

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

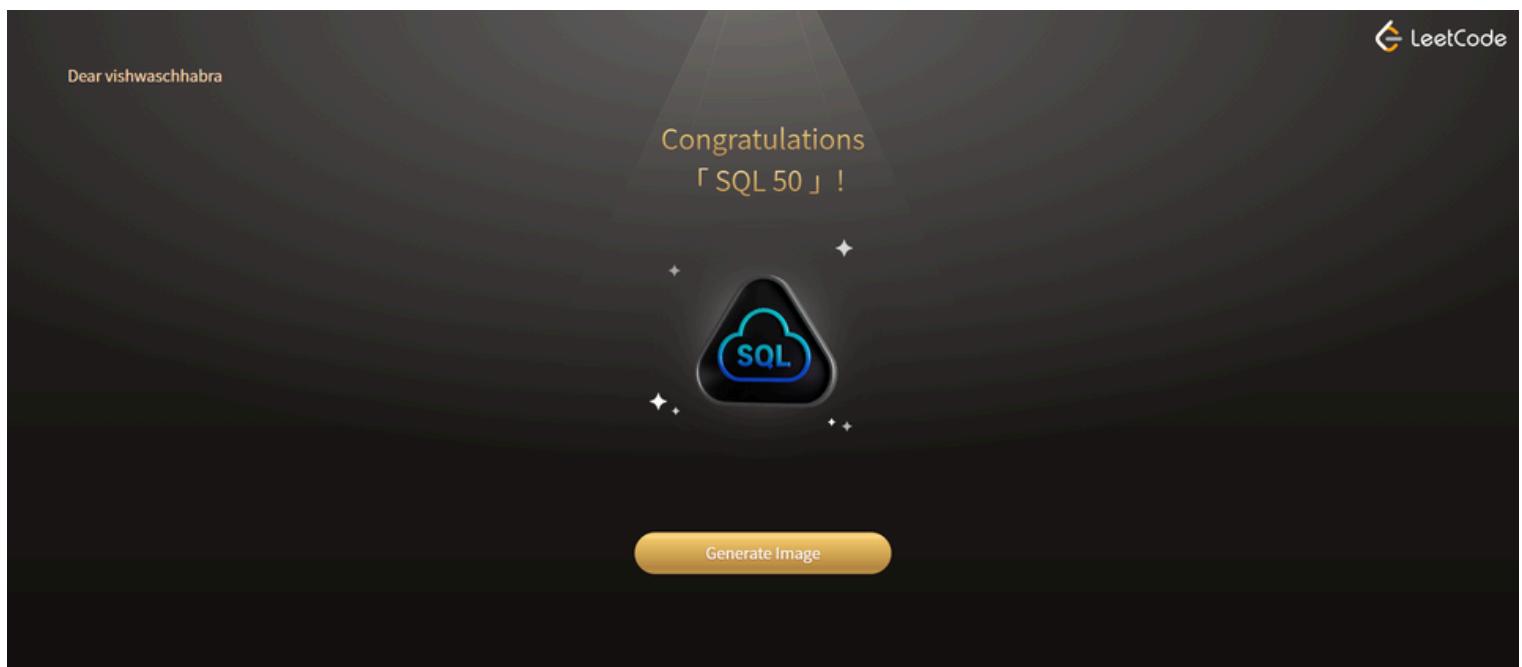
SQL Department (Batch 2)



LeetCode SQL 50 Badge Submission

I have successfully completed 50 structured SQL problems on LeetCode, covering a variety of real-world database concepts including SELECT statements, joins, group by, aggregate functions, subqueries, and window functions. This exercise has significantly improved my understanding of SQL syntax, logic building, and query optimization.

This PDF includes screenshots of each problem I solved, with direct links to the respective LeetCode questions for quick reference and verification. I hope this demonstrates my commitment to learning and consistent problem-solving approach throughout the SQL module.



Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1757. Recyclable and Low Fat Products

Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: Products

Column Name	Type
product_id	int
low_fats	enum
recyclable	enum

product_id is the primary key (column with unique values) for this table.
low_fats is an ENUM (category) of type ('Y', 'N') where 'Y' means this product is low fat and 'N' means it is not.
recyclable is an ENUM (category) of types ('Y', 'N') where 'Y' means this product is recyclable and 'N' means it is not.

Write a solution to find the ids of products that are both low fat and recyclable.

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

The result format is in the following example.

Example 1:

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

Expected

product_id
1
3

Accepted Runtime: 83 ms

Case 1

Input
Products =

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

Expected

product_id
1
3

MySQL v Auto

```
1 SELECT product_id
2 FROM Products
3 WHERE low_fats = 'Y' AND recyclable = 'Y';
```

Ln 3, Col 43

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

584. Find Customer Referee

Solved

Easy Topics Companies Hint

SQL Schema > Pandas Schema >

Table: Customer

Column Name	Type
id	int
name	varchar
referee_id	int

In SQL, id is the primary key column for this table.
Each row of this table indicates the id of a customer, their name, and the id of the customer who referred them.

Find the names of the customer that are **not referred by** the customer with `id = 2`.

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

The result format is in the following example.

Example 1:

Input:
Customer table:

id	name	referee_id
1	Will	null
2	Jane	null
3	Alex	2
4	Bill	null
5	Zack	1
6	Mark	2

Output

name
Will
Jane
Bill
Zack

Accepted Runtime: 86 ms

Case 1

Input
Customer =

id	name	referee_id
1	Will	null
2	Jane	null
3	Alex	2
4	Bill	null
5	Zack	1
6	Mark	2

MySQL v Auto

```
1 SELECT
2   name
3   FROM
4     customer
5   WHERE
6     referee_id != 2 OR referee_id IS NULL;
```

Ln 6, Col 43

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

595. Big Countries Solved

MySQL Auto

```
1 SELECT
2     name, population, area
3 FROM
4     world
5 WHERE
6     area >= 3000000
7     OR population >= 2500000;
```

Saved Ln 7, Col 35

[Testcase](#) | [Test Result](#)

Accepted Runtime: 69 ms

Case 1

Input

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

Output

name	population	area
Afghanistan	25500100	652230
Algeria	37100000	2381741

Expected

Example 1:

3.1K 322 0 Online

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1683. Invalid Tweets Solved

MySQL Auto

```
1 SELECT
2     tweet_id
3 FROM
4     tweets
5 WHERE
6     LENGTH(content) > 15;
```

Saved Ln 6, Col 26

[Testcase](#) | [Test Result](#)

Accepted Runtime: 79 ms

Case 1

Input

tweet_id	content
1	Let us Code
2	More than fifteen chars are here!

Output

tweet_id
2

Expected

Example 1:

Input:
Tweets table:

tweet_id	content
1	Let us Code
2	More than fifteen chars are here!

1.3K 229 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1148. Article Views I Solved

MySQL

```
1 SELECT
2     author_id AS id
3 FROM
4     views
5 WHERE
6     author_id = viewer_id
7 GROUP BY author_id
8 ORDER BY author_id ASC;
```

Saved

Ln 8, Col 24

[Testcase](#) | [Test Result](#)

Accepted Runtime: 92 ms

Case 1

Input

Views =

article_id	author_id	viewer_id	view_date
1	3	5	2019-08-01
1	3	6	2019-08-02
2	7	7	2019-08-01
2	7	6	2019-08-02
4	7	1	2019-07-22
3	4	4	2019-07-21

View more

Output

id
—
4
7

19K 245 0 Online

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1934. Confirmation Rate Solved

MySQL

```
1 SELECT
2     s.user_id,
3     ROUND(
4         COALESCE(SUM(CASE WHEN c.action = 'confirmed' THEN 1 ELSE 0 END) / COUNT(c.user_id), 0),
5         2
6     ) AS confirmation_rate
7 FROM
8     Signups s
9     LEFT JOIN
10    Confirmations c ON s.user_id = c.user_id
11 GROUP BY
12     s.user_id;
```

Saved

Ln 1, Col 7

[Testcase](#) | [Test Result](#)

Accepted Runtime: 158 ms

Case 1

Input

Signups =

user_id	time_stamp
3	2020-03-21 10:16:13
7	2020-01-04 13:57:59
2	2020-07-29 23:09:44
6	2020-12-09 10:39:37

Confirmations =

user_id	time_stamp	action
3	2021-01-06 03:30:46	timeout
3	2021-07-14 14:00:00	timeout
7	2021-06-12 11:57:29	confirmed

13K 207 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

570. Managers with at Least 5 Direct Reports

Solved

Medium Topics Companies Hint

SQL Schema > Pandas Schema >

Table: Employee

Column Name	Type
<code>id</code>	<code>int</code>
<code>name</code>	<code>varchar</code>
<code>department</code>	<code>varchar</code>
<code>managerId</code>	<code>int</code>

`id` is the primary key (column with unique values) for this table. Each row of this table indicates the name of an employee, their department, and the `id` of their manager. If `managerId` is null, then the employee does not have a manager. No employee will be the manager of themselves.

Write a solution to find managers with at least **five direct reports**.

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

The result format is in the following example.

Example 1:

Input:
Employee table:

<code>id</code>	<code>name</code>	<code>department</code>	<code>managerId</code>
101	John	A	null
102	Dan	A	101
103	James	A	101
104	Amy	A	101
105	Anne	A	101
106	Ron	B	101

Output

<code>name</code>
John

MySQL v Auto

```
1 SELECT e.name
2 FROM Employee e
3 JOIN (
4     SELECT managerId
5     FROM Employee
6     WHERE managerId IS NOT NULL
7     GROUP BY managerId
8     HAVING COUNT(*) >= 5
9 ) m ON e.id = m.managerId;
```

Saved Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 88 ms

Case 1

Input

Employee =

<code>id</code>	<code>name</code>	<code>department</code>	<code>managerId</code>
101	John	A	null
102	Dan	A	101
103	James	A	101
104	Amy	A	101
105	Anne	A	101
106	Ron	B	101

Output

<code>name</code>
John

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1280. Students and Examinations

Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: Students

Column Name	Type
<code>student_id</code>	<code>int</code>
<code>student_name</code>	<code>varchar</code>

`student_id` is the primary key (column with unique values) for this table. Each row of this table contains the ID and the name of one student in the school.

Table: Subjects

Column Name	Type
<code>subject_name</code>	<code>varchar</code>

`subject_name` is the primary key (column with unique values) for this table. Each row of this table contains the name of one subject in the school.

Table: Examinations

Column Name	Type

MySQL v Auto

```
1 SELECT
2     s.student_id,
3     s.student_name,
4     sub.subject_name,
5     COUNT(e.student_id) AS attended_exams
6 FROM
7     Students s
8 CROSS JOIN
9     Subjects sub
10 LEFT JOIN
11     Examinations e ON s.student_id = e.student_id AND sub.subject_name = e.subject_name
12 GROUP BY
13     s.student_id, s.student_name, sub.subject_name
14 ORDER BY
15     s.student_id, sub.subject_name;
```

Saved Ln 9, Col 17

Testcase | Test Result

Accepted Runtime: 151 ms

Case 1

Input

Students =

<code>student_id</code>	<code>student_name</code>
1	Alice
2	Bob
13	John
6	Alex

Subjects =

<code>subject_name</code>

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1378. Replace Employee ID With The Unique Identifier

[Easy](#) | [Topics](#) | [Companies](#)

[SQL Schema](#) > [Pandas Schema](#) >

Table: Employees

Column Name	Type
<code>id</code>	<code>int</code>
<code>name</code>	<code>varchar</code>

`id` is the primary key (column with unique values) for this table. Each row of this table contains the id and the name of an employee in a company.

Table: EmployeeUNI

Column Name	Type
<code>id</code>	<code>int</code>
<code>unique_id</code>	<code>int</code>

(`id`, `unique_id`) is the primary key (combination of columns with unique values) for this table. Each row of this table contains the id and the corresponding unique id of an employee in the company.

Write a solution to show the **unique ID** of each user. If a user does not have a unique ID replace just

Accepted Runtime: 93 ms

```
SELECT eu.unique_id, e.name
FROM Employees AS e
LEFT JOIN EmployeeUNI AS eu
ON e.id = eu.id
ORDER BY e.id;
```

Input

Employees =

<code>id</code>	<code>name</code>
1	Alice
7	Bob
11	Meir
90	Winston
3	Jonathan

EmployeeUNI =

<code>id</code>	<code>unique_id</code>
3	1
11	2

Output

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

577. Employee Bonus

[Easy](#) | [Topics](#) | [Companies](#) | [Hint](#)

[SQL Schema](#) > [Pandas Schema](#) >

Table: Employee

Column Name	Type
<code>empId</code>	<code>int</code>
<code>name</code>	<code>varchar</code>
<code>supervisor</code>	<code>int</code>
<code>salary</code>	<code>int</code>

`empId` is the column with unique values for this table. Each row of this table indicates the name and the ID of an employee in addition to their salary and the id of their manager.

Table: Bonus

Column Name	Type
<code>empId</code>	<code>int</code>
<code>bonus</code>	<code>int</code>

`empId` is the column of unique values for this table. `empId` is a foreign key (reference column) to `empId` from the Employee table. Each row of this table contains the id of an employee and their respective bonus.

Accepted Runtime: 114 ms

```
SELECT e.name, b.bonus
FROM Employee e
LEFT JOIN Bonus b ON e.empId = b.empId
WHERE b.bonus < 1000 OR b.bonus IS NULL;
```

Input

Employee =

<code>empId</code>	<code>name</code>	<code>supervisor</code>	<code>salary</code>
3	Brad	null	4000
1	John	3	1000
2	Dan	3	2000
4	Thomas	3	4000

Bonus =

<code>empId</code>	<code>bonus</code>
2	500
4	2000

Output

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[1661. Average Time of Process per Machine](#)

Description | Editorial | Solutions | Submissions | Solved 1

MySQL | Accepted | Auto

```
1 SELECT
2     a.machine_id,
3     ROUND(AVG(b.timestamp - a.timestamp), 3) AS processing_time
4 FROM
5     Activity a
6 JOIN
7     Activity b ON a.machine_id = b.machine_id
8         AND a.process_id = b.process_id
9 WHERE
10    a.activity_type = 'start' AND b.activity_type = 'end'
11 GROUP BY
12    a.machine_id;
```

Saved

Testcase | Test Result

Accepted Runtime: 77 ms

Case 1

Input

Activity	machine_id	process_id	activity_type	timestamp
0	0	0	start	0.712
0	0	0	end	1.52
0	1	1	start	3.14
0	1	1	end	4.12
1	0	0	start	0.55
1	0	0	end	1.55

Output

machine_id	processing_time
0	0.805
1	0.571

View more

[197. Rising Temperature](#)

Description | Editorial | Solutions | Submissions | Solved 1

MySQL | Accepted | Auto

```
1 SELECT w1.id
2 FROM Weather w1, Weather w2
3 WHERE DATEDIFF(w1.recordDate, w2.recordDate) = 1
4 AND w1.temperature > w2.temperature;
```

Saved

Testcase | Test Result

Accepted Runtime: 77 ms

Case 1

Input

Weather	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

Expected

Id
2
4

Example 1:

Input:

Weather table:

	id	recordDate	temperature
1	2015-01-01	10	

Output:

id
2

3.7K 497

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[1581. Customer Who Visited but Did Not Make Any Transactions](#)

MySQL

```
1 SELECT
2     v.customer_id,
3     COUNT(v.visit_id) AS count_no_trans
4 FROM
5     Visits v
6 LEFT JOIN
7     Transactions t ON v.visit_id = t.visit_id
8 WHERE
9     t.transaction_id IS NULL
10 GROUP BY
11     v.customer_id;
```

Saved

Testcase | Test Result

Accepted Runtime: 118 ms

Case 1

Input

Visits =

visit_id	customer_id
1	23
2	9
4	30
5	54
6	96
7	54

View more

Transactions =

transaction_id	visit_id	amount

Write a solution to find the IDs of the users who visited without making any transactions and the number of times they made those type of visits.

2.9K 264 0 Online

[1068. Product Sales Analysis I](#)

MySQL

```
1 SELECT
2     p.product_name, s.year, s.price
3 FROM
4     product p
5     JOIN
6     sales s ON p.product_id = s.product_id;
```

Saved

Testcase | Test Result

Accepted Runtime: 93 ms

Case 1

Input

Sales =

sale_id	product_id	year	quantity	price
1	100	2008	10	5000
2	100	2009	12	5000
7	200	2011	15	9000

Product =

product_id	product_name
100	Nokia
200	Apple
300	Samsung

Output

product_name	year	price
Nokia	2008	5000

(sale_id, year) is the primary key (combination of columns with unique values) of this table.

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

Each row of this table shows a sale on the product product_id in a certain year.

Note that the price is per unit.

Table: Product

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

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

12K 147 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1633. Percentage of Users Attended a Contest

Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: `Users`

Column Name	Type
<code>user_id</code>	int
<code>user_name</code>	varchar

`user_id` is the primary key (column with unique values) for this table. Each row of this table contains the name and the id of a user.

Table: `Register`

Column Name	Type
<code>contest_id</code>	int
<code>user_id</code>	int

(`contest_id`, `user_id`) is the primary key (combination of columns with unique values) for this table. Each row of this table contains the id of a user and the contest they registered into.

Write a solution to find the percentage of the users registered in each contest rounded to two decimals.

Return the result table ordered by `percentage` in descending order. In case of a tie, order it by `contest_id`.

965 Upvotes 166 Comments 0 Online

[Code](#)

MySQL Auto

```
1 SELECT r.contest_id,
2        ROUND(COUNT(DISTINCT r.user_id) * 100.0 / (SELECT COUNT(DISTINCT user_id) FROM Users), 2) AS
3        percentage
4     FROM
5        Register r
6     GROUP BY
7        r.contest_id
8     ORDER BY
9        percentage DESC, r.contest_id ASC;
```

Saved Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 106 ms

Case 1

Input

Users =

<code>user_id</code>	<code>user_name</code>
6	Alice
2	Bob
7	Alex

Register =

<code>contest_id</code>	<code>user_id</code>
215	6
209	2
208	2
210	6
208	6

0 Online

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1075. Project Employees I

Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: `Project`

Column Name	Type
<code>project_id</code>	int
<code>employee_id</code>	int

(`project_id`, `employee_id`) is the primary key of this table. `employee_id` is a foreign key to `Employee` table. Each row of this table indicates that the employee with `employee_id` is working on the project with `project_id`.

Table: `Employee`

Column Name	Type
<code>employee_id</code>	int
<code>name</code>	varchar
<code>experience_years</code>	int

`employee_id` is the primary key of this table. It's guaranteed that `experience_years` is not NULL. Each row of this table contains information about one employee.

Write an SQL query that reports the `average` experience years of all the employees for each project rounded to 2 digits.

837 Upvotes 121 Comments 0 Online

[Code](#)

MySQL Auto

```
1 SELECT p.project_id,
2        ROUND(AVG(e.experience_years), 2) AS average_years
3     FROM
4        Project p
5     JOIN
6        Employee e ON p.employee_id = e.employee_id
7     GROUP BY
8        p.project_id;
```

Saved Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 107 ms

Case 1

Input

Project =

<code>project_id</code>	<code>employee_id</code>
1	1
1	2
1	3
2	1
2	4

Employee =

<code>employee_id</code>	<code>name</code>	<code>experience_years</code>
1	Khaled	3
2	Ali	2
3	John	1
4	Doe	2

0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

620. Not Boring Movies

Solved

MySQL

```
1 SELECT
2 *
3 FROM
4 cinema
5 WHERE
6 id % 2 != 0 AND description != 'boring'
7 ORDER BY rating DESC;
```

Saved

Ln 7, Col 22

[Testcase](#) | [Test Result](#)

Accepted Runtime: 78 ms

Case 1

Input

cinema			
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

Expected

id	movie	description	rating
5	House card	Interesting	9.1
1	War	great 3D	8.9

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

13K 167

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1251. Average Selling Price

Solved

MySQL

```
1 SELECT
2 p.product_id,
3 ROUND(
4 COALESCE(SUM(p.price * us.units) / SUM(us.units), 0),
5 2
6 ) AS average_price
7 FROM
8 Prices p
9 LEFT JOIN
10 UnitsSold us ON p.product_id = us.product_id
11 AND us.purchase_date BETWEEN p.start_date AND p.end_date
12 GROUP BY
13 p.product_id;
```

Saved

Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 103 ms

Case 1

Input

Prices			
product_id	start_date	end_date	price
1	2019-02-17	2019-02-28	5
1	2019-03-01	2019-03-22	20
2	2019-02-01	2019-02-20	15
2	2019-02-21	2019-03-31	30

Output

product_id	purchase_date	units
1	2019-02-25	100
1	2019-03-25	15

UnitsSold =

product_id	purchase_date	units
1	2019-02-25	100
1	2019-03-25	15

This table may contain duplicate rows.

Each row of this table indicates the date, units, and product_id of each product sold.

1.6K 356

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

550. Game Play Analysis IV

Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Activity

Column Name	Type
player_id	int
device_id	int
event_date	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 report the fraction of players that logged in again on the day after they first logged in, rounded to 2 decimal places. In other words, you need to determine the number of players who logged in on the day immediately following their initial login, and divide it by the number of total players.

The result format is in the following example.

Example 1:

Input:
Activity table:

player_id	device_id	event_date	games_played
1	2	2016-03-01	5
1	2	2016-03-02	6
2	3	2017-06-25	1
3	1	2016-03-02	0
3	4	2018-07-03	5

13K 253 0 Online

Code

MySQL Auto

```
1 SELECT
2     ROUND(
3         COUNT(DISTINCT A.player_id) * 1.0 / (SELECT COUNT(DISTINCT player_id) FROM Activity),
4         2
5     ) AS fraction
6     FROM
7     Activity A
8     JOIN
9     (
10        SELECT player_id, MIN(event_date) AS first_login
11        FROM Activity
12        GROUP BY player_id
13    ) T ON A.player_id = T.player_id
14     WHERE
15     DATEDIFF(A.event_date, T.first_login) = 1;
```

Saved Line 1, Col 1

Testcase | **Test Result**

Accepted Runtime: 70 ms

Case 1

Input

Activity =

player_id	device_id	event_date	games_played
1	2	2016-03-01	5
1	2	2016-03-02	6
2	3	2017-06-25	1
3	1	2016-03-02	0
3	4	2018-07-03	5

Output

| fraction |

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1174. Immediate Food Delivery II

Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Delivery

Column Name	Type
delivery_id	int
customer_id	int
order_date	date
customer_pref_delivery_date	date

delivery_id is the column of unique values of this table.

The table holds information about food delivery to customers that make orders at some date and specify a preferred delivery date (on the same order date or after it).

If the customer's preferred delivery date is the same as the order date, then the order is called **immediate**; otherwise, it is called **scheduled**.

The **first order** of a customer is the order with the earliest order date that the customer made. It is guaranteed that a customer has precisely one first order.

Write a solution to find the percentage of immediate orders in the first orders of all customers, rounded to 2 decimal places.

The result format is in the following example.

Example 1:

Input:
Delivery table:

delivery_id	customer_id	order_date	customer_pref_delivery_date
1	1	2019-08-01	2019-08-02
2	2	2019-08-02	2019-08-02

1K 216 0 Online

Code

MySQL Auto

```
1 WITH FirstOrders AS (
2     SELECT
3         customer_id,
4         MIN(order_date) AS first_order_date
5     FROM
6         Delivery
7     GROUP BY
8         customer_id
9     ),
10    FirstOrderDetails AS (
11        SELECT
12            D.*
13        FROM
14            Delivery D
15        JOIN
16            FirstOrders FO ON D.customer_id = FO.customer_id AND D.order_date = FO.first_order_date
17    )
18    SELECT
19        ROUND(
20            SUM(CASE WHEN order_date = customer_pref_delivery_date THEN 1 ELSE 0 END) * 100.0 / COUNT(*),
21            2
22        ) AS immediate_percentage
23    FROM
24        FirstOrderDetails;
```

Saved Line 20, Col 103

Testcase | **Test Result**

Accepted Runtime: 72 ms

Case 1

Input

Delivery =

delivery_id	customer_id	order_date	customer_pref_delivery_date
1	1	2019-08-01	2019-08-02
2	2	2019-08-02	2019-08-02

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1193. Monthly Transactions I

Solved

MySQL ▾ Auto

```
1 SELECT
2   DATE_FORMAT(trans_date, '%Y-%m') AS month,
3   country,
4   COUNT(id) AS trans_count,
5   SUM(CASE WHEN state = 'approved' THEN 1 ELSE 0 END) AS approved_count,
6   SUM(amount) AS trans_total_amount,
7   SUM(CASE WHEN state = 'approved' THEN amount ELSE 0 END) AS approved_total_amount
8 FROM
9   Transactions
10 GROUP BY
11   month, country;
```

Saved

Ln 10, Col 9

[Testcase](#) | [Test Result](#)

Accepted Runtime: 77 ms

Case 1

Input

Transactions =

id	country	state	amount	trans_date
121	US	approved	1000	2018-12-18
122	US	declined	2000	2018-12-19
123	US	approved	2000	2019-01-01
124	DE	approved	2000	2019-01-07

Output

month	country	trans_count	approved_count	trans_total_amount	approved_total_amount
2018-12	US	2	1	3000	1000

1.1K 178 0 Online

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1211. Queries Quality and Percentage

Solved

MySQL ▾ Auto

```
1 SELECT
2   query_name,
3   ROUND(AVG(rating * 1.0 / position), 2) AS quality,
4   ROUND(SUM(CASE WHEN rating < 3 THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS poor_query_percentage
5 FROM
6   Queries
7 GROUP BY
8   query_name;
```

Saved

Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 72 ms

Case 1

Input

Queries =

query_name	result	position	rating
Dog	Golden Retriever	1	5
Dog	German Shepherd	2	5
Dog	Mule	200	1
Cat	Shirazi	5	2
Cat	Siamese	3	3
Cat	Sphynx	7	4

Output

query_name	quality	poor_query_percentage
Dog	2.5	33.33
Cat	0.66	33.33

896 175 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

2356. Number of Unique Subjects Taught by Each Teacher

Description | **Editorial** | **Solutions** | **Submissions**

MySQL | **Auto**

```
1 SELECT
2   teacher_id,
3   COUNT(DISTINCT subject_id) AS cnt
4 FROM
5   Teacher
6 GROUP BY
7   teacher_id;
```

Testcase | **Test Result**

Accepted Runtime: 73 ms

Case 1

Input

teacher_id	subject_id	dept_id
1	2	3
1	2	4
1	3	3
2	1	1
2	2	1
2	3	1

Output

teacher_id	cnt
1	2
2	4

Example 1:

Input:

Teacher table:

teacher_id	subject_id	dept_id
1	2	3
1	2	4
1	3	3
2	1	1
2	2	1
2	3	1

611 73 0 Online

[SQL 50](#)

1141. User Activity for the Past 30 Days I

Description | **Editorial** | **Solutions** | **Submissions**

MySQL | **Auto**

```
1 SELECT
2   activity_date AS day,
3   COUNT(DISTINCT user_id) AS active_users
4 FROM
5   Activity
6 WHERE
7   activity_date BETWEEN DATE_SUB('2019-07-27', INTERVAL 29 DAY) AND '2019-07-27'
8 GROUP BY
9   activity_date;
```

Testcase | **Test Result**

Accepted Runtime: 78 ms

Case 1

Input

user_id	session_id	activity_date	activity_type
1	1	2019-07-20	open_session
1	1	2019-07-20	scroll_down
1	1	2019-07-20	end_session
2	4	2019-07-20	open_session
2	4	2019-07-21	send_message
2	4	2019-07-21	end_session

Output

day	active_users
2019-07-20	2
2019-07-21	2

Example 1:

Input:

Activity table:

user_id	session_id	activity_date	activity_type
1	1	2019-07-20	open_session
1	1	2019-07-20	scroll_down
1	1	2019-07-20	end_session
2	4	2019-07-20	open_session
2	4	2019-07-21	send_message
2	4	2019-07-21	end_session

881 181 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[1070. Product Sales Analysis III](#)

MySQL

```
1 SELECT
2     S.product_id,
3     T.first_year,
4     S.quantity,
5     S.price
6 FROM
7     Sales S
8 JOIN
9     (
10         SELECT
11             product_id,
12             MIN(year) AS first_year
13         FROM
14             Sales
15         GROUP BY
16             product_id
17     ) AS T ON S.product_id = T.product_id AND S.year = T.first_year;
```

Saved

Accepted Runtime: 85 ms

Case 1

Input

Sales	sale_id	product_id	year	quantity	price
	1	100	2008	10	5000
	2	100	2009	12	5000
	7	200	2011	15	9000

Output

product_id	first_year	quantity	price

Example 1:

618 275 0 Online

[596. Classes With at Least 5 Students](#)

MySQL

```
1 SELECT
2     class
3 FROM
4     Courses
5 GROUP BY
6     class
7 HAVING
8     COUNT(student) >= 5;
```

Saved

Accepted Runtime: 75 ms

Case 1

Input

Courses	student	class
	A	Math
	B	English
	C	Math
	D	Biology
	E	Math
	F	Computer

Output

class	
	Math

Example 1:

Input:

Courses table:

student	class
A	Math
B	English

12K 104 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1729. Find Followers Count Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: Followers

Column Name	Type
user_id	int
follower_id	int

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

This table contains the IDs of a user and a follower in a social media app where the follower follows the user.

Write a solution that will, for each user, return the number of followers.

Return the result table ordered by `user_id` in ascending order.

The result format is in the following example.

Example 1:

Input:

Followers table:

user_id	follower_id
0	1
1	0
2	0
2	1

674 65 0 Online

Code

MySQL Auto

```
1 SELECT
2     user_id,
3     COUNT(follower_id) AS followers_count
4 FROM
5     Followers
6 GROUP BY
7     user_id
8 ORDER BY
9     user_id ASC;
```

Saved Ln 1, Col 1

Testcase | **Test Result**

Accepted Runtime: 81 ms

Case 1

Input

Followers =

user_id	follower_id
0	1
1	0
2	0
2	1

Output

user_id	followers_count
0	1
1	1
2	2

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

619. Biggest Single Number Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: MyNumbers

Column Name	Type
num	int

This table may contain duplicates (In other words, there is no primary key for this table in SQL).

Each row of this table contains an integer.

A **single number** is a number that appeared only once in the `MyNumbers` table.

Find the largest **single number**. If there is no **single number**, report `null`.

The result format is in the following example.

Example 1:

Input:

MyNumbers table:

num
8
8
3
3
1
4

810 109 0 Online

Code

MySQL Auto

```
1 SELECT
2     MAX(t.num) AS num
3 FROM
4     (
5         SELECT
6             num
7         FROM
8             MyNumbers
9         GROUP BY
10            num
11        HAVING
12            COUNT(num) = 1
13    ) AS t;
```

Saved Ln 1, Col 1

Testcase | **Test Result**

Accepted Runtime: 96 ms

Case 1 Case 2

Input

MyNumbers =

num
8
8
3
3
1
4

View more

Output

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1789. Primary Department for Each Employee Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: Employee

Column Name	Type
employee_id	int
department_id	int
primary_flag	varchar

(employee_id, department_id) is the primary key (combination of columns with unique values) for this table.
employee_id is the id of the employee.
department_id is the id of the department to which the employee belongs.
primary_flag is an ENUM (category) of type ('Y', 'N'). If the flag is 'Y', the department is the primary department for the employee. If the flag is 'N', the department is not the primary.

Employees can belong to multiple departments. When the employee joins other departments, they need to decide which department is their primary department. Note that when an employee belongs to only one department, their primary column is 'N'.

Write a solution to report all the employees with their primary department. For employees who belong to one department, report their only department.

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

The result format is in the following example.

Input

employee_id	department_id	primary_flag
1	1	N
2	1	Y
2	2	N
3	3	N
4	2	N
4	3	Y

Output

Accepted Runtime: 68 ms

Case 1

0 Online

689 176 0 0

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1731. The Number of Employees Which Report to Each Employee Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table: Employees

Column Name	Type
employee_id	int
name	varchar
reports_to	int
age	int

employee_id is the column with unique values for this table.
This table contains information about the employees and the id of the manager they report to. Some employees do not report to anyone (reports_to is null).

For this problem, we will consider a **manager** an employee who has at least 1 other employee reporting to them.

Write a solution to report the ids and the names of all **managers**, the number of employees who report **directly** to them, and the average age of the reports rounded to the nearest integer.

Return the result table ordered by employee_id.

The result format is in the following example.

Example 1:

Input

employee_id	name	reports_to	age
9	Hercy	null	43
6	Alice	9	41
4	Bob	9	36
2	Winston	null	37

Output

employee_id	name	reports_count	average_age
-------------	------	---------------	-------------

Accepted Runtime: 102 ms

Case 1 Case 2

0 Online

725 139 0 0

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1045. Customers Who Bought All Products Solved

[Medium](#) [Topics](#) [Companies](#)

[SQL Schema](#) > [Pandas Schema](#) >

Table: Customer

Column Name	Type
customer_id	int
product_key	int

This table may contain duplicates rows.
customer_id is not NULL.
product_key is a foreign key (reference column) to Product table.

Table: Product

Column Name	Type
product_key	int

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

Write a solution to report the customer ids from the Customer table that bought all the products in the Product table.

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

The result format is in the following example.

940 likes 86 comments 0 Online

[Code](#)

MySQL Auto

```
1 SELECT
2     customer_id
3 FROM
4     Customer
5 GROUP BY
6     customer_id
7 HAVING
8     COUNT(DISTINCT product_key) = (SELECT COUNT(product_key) FROM Product);
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 102 ms

Case 1

Input

Customer =

customer_id	product_key
1	5
2	6
3	5
3	6
1	6

Product =

product_key
5
6

Output

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

610. Triangle Judgement Solved

[Easy](#) [Topics](#) [Companies](#)

[SQL Schema](#) > [Pandas Schema](#) >

Table: Triangle

Column Name	Type
x	int
y	int
z	int

In SQL, (x, y, z) is the primary key column for this table.
Each row of this table contains the lengths of three line segments.

Report for every three line segments whether they can form a triangle.

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

The result format is in the following example.

Example 1:

Input:

Triangle table:

x	y	z
13	15	30
10	20	15

730 likes 96 comments 0 Online

[Code](#)

MySQL Auto

```
1 SELECT
2     x,
3     y,
4     z,
5     CASE
6         WHEN x + y > z AND x + z > y AND y + z > x THEN 'Yes'
7         ELSE 'No'
8     END AS triangle
9 FROM
10    triangle;
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 80 ms

Case 1

Input

Triangle =

x	y	z
—	—	—
13	15	30
10	20	15

Output

x	y	z	triangle
—	—	—	—
13	15	30	No
10	20	15	Yes

Expected

x	y	z	triangle
---	---	---	----------

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

180. Consecutive Numbers Solved

MySQL

```
1 SELECT
2     DISTINCT L1.num AS ConsecutiveNums
3 FROM
4     Logs L1,
5     Logs L2,
6     Logs L3
7 WHERE
8     L1.id = L2.id - 1
9     AND L2.id = L3.id - 1
10    AND L1.num = L2.num
11    AND L2.num = L3.num;
```

In SQL, id is the primary key for this table.
id is an autoincrement column starting from 1.

Find all numbers that appear at least three times consecutively.

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

The result format is in the following example.

Example 1:

Input:
Logs table:

id	num
1	1
2	1
3	1
4	2
5	1
6	2

Output

ConsecutiveNums
1

2.4K 245 0 Online

[Testcase](#) | [Test Result](#)

Accepted Runtime: 61 ms

Case 1

Input

id	num
1	1
2	1
3	1
4	2
5	1
6	2

[View more](#)

Output

ConsecutiveNums
1

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1164. Product Price at a Given Date Solved

MySQL

```
1 WITH LatestPrices AS (
2     SELECT
3         product_id,
4         new_price,
5         change_date,
6         ROW_NUMBER() OVER(PARTITION BY product_id ORDER BY change_date DESC) as rn
7     FROM
8     Products
9     WHERE
10        change_date <= '2019-08-16'
11    ),
12 ALLProducts AS (
13     SELECT DISTINCT product_id FROM Products
14   )
15 SELECT
16     ap.product_id,
17     COALESCE(ip.new_price, 10) AS price
18   FROM
19     ALLProducts ap
20   LEFT JOIN
21     LatestPrices lp
22   ON ap.product_id = lp.product_id
23   AND lp.rn = 1;
```

(product_id, change_date) is the primary key (combination of columns with unique values) of this table.
Each row of this table indicates that the price of some product was changed to a new price at some date.

Initially, all products have price 10.

Write a solution to find the prices of all products on the date 2019-08-16.

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

The result format is in the following example.

Example 1:

Input:
Products table:

product_id	new_price	change_date
1	20	2019-08-14

Output

product_id	new_price	change_date
1	20	2019-08-14

12K 171 0 Online

[Testcase](#) | [Test Result](#)

Accepted Runtime: 95 ms

Case 1

Input

product_id	new_price	change_date
1	20	2019-08-14

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1204. Last Person to Fit in the Bus Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Queue

Column Name	Type
person_id	int
person_name	varchar
weight	int
turn	int

person_id column contains unique values.
This table has the information about all people waiting for a bus.
The person_id and turn columns will contain all numbers from 1 to n, where n is the number of rows in the table.
turn determines the order of which the people will board the bus, where turn=1 denotes the first person to board and turn=n denotes the last person to board.
weight is the weight of the person in kilograms.

There is a queue of people waiting to board a bus. However, the bus has a weight limit of 1000 kilograms, so there may be some people who cannot board.

Write a solution to find the person_name of the last person that can fit on the bus without exceeding the weight limit. The test cases are generated such that the first person does not exceed the weight limit.

Note that only one person can board the bus at any given turn.

The result format is in the following example.

931 Upvotes 138 Comments 0 Online

[Code](#) MySQL Auto

```
1 SELECT
2   person_name
3 FROM (
4   SELECT
5     person_name,
6     weight,
7     turn,
8     SUM(weight) OVER (ORDER BY turn) AS cumulative_weight
9   FROM
10  Queue
11 ) AS T
12 WHERE
13   cumulative_weight <= 1000
14 ORDER BY
15   turn DESC
16 LIMIT 1;
```

Saved Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 71 ms

Case 1

Input

person_id	person_name	weight	turn
5	Alice	250	1
4	Bob	175	5
3	Alex	350	2
6	John Cena	400	3
1	Winston	500	6
2	Marie	200	4

Output

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1907. Count Salary Categories Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Accounts

Column Name	Type
account_id	int
income	int

account_id is the primary key (column with unique values) for this table.
Each row contains information about the monthly income for one bank account.

Write a solution to calculate the number of bank accounts for each salary category. The salary categories are:

- "Low Salary": All the salaries strictly less than \$20000.
- "Average Salary": All the salaries in the inclusive range [\$20000, \$50000].
- "High Salary": All the salaries strictly greater than \$50000.

The result table must contain all three categories. If there are no accounts in a category, return @.

Return the result table in any order.

The result format is in the following example.

Example 1:

596 Upvotes 125 Comments 0 Online

[Code](#) MySQL Auto

```
1 SELECT
2   'Low Salary' AS category,
3   SUM(CASE WHEN income < 20000 THEN 1 ELSE 0 END) AS accounts_count
4 FROM
5   Accounts
6 UNION ALL
7
8 SELECT
9   'Average Salary' AS category,
10  SUM(CASE WHEN income >= 20000 AND income <= 50000 THEN 1 ELSE 0 END) AS accounts_count
11 FROM
12  Accounts
13 UNION ALL
14
15 SELECT
16   'High Salary' AS category,
17   SUM(CASE WHEN income > 50000 THEN 1 ELSE 0 END) AS accounts_count
18 FROM
19  Accounts;
```

Saved Ln 1, Col 1

[Testcase](#) | [Test Result](#)

Accepted Runtime: 76 ms

Case 1

Input

account_id	income
3	1088939
2	12747
8	87709

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

Description | Editorial | Solutions | Submissions

1978. Employees Whose Manager Left the Company

Solved

MySQL Auto

```
1 SELECT employee_id
2 FROM Employees
3 WHERE salary < 30000
4 AND manager_id IS NOT NULL
5 AND manager_id NOT IN (SELECT employee_id FROM Employees)
6 ORDER BY employee_id;
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 85 ms

Case 1

Input

Employee ID	Name	Manager ID	Salary
3	Mila	9	60301
12	Antonella	null	31000
13	Emery	null	67084
1	Kalel	11	21241
9	Mikaela	null	50937
11	Joziah	6	28485

Output

Employee ID
11

Expected

Employee ID
11

Employees table:

501 Upvotes | 62 Comments | 0 Online

Example 1:

Input:

Employees table:

501 Upvotes | 62 Comments | 0 Online

[SQL 50](#)

Description | Editorial | Solutions | Submissions

626. Exchange Seats

Solved

MySQL Auto

```
1 SELECT
2     CASE
3         WHEN MOD(id, 2) != 0 AND id = (SELECT COUNT(*) FROM Seat) THEN id
4         WHEN MOD(id, 2) != 0 THEN id + 1
5         ELSE id - 1
6     END AS id,
7     student
8 FROM Seat
9 ORDER BY id;
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 74 ms

Case 1

Input

ID	Student
1	Abbot
2	Doris
3	Emerson
4	Green
5	Jeames

Output

ID	Student
1	Doris
2	Abbot
3	Green
4	Emerson

Seat table:

1.7K Upvotes | 138 Comments | 0 Online

Example 1:

Input:

Seat table:

ID	Student
1	Abbot
2	Doris

1.7K Upvotes | 138 Comments | 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1341. Movie Rating Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Movies

Column Name	Type
movie_id	int
title	varchar

movie_id is the primary key (column with unique values) for this table.
title is the name of the movie.

Table: Users

Column Name	Type
user_id	int
name	varchar

user_id is the primary key (column with unique values) for this table.
The column 'name' has unique values.

Table: MovieRating

Column Name	Type
movie_id	int
user_id	int
rating	int

791 196 0 Online

[Code](#) MySQL Auto

```
1 (
2     SELECT T1.name AS results
3     FROM Users AS T1
4     JOIN MovieRating AS T2 ON T1.user_id = T2.user_id
5     GROUP BY T1.user_id, T1.name
6     ORDER BY COUNT(T2.movie_id) DESC, T1.name ASC
7     LIMIT 1
8 )
9 UNION ALL
10 (
11     SELECT T1.title AS results
12     FROM Movies AS T1
13     JOIN MovieRating AS T2 ON T1.movie_id = T2.movie_id
14     WHERE T2.created_at >= '2020-02-01' AND T2.created_at < '2020-03-01'
15     GROUP BY T1.movie_id, T1.title
16     ORDER BY AVG(T2.rating) DESC, T1.title ASC
17     LIMIT 1
18 );
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 170 ms

Case 1

Input

Movies =

movie_id	title
1	Avengers
2	Frozen 2
3	Joker

Users =

user_id	name
---------	------

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1321. Restaurant Growth Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Customer

Column Name	Type
customer_id	int
name	varchar
visited_on	date
amount	int

customer_id, visited_on is the primary key for this table.
This table contains data about customer transactions in a restaurant.
visited_on is the date on which the customer with ID (customer_id) has visited the restaurant.
amount is the total paid by a customer.

You are the restaurant owner and you want to analyze a possible expansion (there will be at least one customer every day). Compute the moving average of how much the customer paid in a seven days window (i.e., current day + 6 days before). average_amount should be rounded to two decimal places.

Return the result table ordered by visited_on in ascending order.

The result format is in the following example.

Example 1:

Input:

Customer table:

customer_id	name	visited_on	amount
1	Jhon	2019-01-01	100
2	Daniel	2019-01-02	110
3	Jade	2019-01-03	120
4	Khaled	2019-01-04	130
5	Winston	2019-01-05	110
6	Elvis	2019-01-06	140

989 146 0 Online

[Code](#) MySQL Auto

```
1 WITH DailySales AS (
2     SELECT
3         visited_on,
4         SUM(amount) AS daily_amount
5     FROM
6         Customer
7     GROUP BY
8         visited_on
9 ),
10 RankedDailySales AS (
11     SELECT
12         visited_on,
13         daily_amount,
14         ROW_NUMBER() OVER (ORDER BY visited_on) AS row_num
15     FROM
16         DailySales
17 ),
18 WindowCalculations AS (
19     SELECT
20         visited_on,
21         daily_amount,
22         SUM(daily_amount) OVER (
23             ORDER BY visited_on
24             ROWS BETWEEN 6 PRECEDING AND CURRENT ROW
25 ) AS seven_day_sum,
26         AVG(daily_amount) OVER (
27             ORDER BY visited_on
28             ROWS BETWEEN 6 PRECEDING AND CURRENT ROW
29 ) AS seven_day_avg,
30         row_num
31     FROM
32         RankedDailySales
33 )
34 SELECT
35     visited_on,
36     seven_day_sum AS amount,
37     ROUND(seven_day_avg, 2) AS average_amount
38     FROM
39     WindowCalculations
40     WHERE
41     row_num >= 7
42     ORDER BY
43     visited_on;
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 61 ms

Case 1

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

602. Friend Requests II: Who Has the Most Friends Solved

Medium Topics Companies Hint

SQL Schema > Pandas Schema >

Table: RequestAccepted

Column Name	Type
requester_id	int
accepter_id	int
accept_date	date

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

This table contains the ID of the user who sent the request, the ID of the user who received the request, and the date when the request was accepted.

Write a solution to find the people who have the most friends and the most friends number.

The test cases are generated so that only one person has the most friends.

The result format is in the following example.

Example 1:

Input:

```
RequestAccepted table:
+-----+-----+-----+
| requester_id | accepter_id | accept_date |
+-----+-----+-----+
| 1           | 2           | 2016/06/03 |
| 1           | 3           | 2016/06/08 |
| 2           | 3           | 2016/06/08 |
| 3           | 4           | 2016/06/09 |
+-----+-----+-----+
```

Output:

id	num
—	—
3	3

Accepted Runtime: 87 ms

Case 1

Input

RequestAccepted =

requester_id	accepter_id	accept_date
1	2	2016/06/03
1	3	2016/06/08
2	3	2016/06/08
3	4	2016/06/09

Output

id	num
—	—
3	3

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

585. Investments in 2016 Solved

Medium Topics Companies Hint

SQL Schema > Pandas Schema >

Table: Insurance

Column Name	Type
pid	int
tiv_2015	float
tiv_2016	float
lat	float
lon	float

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

Each row of this table contains information about one policy where:

pid is the policyholder's policy ID.

tiv_2015 is the total investment value in 2015 and tiv_2016 is the total investment value in 2016.

lat is the latitude of the policy holder's city. It's guaranteed that lat is not NULL.

lon is the longitude of the policy holder's city. It's guaranteed that lon is not NULL.

Write a solution to report the sum of all total investment values in 2016 `tiv_2016`, for all policyholders who:

- have the same `tiv_2015` value as one or more other policyholders, and
- are not located in the same city as any other policyholder (i.e. the (`lat`, `lon`) attribute pairs must be unique).

Round `tiv_2016` to two decimal places.

Accepted Runtime: 72 ms

Case 1

Input

Insurance =

pid	tiv_2015	tiv_2016	lat	lon
1	10	5	10	10
2	20	20	20	20
3	10	30	20	20
4	10	40	40	40

Output

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

185. Department Top Three Salaries

MySQL

```
WITH RankedEmployees AS (
    SELECT
        E.id,
        E.name AS EmployeeName,
        E.salary,
        E.departmentId,
        DENSE_RANK() OVER (PARTITION BY E.departmentId ORDER BY E.salary DESC) AS salary_rank
    FROM
        Employee AS E
)
SELECT
    D.name AS Department,
    RE.EmployeeName AS Employee,
    RE.salary AS Salary
FROM
    RankedEmployees AS RE
JOIN
    Department AS D ON RE.departmentId = D.id
WHERE
    RE.salary_rank <= 3
ORDER BY
    D.name, RE.salary DESC;
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 106 ms

Case 1

Input

Employee
id name salary departmentId
-- --- ----- -----
1 Joe 85000 1
2 Henry 80000 2

2.4K 250 0 Online

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1667. Fix Names in a Table

MySQL

```
SELECT
    user_id,
    CONCAT(
        UPPER(SUBSTRING(name, 1, 1)),
        LOWER(SUBSTRING(name, 2, LENGTH(name))))
    ) AS name
FROM
    Users
ORDER BY
    user_id;
```

Saved

[Testcase](#) | [Test Result](#)

Accepted Runtime: 79 ms

Case 1

Input

Users
user_id name
--- ---
1 aLice
2 bOB

Output

user_id	name
--- ---	
1 Alice	
2 Bob	

Expected

User ID	Name
--- ---	
1 Alice	
2 Bob	

983 112 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[1527. Patients With a Condition](#)

MySQL v Auto

```
1 SELECT patient_id, patient_name, conditions
2 FROM Patients
3 WHERE conditions REGEXP '^(| )DIAB1';
```

Saved

Testcase | Test Result

Accepted Runtime: 75 ms

Case 1

Input

patient_id	patient_name	conditions
1	Daniel	YFEV COUGH
2	Alice	
3	Bob	DIAB100 MYOP
4	George	ACNE DIAB100
5	Alain	DIAB201

Output

patient_id	patient_name	conditions
3	Bob	DIAB100 MYOP
4	George	ACNE DIAB100

Expected

patient_id	patient_name	conditions
3	Bob	DIAB100 MYOP
4	George	ACNE DIAB100

Description

MySQL Schema > Pandas Schema >

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.

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

The result format is in the following example.

Example 1:

Input:

patient_id	patient_name	conditions
1	Daniel	YFEV COUGH

Output

patient_id	patient_name	conditions
3	Bob	DIAB100 MYOP
4	George	ACNE DIAB100

Online

772 98 0 0 Online

[196. Delete Duplicate Emails](#)

MySQL v Auto

```
1 WITH DuplicateEmails AS (
2     SELECT
3         id,
4         ROW_NUMBER() OVER(PARTITION BY email ORDER BY id ASC) as rn
5     FROM Person
6 )
7 DELETE FROM Person
8 WHERE id IN (
9     SELECT id
10    FROM DuplicateEmails
11   WHERE rn > 1
12 );
```

Saved

Testcase | Test Result

Accepted Runtime: 68 ms

Case 1

Input

id	email
1	john@example.com
2	bob@example.com
3	john@example.com

Output

id	email
1	john@example.com
2	bob@example.com

Description

MySQL Schema > Pandas Schema >

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 **delete** all duplicate emails, keeping only one unique email with the smallest id.

For SQL users, please note that you are supposed to write a `DELETE` statement and not a `SELECT` one.

For Pandas users, please note that you are supposed to modify `Person` in place.

After running your script, the answer shown is the `Person` table. The driver will first compile and run your piece of code and then show the `Person` table. The final order of the `Person` table **does not matter**.

The result format is in the following example.

Example 1:

Input:

id	email
1	john@example.com
2	bob@example.com
3	john@example.com

Output

id	email
1	john@example.com
2	bob@example.com

Online

19K 291 0 0 Online

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

176. Second Highest Salary

Solved

Medium Topics Companies

SQL Schema > Pandas Schema >

Table: Employee

Column Name	Type
<code>id</code>	<code>int</code>
<code>salary</code>	<code>int</code>

`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` (return `None` in Pandas).

The result format is in the following example.

Example 1:

Input:
Employee table:

<code>id</code>	<code>salary</code>
1	100
2	200
3	300

Output:

<code>SecondHighestSalary</code>
200

Expected:

<code>SecondHighestSalary</code>
200

Code

```
MySQL v Auto
1 SELECT
2   (
3     SELECT DISTINCT salary
4     FROM Employee
5     ORDER BY salary DESC
6     LIMIT 1 OFFSET 1
7   ) AS SecondHighestSalary;
```

Testcase Test Result

Accepted Runtime: 98 ms

Case 1 Case 2

Input

Employee =

<code>id</code>	<code>salary</code>
1	100
2	200
3	300

Output

`SecondHighestSalary`

<code>SecondHighestSalary</code>
200

Expected

`SecondHighestSalary`

<code>SecondHighestSalary</code>
200

[SQL 50](#)

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

1484. Group Sold Products By The Date

Solved

Easy Topics Companies

SQL Schema > Pandas Schema >

Table Activities:

Column Name	Type
<code>sell_date</code>	<code>date</code>
<code>product</code>	<code>varchar</code>

There is no primary key (column with unique values) for this table. It may contain duplicates. Each row of this table contains the product name and the date it was sold in a market.

Write a solution to find for each date the number of different products sold and their names.

The sold products names for each date should be sorted lexicographically.

Return the result table ordered by `sell_date`.

The result format is in the following example.

Example 1:

Input:
Activities table:

<code>sell_date</code>	<code>product</code>
2020-05-30	Headphone
2020-06-01	Pencil
2020-06-02	Mask
2020-05-30	Basketball
2020-06-01	Bible
2020-06-02	Mask

Output:

<code>sell_date</code>	<code>num_sold</code>	<code>products</code>
2020-05-30	2	Headphone, Basketball
2020-06-01	2	Pencil, Bible
2020-06-02	2	Mask, Mask

Code

```
MySQL v Auto
1 SELECT
2   sell_date,
3   COUNT(DISTINCT product) AS num_sold,
4   GROUP_CONCAT(DISTINCT product ORDER BY product ASC) AS products
5   FROM
6   Activities
7   GROUP BY
8   sell_date
9   ORDER BY
10  sell_date;
```

Testcase Test Result

Accepted Runtime: 81 ms

Case 1

Input

Activities =

<code>sell_date</code>	<code>product</code>
2020-05-30	Headphone
2020-06-01	Pencil
2020-06-02	Mask
2020-05-30	Basketball
2020-06-01	Bible
2020-06-02	Mask

Output

`sell_date` `num_sold` `products`

<code>sell_date</code>	<code>num_sold</code>	<code>products</code>
2020-05-30	2	Headphone, Basketball
2020-06-01	2	Pencil, Bible
2020-06-02	2	Mask, Mask

Week 6 Assignment

(LeetCode Problems)

Vishwas Chhabra

Celebal Summer Internship 2025

SQL Department (Batch 2)



#Every screenshot is embedded with the direct LeetCode question link. Click to view.

1327. List the Products Ordered in a Period

MySQL

```
1 SELECT
2     P.product_name,
3     O.total_units AS unit
4 FROM
5     Products AS P
6 JOIN (
7     SELECT
8         product_id,
9         SUM(unit) AS total_units
10    FROM
11        Orders
12   WHERE
13       order_date >= '2020-02-01' AND order_date <= '2020-02-29'
14   GROUP BY
15       product_id
16   HAVING
17       SUM(unit) >= 100
18 ) AS O
19 ON P.product_id = O.product_id;
```

Accepted Runtime: 112 ms

Case 1

Input

Products =

product_id	product_name	product_category
1	Leetcode Solutions	Book
2	Jewels of Stringology	Book
3	HP	Laptop
4	Lenovo	Laptop

Write a solution to get the names of products that have at least 100 units ordered in February 2020 and their amount.

484 73 0 Online

1517. Find Users With Valid E-Mails

MySQL

```
1 SELECT user_id, name, mail
2 FROM Users
3 WHERE
4     mail LIKE '%@leetcode.com'
5     AND BINARY RIGHT(mail, 13) = "@leetcode.com"
6     AND LEFT(mail, LENGTH(mail) - 13) REGEXP '^[a-zA-Z][a-zA-Z0-9-.]*$';
```

Accepted Runtime: 69 ms

Case 1

Input

Users =

user_id	name	mail
1	Winston	winston@leetcode.com
2	Jonathan	jonathanisgreat
3	Annabelle	bella@leetcode.com
4	Sally	sally.com@leetcode.com
5	Marwan	quarz#2020@leetcode.com
6	David	david69@gmail.com

Write a solution to find the users who have valid emails.

A valid e-mail has a prefix name and a domain where:

- The prefix name is a string that may contain letters (upper or lower case), digits, underscore (_), period (.), and/or dash (-). The prefix name must start with a letter.
- The domain is "@leetcode.com".

Return the result table in any order.

The result format is in the following example.

Example 1:

607 120 0 Online