You have 30 minutes to complete the 5 questions.

Consider the tables **TableA** and **TableB**, which contain customer level and transaction level info, respectively.

**TableA** - Customer Info

|  |  |
| --- | --- |
| **cust\_id**  int (primary key) | **name** (first last format)  varchar(50) |
| 1 | Steve Jobs |
| 2 | Bill Gates |
| … | … |

**TableB** - Transaction Info

|  |  |  |  |
| --- | --- | --- | --- |
| **cust\_id**  int | **tran\_id**  int (primary key) | **time**  datetime | **amount**  money |
| 1 | 1 | '2017-12-31 23:59:59' | 500 |
| 1 | 2 | '2018-01-31 20:40:00' | 150 |
| 2 | 3 | '2018-01-31 21:41:44' | 900 |
| … | … | … | … |

Assume no customer has two transactions at the exact same time

Write SQL code to output the following:

**1.**

The unique names of customers who have spent over $500 on a single transaction

**2.**

The number of the customers with both:

1. a first name that begins with 'S', and
2. >5 transactions since March 1, 2018

**3.**

The number of transactions where the sum of the amounts of:

1. the current transaction, and
2. the previous transaction (i.e. most recent transaction before the current transaction) that this customer made,

is above $1000.

Do not include the first transaction from each customer, even if it's above $1000.

**4.**

Suppose that the primary key constraint in TableA was removed and duplicate records were inserted. List all duplicate records

For example, if TableA was:

|  |  |
| --- | --- |
| **cust\_id** | **name** |
| 1 | Steve Jobs |
| 1 | Steve Jobs |
| 2 | Bill Gates |
| 3 | Elon Musk |

The output would be:

|  |  |
| --- | --- |
| **cust\_id** | **name** |
| 1 | Steve Jobs |
| 1 | Steve Jobs |

**5.**

Output the rows of TableB which contain only the third transaction from each customer

For example if TableB was:

|  |  |  |  |
| --- | --- | --- | --- |
| **cust\_id**  int | **tran\_id**  int (primary key) | **time**  datetime | **amount**  money |
| 1 | 1 | '2017-12-31 23:59:59' | 500 |
| 2 | 2 | '2018-01-31 20:40:00' | 150 |
| 3 | 3 | '2018-01-31 21:41:44' | 900 |
| 2 | 4 | '2018-02-20 00:00:00' | 200 |
| 2 | 5 | '2018-02-21 00:00:00' | 300 |
| 2 | 6 | '2018-02-22 00:00:00' | 400 |

The output would be:

|  |  |  |  |
| --- | --- | --- | --- |
| **cust\_id**  int | **tran\_id**  int (primary key) | **time**  datetime | **amount**  money |
| 2 | 5 | '2018-02-21 00:00:00' | 300 |