t "_blank)

OR:

### Example

SELECT \* FROM Orders  
WHERE OrderDate BETWEEN '1996-07-01' AND '1996-07-31';

Problem5\_solution:

update patients

Set allergies = 'NKA'

WHERE allergies IS null;

Problem5\_explanation:

The UPDATE statement is used to modify the existing records in a table.

### UPDATE Syntax

UPDATE table\_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;

**Note:** Be careful when updating records in a table! Notice the WHERE clause in the UPDATE statement. The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated!

The following SQL statement updates the first customer (CustomerID = 1) with a new contact person and a new city.

### Example

UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;

## UPDATE Multiple Records

It is the WHERE clause that determines how many records will be updated.

The following SQL statement will update the ContactName to "Juan" for all records where country is "Mexico":

### Example

UPDATE Customers  
SET ContactName='Juan'  
WHERE Country='Mexico';

## Update Warning!

Be careful when updating records. If you omit the WHERE clause, ALL records will be updated!

Problem6\_solution:

select concat(first\_name,' ', last\_name) AS full\_name FROM patients ;

Problem6\_solution:

SELECT count(birth\_date)

FROM patients

where year(birth\_date)= 2010;

Problem6\_breakdown:

\* YEAR(birth\_date) extracts the year part from each birth\_date value in the patients table, and then it is compared to 2010 to find patients born in that year.

\* The COUNT function in SQL is used to count the number of rows that match a specified condition

\* This query will return a single row with a single column: the total number of patients born in the year 2010. The COUNT(\*) function counts all rows that satisfy the WHERE clause condition.

Problem7\_solution:

SELECT first\_name,last\_name,Max(height) AS height

FROM patients;

Problem8\_solution:

SELECT \*

FROM patients

where patient\_id in (1,45,534,879,1000);

Problem8\_breakdown: To query multiple rows you can use in function.

Problem9\_solution:

SELECT count(\*) AS total\_admission

FROM admissions;

Problem10\_solution:

SELECT \*

FROM admissions

where admission\_date = discharge\_date;

Problem11\_solution:

SELECT patient\_id, count(patient\_id) As total\_patient FROM admissions

where patient\_id = 579;

Problem12\_solution:

SELECT Distinct city FROM patients

where province\_id = 'NS';

Problem13\_solution:

SELECT first\_name,last\_name,birth\_date FROM patients

Where height > 160 AND weight > 70;

Problem14\_solution:

SELECT first\_name,last\_name,allergies FROM patients

Where allergies IS NOT NULL AND city = 'Hamilton';

Problem14\_solution:

select patients.first\_name,patients.last\_name,province\_names.province\_name

FROM patients

Inner join province\_names on patients.province\_id = province\_names.province\_id;

Problem15\_solution:

SELECT patients.patient\_id, first\_name, last\_name FROM patients

Inner Join admissions ON patients.patient\_id = admissions.patient\_id

where diagnosis = 'Dementia';

Problem16\_solution:

SELECT doctors.first\_name, doctors.last\_name, Count(attending\_doctor\_id)AS total\_admissions

FROM doctors

Inner Join admissions ON admissions.attending\_doctor\_id = doctors.doctor\_id

group by doctor\_id

Problem17\_Solution:

Select doctors.doctor\_id, concat(first\_name,last\_name) as Fullname,

Min(admission\_date) AS first\_admission\_date, Max(admission\_date) AS last\_admission\_date

from admissions

Inner join doctors On admissions.attending\_doctor\_id = doctors.doctor\_id

group by doctor\_id;

Problem18\_Solution:

select province\_names.province\_name, count(\*) AS patient\_count

FROM patients

join province\_names ON patients.province\_id = province\_names.province\_id

group by province\_name

order by patient\_count desc;

Group By:

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

ORDER BY COUNT(CustomerID) DESC;

Problem19\_Solution:

SELECT first\_name FROM patients

group by first\_name

having count(first\_name) = 1

Problem20\_Solution:

select city, Sum(patient\_id) AS num\_patients from patients

group by city

Order by num\_patients desc, city ASC

Problem21\_Solution:

Select allergies, count(1) as total\_diagnosis from patients

Where allergies is not null

group by allergies

order by total\_diagnosis desc

Problem22\_Solution:

select province\_id, Sum(height) as total\_height From patients

group by province\_id

Having Sum(height) >= 7000;

Problem23\_Solution:

select doctors.doctor\_id, concat(first\_name," ", last\_name) AS full\_name,

Min(admissions.admission\_date) AS first\_admission\_date,Max(admissions.admission\_date) as last\_admission\_date

From admissions

Join doctors on admissions.attending\_doctor\_id = doctors.doctor\_id

group by doctor\_id

Problem24\_Solution:

select province\_names.province\_name, Count(patients.patient\_id) as patient\_count

From patients

join province\_names ON patients.province\_id = province\_names.province\_id

group by province\_name

order by patient\_count desc

Problem25\_Solution:

select day(admission\_date) as day\_number, COUNT(admission\_date) as number\_of\_admission

from admissions

group by day\_number

order by number\_of\_admission desc;

Problem26\_Solution:

Select patient\_id, admission\_date,discharge\_date

from admissions

where patient\_id = 542

group by patient\_id

having Max(admission\_date)

Problem27\_Solution:

Select distinct year(birth\_date) from patients

order by birth\_date

SELECT year(birth\_date)

FROM patients

GROUP BY Year(birth\_date)

Problem28\_Solution:

SELECT patient\_id, first\_name from patients

where first\_name like 's%s' AND len(first\_name) >=6;

Problem29\_Solution:

SELECT first\_name FROM patients

order by LEN(first\_name),first\_name

Problem\_29\_Explanation:

**Length of the First Name (**LEN(first\_name)**):** This part of the ORDER BY clause sorts the names by their length. LEN is a function that returns the number of characters in a string, so this part of the order criteria prioritizes shorter names over longer ones. Names with fewer characters appear first in the result set.

1. **Alphabetical Order of the First Name (**first\_name**):** After sorting by length, if there are multiple names of the same length, this second criterion comes into play. It sorts names of the same length alphabetically. This is a secondary sorting criterion, meaning it only affects the order of names that are already tied in length.

Imagine you have the following names in the patients table: "Ana", "John", "Lee", "Bob", "Chris". The sorting would work as follows:

1. Sort by length:
   * "Lee", "Ana", "Bob" (all 3 letters)
   * "John" (4 letters)
   * "Chris" (5 letters)
2. Sort alphabetically within the same length:
   * "Ana", "Bob", "Lee" (sorted alphabetically among 3-letter names)
   * "John" (only one 4-letter name)
   * "Chris" (only one 5-letter name)

The final ordered list would be: "Ana", "Bob", "Lee", "John", "Chris".

Problem\_30\_solution:

SELECT MAX (Income)

FROM Employee

WHERE Income NOT IN (SELECT Max (Income)

FROM Employee);

Problem\_30\_Explanation:

First we selected the max from that column in the table then we searched for the max value again in that column with excluding the max value which has already been found, so it results in the 2nd maximum value.

Problem\_31\_solution:

select first\_name,last\_name,allergies from patients

where allergies = 'Penicillin' OR allergies = 'Morphine'

order by allergies,first\_name,last\_name;

Problem31\_alternate\_solution:

SELECT

first\_name,

last\_name,

allergies

FROM patients

WHERE

allergies IN ('Penicillin', 'Morphine')

ORDER BY

allergies,

first\_name,

last\_name;

Problem\_32\_Solution:

select first\_name,last\_name,'Patient' AS role from patients

union all

select first\_name,last\_name, 'Doctor' AS role from doctors

Problem\_32\_explanation:

* We are assigning a static string 'Patient' to a new column named role.
* We are assigning a static string 'Doctor' to a new column named role.

Problem\_33\_Solution:

Select first\_name,last\_name,birth\_date from patients

where Year(birth\_date) = 1970

order by month(birth\_date),day(birth\_date)

Problem\_34\_Solution:

Select patient\_id,attending\_doctor\_id,diagnosis from admissions

where (patient\_id % 2 != 0 AND attending\_doctor\_id In(1,5,19))

or

(attending\_doctor\_id Like '%2' AND LEN(patient\_id)= 3)

Problem\_35\_Solution:

SELECT concat(upper(last\_name), ',', lower(first\_name)) as fullname

from patients

order by first\_name desc

Problem\_36\_Solution:

select (max(weight) - min(weight)) AS weighthed\_sum from patients

where last\_name = 'Maroni'

Problem\_36\_Solution:

select concat(patients.first\_name," ", patients.last\_name) as patient\_fullname, admissions.diagnosis,

concat(doctors.first\_name, " ", doctors.last\_name) AS doctors\_fullname from patients

join admissions on patients.patient\_id = admissions.patient\_id

Join doctors On admissions.attending\_doctor\_id = doctors.doctor\_id

Problem\_37\_Solution:

select concat(first\_name,' ', last\_name) AS fullname, Round(height /30.48,1) As height\_feet,

Round(weight \*2.205) as weight\_pounds, birth\_date,

Case

when gender = 'M' Then 'Male'

Else 'Female'

End AS gender\_abbreviation

From patients;

Problem\_37\_Explanation:

## The SQL CASE Expression

The CASE expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

## CASE Syntax

CASE  
    WHEN condition1 THEN result1  
    WHEN condition2 THEN result2  
    WHEN conditionN THEN resultN  
    ELSE result  
END;

## Definition and Usage

The ROUND() function rounds a number to a specified number of decimal places.

****Tip:**** Also look at the [FLOOR()](https://www.w3schools.com/sql/func_sqlserver_floor.asp) and [CEILING()](https://www.w3schools.com/sql/func_sqlserver_ceiling.asp) functions.

## Syntax

ROUND(number, decimals, operation)

## Parameter Values

|  |  |
| --- | --- |
| Parameter | Description |
| number | Required. The number to be rounded |
| decimals | Required. The number of decimal places to round number to |
| operation | Optional. If 0, it rounds the result to the number of decimal. If another value than 0, it truncates the result to the number of decimals. Default value is 0 |

Problem38\_Soluttion:

select products.product\_name,suppliers.company\_name,categories.category\_name

From categories

join products on categories.category\_id = products.category\_id

join suppliers on products.supplier\_id = suppliers.supplier\_id

Problem\_38\_Solution:

select city, company\_name, contact\_name from customers

where city like '%L%'

order by contact\_name

Problem\_38\_Explanation:

Using Like %L% query which contains the letter 'L' in the city name

Problem\_39\_Explanation:

select Round(avg(unit\_price),2),SUM(units\_in\_stock), SUM(discontinued)

from products;

Problem\_40\_Solution:

select city,company\_name,contact\_name,'customers'

from customers

union ALL

select city,company\_name,contact\_name,'suppliers'

from suppliers

Problem\_41\_Solution:

select employees.first\_name, employees.last\_name, count(orders.order\_id) AS num\_orders,

case

WHEN orders.shipped\_date <= orders.required\_date THEN 'Late'

ELSE 'On Time'

END AS shipped

from employees

join orders on employees.employee\_id = orders.employee\_id

GROUP BY

employees.first\_name,

employees.last\_name,

shipped

order by last\_name,first\_name,num\_orders DESC

Problem\_42:

SELECT first\_name, last\_name, department, hire\_date

FROM employees

WHERE DATE\_FORMAT(hire\_date, '%Y-%m') = (

SELECT DATE\_FORMAT(hire\_date, '%Y-%m')

FROM employees

WHERE first\_name = 'John' AND last\_name = 'Smith'

);

Problem\_43:

**Retrieve the** first\_name**,** last\_name**,** department**, and** salary **of the employee with the second highest salary.**

Select first\_name, last\_name, department, salary

From employees  
where salary = (Select Max(salary) from employees where salary < (Select Max(salary) from employees))

Select first\_name, last\_name, department, salary  
From employees  
Order By Desc  
Limit 1 Offset 1;

Problem\_44:

Table: Weather

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

| Column Name | Type |

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

| id | int |

| recordDate | date |

| temperature | int |

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

id is the column with unique values for this table.

There are no different rows with the same recordDate.

This table contains information about the temperature on a certain day.

Write a solution to find all dates' id with higher temperatures compared to its previous dates (yesterday).

Return the result table in ****any order****.

The result format is in the following example.

****Example 1:****

**Input:**

Weather table:

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

| id | recordDate | temperature |

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

| 1 | 2015-01-01 | 10 |

| 2 | 2015-01-02 | 25 |

| 3 | 2015-01-03 | 20 |

| 4 | 2015-01-04 | 30 |

+----+------------+-------------+**Output:**

+----+

| id |

+----+

| 2 |

| 4 |

+----+**Explanation:**

In 2015-01-02, the temperature was higher than the previous day (10 -> 25).

In 2015-01-04, the temperature was higher than the previous day (20 -> 30).

Select today.id, today.temperature

From Weather as today

where today.temperature  >(Select yesterday.temperature

from Weather As yesterday)

SELECT

today.id

FROM

Weather today

JOIN Weather yesterday ON today.recordDate = yesterday.recordDate +INTERVAL 1 DAY

WHERE

today.temperature > yesterday.temperature

problem-44: 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.

 problem 45 :

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.

Return the result table sorted in ****any order****.

The result format is in the following example.

****Example 1:****

**Input:**

Visits

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

| visit\_id | customer\_id |

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

| 1 | 23 |

| 2 | 9 |

| 4 | 30 |

| 5 | 54 |

| 6 | 96 |

| 7 | 54 |

| 8 | 54 |

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

Transactions

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

| transaction\_id | visit\_id | amount |

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

| 2 | 5 | 310 |

| 3 | 5 | 300 |

| 9 | 5 | 200 |

| 12 | 1 | 910 |

| 13 | 2 | 970 |

+----------------+----------+--------+**Output:**

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

| customer\_id | count\_no\_trans |

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

| 54 | 2 |

| 30 | 1 |

| 96 | 1 |

+-------------+----------------+**Explanation:**

Customer with id = 23 visited the mall once and made one transaction during the visit with id = 12.

Customer with id = 9 visited the mall once and made one transaction during the visit with id = 13.

Customer with id = 30 visited the mall once and did not make any transactions.

Customer with id = 54 visited the mall three times. During 2 visits they did not make any transactions, and during one visit they made 3 transactions.

Customer with id = 96 visited the mall once and did not make any transactions.

As we can see, users with IDs 30 and 96 visited the mall one time without making any transactions. Also, user 54 visited the mall twice and did not make any transactions.

Answer:

SELECT

V.customer\_id,

Count(V.customer\_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

V.customer\_id;

Problem 46:

**[620. Not Boring Movies](https://leetcode.com/problems/not-boring-movies/)**

Solved

Easy

Topics

Companies

SQL Schema

Pandas Schema

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****.

The result format is in the following example.

****Example 1:****

**Input:**

Cinema table:

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

| id | movie | description | rating |

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

| 1 | War | great 3D | 8.9 |

| 2 | Science | fiction | 8.5 |

| 3 | irish | boring | 6.2 |

| 4 | Ice song | Fantacy | 8.6 |

| 5 | House card | Interesting | 9.1 |

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

**Output:**

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

| id | movie | description | rating |

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

| 5 | House card | Interesting | 9.1 |

| 1 | War | great 3D | 8.9 |

+----+------------+-------------+--------+**Explanation:**

We have three movies with odd-numbered IDs: 1, 3, and 5. The movie with ID = 3 is boring so we do not include it in the answer.

# Write your MySQL query statement below

Select

id,movie,description,rating

From

Cinema

Where

description != 'boring' AND id%2!=0

ORDER BY rating DESC;