Problem 603: Consecutive Available Seats

Problem Information

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

Problem Description

Table:

Cinema

+-------------+------+ | Column Name | Type | +-------------+------+ | seat_id | int | | free | bool | +-------------+------+ seat_id is an auto-increment column for this table. Each row of this table indicates whether the i

th

seat is free or not. 1 means free while 0 means occupied.

Find all the consecutive available seats in the cinema.

Return the result table

ordered

by

seat_id

in ascending order

.

The test cases are generated so that more than two seats are consecutively available.

The result format is in the following example.

Example 1:

Input:

Cinema table: +---------+------+ | seat_id | free | +---------+------+ | 1 | 1 | | 2 | 0 | | 3 | 1 | | 4 | 1 | | 5 | 1 | +---------+------+

Output:

+---------+ | seat_id | +---------+ | 3 | | 4 | | 5 | +---------+

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