

Problem 3436: Find Valid Emails

Problem Information

Difficulty: Easy

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Table:

Users

+-----+-----+ | Column Name | Type | +-----+-----+ | user_id | int | | email | varchar | +-----+-----+ (user_id) is the unique key for this table. Each row contains a user's unique ID and email address.

Write a solution to find all the

valid email addresses

. A valid email address meets the following criteria:

It contains exactly one

@

symbol.

It ends with

.com

.

The part before the

@

symbol contains only

alphanumeric

characters and

underscores

.

The part after the

@

symbol and before

.com

contains a domain name

that contains only letters

.

Return

the result table ordered by

user_id

in

ascending

order

.

Example:

Input:

Users table:

```
+-----+-----+ | user_id | email | +-----+-----+ | 1 |
alice@example.com | | 2 | bob_at_example.com | | 3 | charlie@example.net | | 4 |
david@domain.com | | 5 | eve@invalid | +-----+-----+
```

Output:

```
+-----+-----+ | user_id | email | +-----+-----+ | 1 | alice@example.com
| | 4 | david@domain.com | +-----+-----+
```

Explanation:

alice@example.com

is valid because it contains one

@

, alice is alphanumeric, and example.com starts with a letter and ends with .com.

bob_at_example.com

is invalid because it contains an underscore instead of an

@

.

charlie@example.net

is invalid because the domain does not end with

.com

.

david@domain.com

is valid because it meets all criteria.

eve@invalid

is invalid because the domain does not end with

.com

.

Result table is ordered by user_id in ascending order.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd
```

```
def find_valid_emails(users: pd.DataFrame) -> pd.DataFrame:
```

Solutions

MySQL Solution:

```
# Write your MySQL query statement below
```

MS SQL Server Solution:

```
/* Write your T-SQL query statement below */
```

PostgreSQL Solution:

```
-- Write your PostgreSQL query statement below
```

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

def find_valid_emails(users: pd.DataFrame) -> pd.DataFrame:
```