Problem 2199: Finding the Topic of Each Post

Problem Information

Difficulty: Hard
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Table:

Keywords

+-------------+---------+ | Column Name | Type | +-------------+---------+ | topic_id | int | | word | varchar | +-------------+---------+ (topic_id, word) is the primary key (combination of columns with unique values) for this table. Each row of this table contains the id of a topic and a word that is used to express this topic. There may be more than one word to express the same topic and one word may be used to express multiple topics.

Table:

Posts

+-------------+---------+ | Column Name | Type | +-------------+---------+ | post_id | int | | content | varchar | +-------------+---------+ post_id is the primary key (column with unique values) for this table. Each row of this table contains the ID of a post and its content. Content will consist only of English letters and spaces.

Leetcode has collected some posts from its social media website and is interested in finding the topics of each post. Each topic can be expressed by one or more keywords. If a keyword of a certain topic exists in the content of a post (

case insensitive

) then the post has this topic.

Write a solution to find the topics of each post according to the following rules:

If the post does not have keywords from any topic, its topic should be

"Ambiguous!"

.

If the post has at least one keyword of any topic, its topic should be a string of the IDs of its topics sorted in ascending order and separated by commas

','

. The string should not contain duplicate IDs.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Keywords table: +----------+----------+ | topic_id | word | +----------+----------+ | 1 | handball | | 1 | football | | 3 | WAR | | 2 | Vaccine | +----------+----------+ Posts table: +---------+--+ | post_id | content | +---------+--+ | 1 | We call it soccer They call it football hahaha | | 2 | Americans prefer basketball while Europeans love handball and football | | 3 | stop the war and play handball | | 4 | warning I planted some flowers this morning and then got vaccinated | +---------+--+

Output:

+---------+------------+ | post_id | topic | +---------+------------+ | 1 | 1 | | 2 | 1 | | 3 | 1,3 | | 4 | Ambiguous! | +---------+------------+

Explanation:

1: "We call it soccer They call it football hahaha" "football" expresses topic 1. There is no other word that expresses any other topic.

2: "Americans prefer basketball while Europeans love handball and football" "handball" expresses topic 1. "football" expresses topic 1. There is no other word that expresses any other topic.

3: "stop the war and play handball" "war" expresses topic 3. "handball" expresses topic 1. There is no other word that expresses any other topic.

4: "warning I planted some flowers this morning and then got vaccinated" There is no word in this sentence that expresses any topic. Note that "warning" is different from "war" although they have a common prefix. This post is ambiguous.

Note that it is okay to have one word that expresses more than one topic.

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_topic(keywords: pd.DataFrame, posts: 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_topic(keywords: pd.DataFrame, posts: pd.DataFrame) -> pd.DataFrame:
```