Trips and Users

Find the **cancellation rate** of requests with unbanned users (**both client and driver must not be banned**) each day between "2013-10-01" and "2013-10-03"

1. Creating table

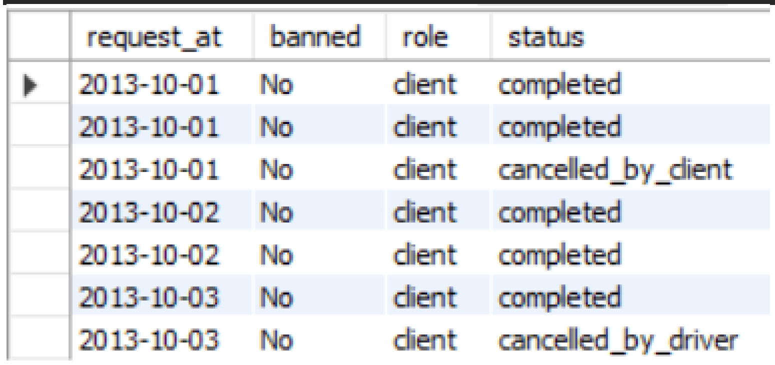
```
Create table If Not Exists Trips (id int, client_id int, driver_id int, city_id
int, status ENUM('completed', 'cancelled_by_driver', 'cancelled_by_client'),
request_at varchar(50))
Create table If Not Exists Users (users_id int, banned varchar(50), role
ENUM('client', 'driver', 'partner'))
Truncate table Trips
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('1', '1', '10', '1', 'completed', '2013-10-01')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('2', '2', '11', '1', 'cancelled_by_driver', '2013-10-01')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('3', '3', '12', '6', 'completed', '2013-10-01')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('4', '4', '13', '6', 'cancelled_by_client', '2013-10-01')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('5', '1', '10', '1', 'completed', '2013-10-02')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('6', '2', '11', '6', 'completed', '2013-10-02')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('7', '3', '12', '6', 'completed', '2013-10-02')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('8', '2', '12', '12', 'completed', '2013-10-03')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('9', '3', '10', '12', 'completed', '2013-10-03')
insert into Trips (id, client_id, driver_id, city_id, status, request_at) values
('10', '4', '13', '12', 'cancelled_by_driver', '2013-10-03')
Truncate table Users
insert into Users (users_id, banned, role) values ('1', 'No', 'client')
insert into Users (users_id, banned, role) values ('2', 'Yes', 'client')
insert into Users (users_id, banned, role) values ('3', 'No', 'client')
insert into Users (users_id, banned, role) values ('4', 'No', 'client')
insert into Users (users_id, banned, role) values ('10', 'No', 'driver')
insert into Users (users_id, banned, role) values ('11', 'No', 'driver')
insert into Users (users_id, banned, role) values ('12', 'No', 'driver')
insert into Users (users_id, banned, role) values ('13', 'No', 'driver')
```

2. Merging tables

```
SELECT
    *
FROM
    Trips AS t
        LEFT JOIN
    Users AS u ON t.client_id = u.users_id;
```

3. Create table from record day between "2013-10-01" and "2013-10-03"

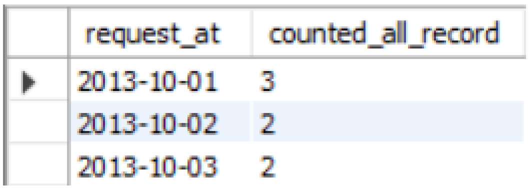
```
CREATE TABLE basic AS SELECT t.request_at, u.banned, u.role, t.status FROM
    Trips AS t
        LEFT JOIN
    Users AS u ON t.client_id = u.users_id
WHERE
    banned = 'No'
        AND request_at BETWEEN '2013-10-01' AND '2013-10-03';
```



	request_at	banned	role	status
▶	2013-10-01	No	client	completed
	2013-10-01	No	client	completed
	2013-10-01	No	client	cancelled_by_client
	2013-10-02	No	client	completed
	2013-10-02	No	client	completed
	2013-10-03	No	client	completed
	2013-10-03	No	client	cancelled_by_driver

4. Creating table of counted all requests

```
CREATE TABLE counted_all AS SELECT request_at, COUNT(*) AS counted_all_record FROM
    basic
GROUP BY request_at;
```

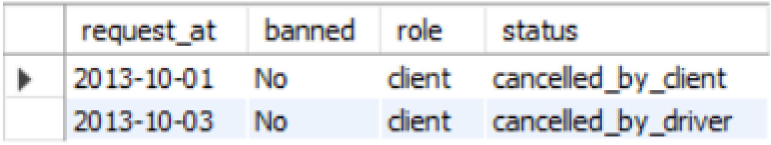


	request_at	counted_all_record
▶	2013-10-01	3
	2013-10-02	2
	2013-10-03	2

5. Counting not canceled requests

Selecting not canceled requests

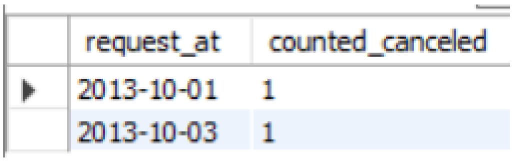
```
SELECT
    *
FROM
    basic
WHERE
    status != 'completed';
WHERE
    banned = 'No';
```



	request_at	banned	role	status
▶	2013-10-01	No	client	cancelled_by_client
	2013-10-03	No	client	cancelled_by_driver

Counting request and creating new table

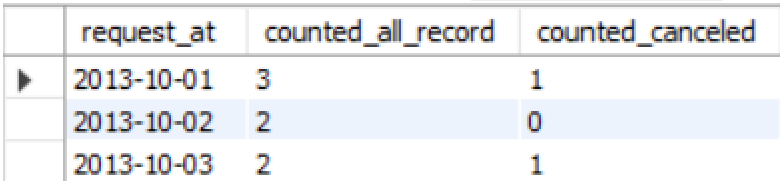
```
CREATE TABLE canceled AS SELECT request_at, COUNT(*) AS counted_canceled FROM
    (SELECT
        *
    FROM
        basic
    WHERE
        status != 'completed') AS t
GROUP BY request_at;
```



	request_at	counted_canceled
▶	2013-10-01	1
	2013-10-03	1

6. Mergin canceled and not canceled table

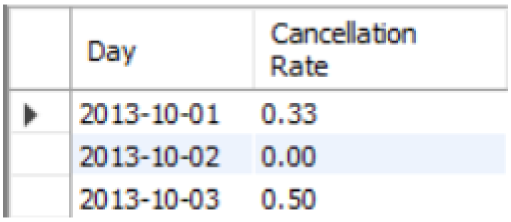
```
SELECT
    a.request_at,
    a.counted_all_record,
    COALESCE(c.counted_canceled, 0) AS counted_canceled
FROM
    counted_all AS a
        LEFT JOIN
    canceled AS c ON a.request_at = c.request_at;
```



	request_at	counted_all_record	counted_canceled
▶	2013-10-01	3	1
	2013-10-02	2	0
	2013-10-03	2	1

7. Divide counted_canceled by counted_all_record

```
with tbl as (
SELECT
    a.request_at,
    a.counted_all_record,
    COALESCE(c.counted_canceled, 0) AS counted_canceled
FROM
    counted_all AS a
        LEFT JOIN
    canceled AS c ON a.request_at = c.request_at)
select request_at as 'Day', round((counted_canceled/counted_all_record),2) as
'Cancellation Rate' from tbl;
```



	Day	Cancellation Rate
▶	2013-10-01	0.33
	2013-10-02	0.00
	2013-10-03	0.50