Problem 620: Not Boring Movies

Problem Information

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

Problem Description

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.

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 not_boring_movies(cinema: 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 not_boring_movies(cinema: pd.DataFrame) -> pd.DataFrame:
```