

Задание 1

Даны две таблицы. Первая — **user**, она содержит информацию о пользователях. Вторая таблица — **clickstream**, она содержит поток пользовательских событий.

user

user_id	created_date	registration_type	contact	country
1253	2022-01-03	PHONE	+79999999999	Rus
1278	2023-02-10	VK	aaaaaaa	Rus
1456	2022-04-09	EMAIL	aaaa@aaa.ru	Bel
1642	2022-02-11	VK	bbbbbbb	Arm

clickstream

datetime	user_id	event	os	screen	video_id	install_channel	buy_type
2023-08-01 14:16:32	1253	video_start	IOS	SEARCH	10233		
2023-08-01 11:05:08	1278	install	ANDROID	MAIN		STORE	
2023-08-01 13:05:08	1278	buy	ANDROID	MAIN			TRIAL
2023-08-02 12:01:08	1278	video_start	ANDROID	COLLECTION	12412		
2023-08-03 14:05:08	1642	install	IOS	MAIN		LANDING	
2023-08-23 13:09:08	1278	buy	ANDROID	MAIN			PREMIUM

Написать SQL-запросы, позволяющие ответить на следующие вопросы.

1.1

Какая доля пользователей, установивших приложение в июне 2022 года, зарегистрировалась через электронную почту?



The screenshot shows a SQL query editor with an 'Input' tab and a 'Run SQL' button. The query is as follows:

```
SELECT
  (COUNT(*) /
   (SELECT
    COUNT(*)
    FROM user AS u1
    WHERE u1.created_date BETWEEN '01-06-2022' AND '30-06-2022'))
FROM user AS u2
WHERE (u2.created_date BETWEEN '01-06-2022' AND '30-06-2022') AND (u2.registration_type = 'EMAIL');
```

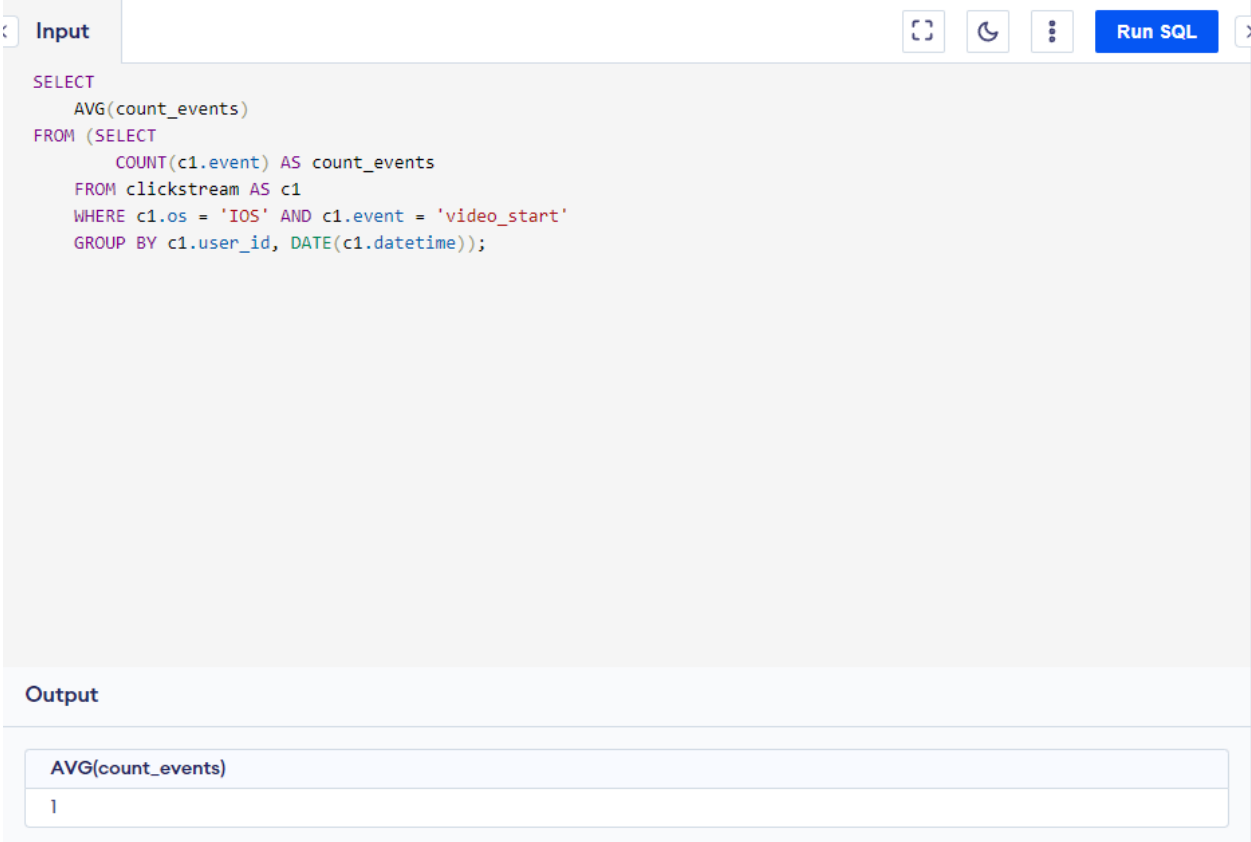
Below the query, the 'Output' section displays the result of the query:

(COUNT(*) / (SELECT COUNT(*) FROM user AS u1 WHERE u1.created_date BETWEEN '01-06-2022' AND '30-06-2022'))
0

В данном запросе мы делим кол-во пользователей, зарегистрировавшихся в июне 2022 года через email, на кол-во всех пользователей, зарегистрировавшихся в июне 2022 года, при помощи подселекта.

1.2

Сколько в среднем просмотров видео совершает в день один пользователь операционной системы iOS?



The screenshot shows a SQL query editor interface. The 'Input' tab is active, displaying a SQL query. The query calculates the average number of video starts per user per day for iOS users. The 'Run SQL' button is visible in the top right. Below the query, the 'Output' tab shows the result of the query, which is a single value: 1.

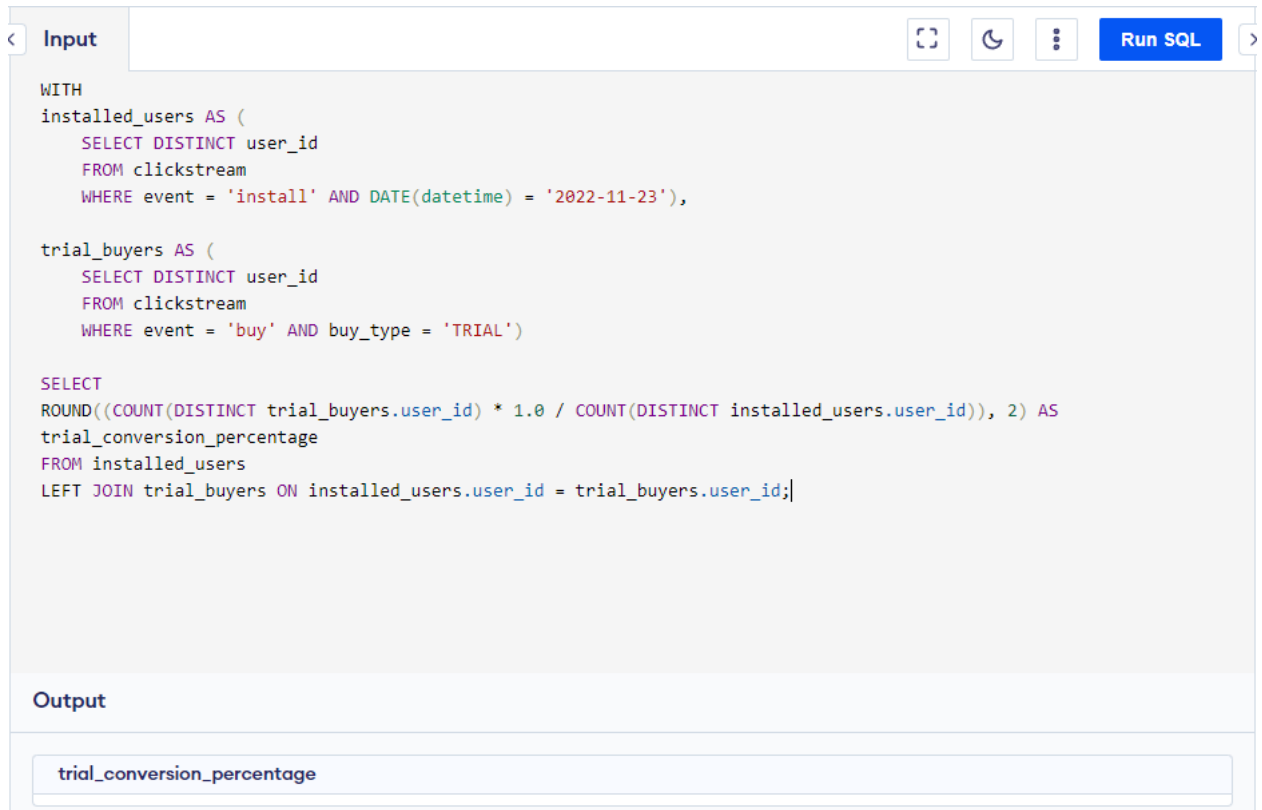
```
SELECT
  AVG(count_events)
FROM (SELECT
  COUNT(c1.event) AS count_events
FROM clickstream AS c1
WHERE c1.os = 'iOS' AND c1.event = 'video_start'
GROUP BY c1.user_id, DATE(c1.datetime));
```

AVG(count_events)
1

При помощи функции *AVG* ссылаясь на столбец *count_events* полученный при помощи подселекта получаем искомое число.

1.3

Какова конверсия из установки приложения в покупку триальной (пробной) подписки для пользователей, установивших приложение 23 ноября 2022 года?



The screenshot shows a SQL query editor with a tab labeled "Input". The query is as follows:

```
WITH
installed_users AS (
  SELECT DISTINCT user_id
  FROM clickstream
  WHERE event = 'install' AND DATE(datetime) = '2022-11-23'),

trial_buyers AS (
  SELECT DISTINCT user_id
  FROM clickstream
  WHERE event = 'buy' AND buy_type = 'TRIAL')

SELECT
ROUND((COUNT(DISTINCT trial_buyers.user_id) * 1.0 / COUNT(DISTINCT installed_users.user_id)), 2) AS
trial_conversion_percentage
FROM installed_users
LEFT JOIN trial_buyers ON installed_users.user_id = trial_buyers.user_id;
```

Below the query, there is an "Output" section with a table header "trial_conversion_percentage".

Если я правильно понял на основе определения (конверсия — это отношение пользователей, которые совершили нужное нам целевое действие, к общему числу пользователей), то нам необходимо поделить кол-во пользователей, купивших пробную версию (*trial_buyers*), на кол-во пользователей, установивших приложение в нужную нам дату (*installed_users*).

При помощи оператора *WITH*, который служит для создания временного табличного выражения, которое можно затем включить в SQL-запрос, создадим 2 таблицы, числитель и знаменатель, к которым мы будем обращаться в нашем уравнении после *SELECT*. Числитель домножаем на 1.0, чтобы перевести число в float, в случае если оно будет int, чтобы не потерять остаток от деления в дальнейшем (опять же, если я правильно понял, как работают арифметические операторы в SQL). Ну и округляем до 2х знаков после запятой.

В нашей таблице нет данных, удовлетворяющих заданному условию, но для проверки запроса, если изменить дату на '2023-08-01', то получим пользователя, который удовлетворяет данному условию (скрин ниже).

Input

[Run SQL](#)

```
WITH
installed_users AS (
  SELECT DISTINCT user_id
  FROM clickstream
  WHERE event = 'install' AND DATE(datetime) = '2023-08-01',

trial_buyers AS (
  SELECT DISTINCT user_id
  FROM clickstream
  WHERE event = 'buy' AND buy_type = 'TRIAL')

SELECT
ROUND((COUNT(DISTINCT trial_buyers.user_id) * 1.0 / COUNT(DISTINCT installed_users.user_id)), 2)
FROM installed_users
LEFT JOIN trial_buyers ON installed_users.user_id = trial_buyers.user_id;
```

Output

```
ROUND((COUNT(DISTINCT trial_buyers.user_id) * 1.0 / COUNT(DISTINCT installed_users.user_id)), 2)
```

```
1
```

1.4

Какой пользователь из Армении совершил больше всего просмотров видео на экране поиска?

Input

Run SQL

```
SELECT
  u.user_id,
  COUNT(c.event) AS search_video_views
FROM user AS u
JOIN clickstream AS c ON u.user_id = c.user_id
WHERE (u.country = 'Arm') AND (c.event = 'video_start') AND (c.screen = 'SEARCH')
GROUP BY u.user_id
ORDER BY search_video_views DESC
LIMIT 1;
```

Output

SQL query successfully executed. However, the result set is empty.

У нас таких пользователей нет. Но, если для проверки работоспособности запроса изменить страну на 'Rus', то получим результат.

Input

Run SQL

```
SELECT
  u.user_id,
  COUNT(c.event) AS search_video_views
FROM user AS u
JOIN clickstream AS c ON u.user_id = c.user_id
WHERE (u.country = 'Rus') AND (c.event = 'video_start') AND (c.screen = 'SEARCH')
GROUP BY u.user_id
ORDER BY search_video_views DESC
LIMIT 1;
```

Output

user_id	search_video_views
1253	1

1.5

Во сколько раз больше установок с лендингов в августе 2023 года относительно августа 2022 года?

Input

Run SQL

```
SELECT
  (SUM(CASE WHEN strftime('%m', datetime) = '08' AND strftime('%Y', datetime) = '2023' AND install_channel =
    'LANDING' THEN 1 ELSE 0 END) * 1.0 /
    SUM(CASE WHEN strftime('%m', datetime) = '08' AND strftime('%Y', datetime) = '2022' AND install_channel =
    'LANDING' THEN 1 ELSE 0 END)) AS landing_installs_ratio
FROM clickstream;
```

Output

landing_installs_ratio

К данному запросу я пришёл путём поиска ответов на просторах интернета, в модуле ничего подобного не объяснялось. Оператор *CASE WHEN*, аналогичен *if* в Python. Далее задаём условие, (учитывая, что я писал запросы в Online SQL Editor, который использует «диалект» SQL Lite, там вроде бы используется такой синтаксис обращения к дате и «вытаскивания» из неё нужного месяца/года и т.д.) если условие выполняется, то =1, если нет =0. Суммирует результат и домножаем на 1.0, по той же причине, что и в задании 1.3. Знаменатель высчитываем точно также. И по итогу должен получиться результат, но в таблице нет удовлетворяющих условию данных.

1.6

Какая доля пользователей Android имеет телефонный номер с кодом +7916?

Input

Run SQL

```
WITH
s1 AS (
  SELECT
    DISTINCT user_id
  FROM clickstream
  WHERE os = 'ANDROID'),

s2 AS (
  SELECT
    DISTINCT s1.user_id, u.contact
  FROM s1
  JOIN user AS u ON s1.user_id = u.user_id)

SELECT
  (SELECT
    COUNT(*)
  FROM s2
  WHERE s2.contact LIKE '+7916%') /
  COUNT(*) AS '+7916 users'
FROM s1
```

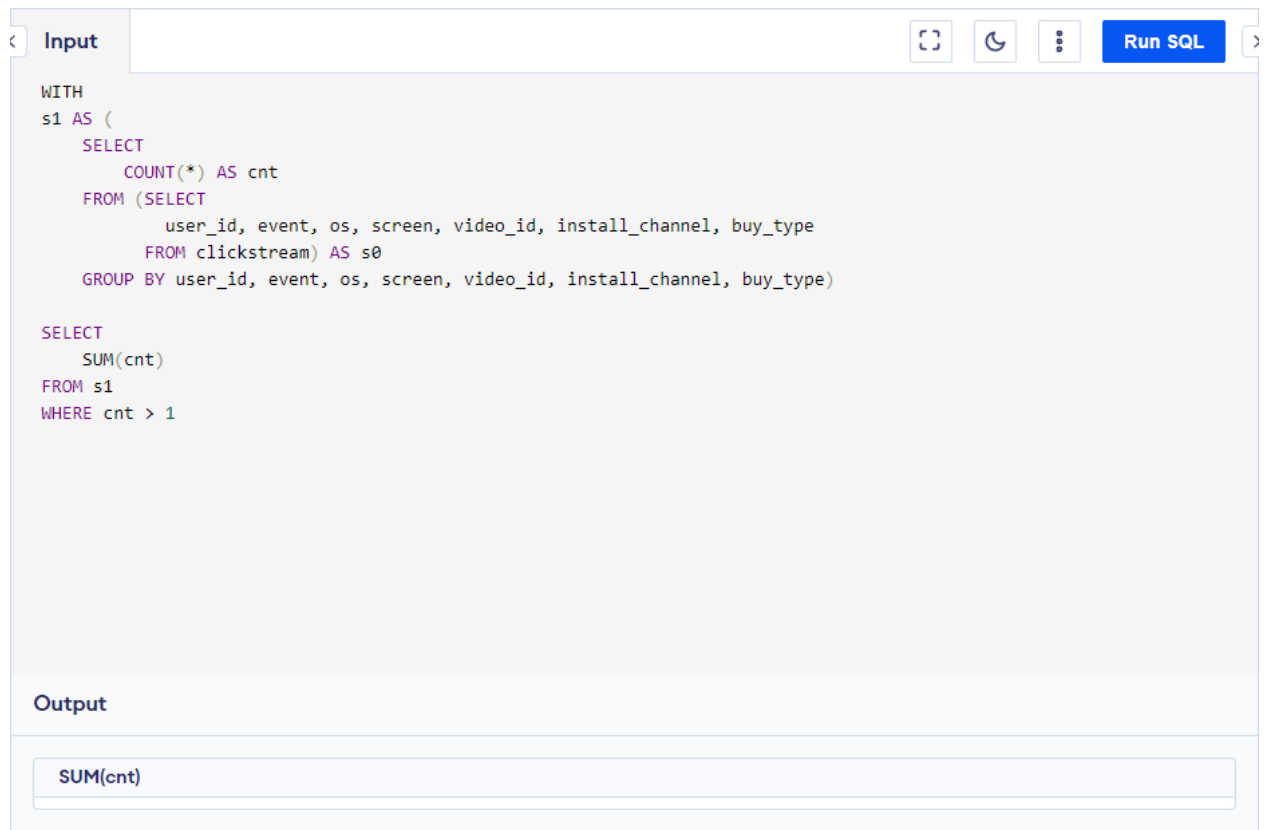
Output

+7916 users
0

Используя оператор *WITH*, создадим 2 переменные. В *s1* сохраним всех владельцев Android, а в *s2* те же пользователи, но с присоединённой информацией о номерах телефонов из таблицы *user*. Ну и далее, обращаясь к нашим временным переменным, получаем необходимую информацию.

1.7

Сколько дублей (одинаковые значения) есть в таблице **clickstream**?



The screenshot shows a SQL query editor with a tab labeled "Input". The query is as follows:

```
WITH
s1 AS (
  SELECT
    COUNT(*) AS cnt
  FROM (SELECT
        user_id, event, os, screen, video_id, install_channel, buy_type
      FROM clickstream) AS s0
  GROUP BY user_id, event, os, screen, video_id, install_channel, buy_type)

SELECT
  SUM(cnt)
FROM s1
WHERE cnt > 1
```

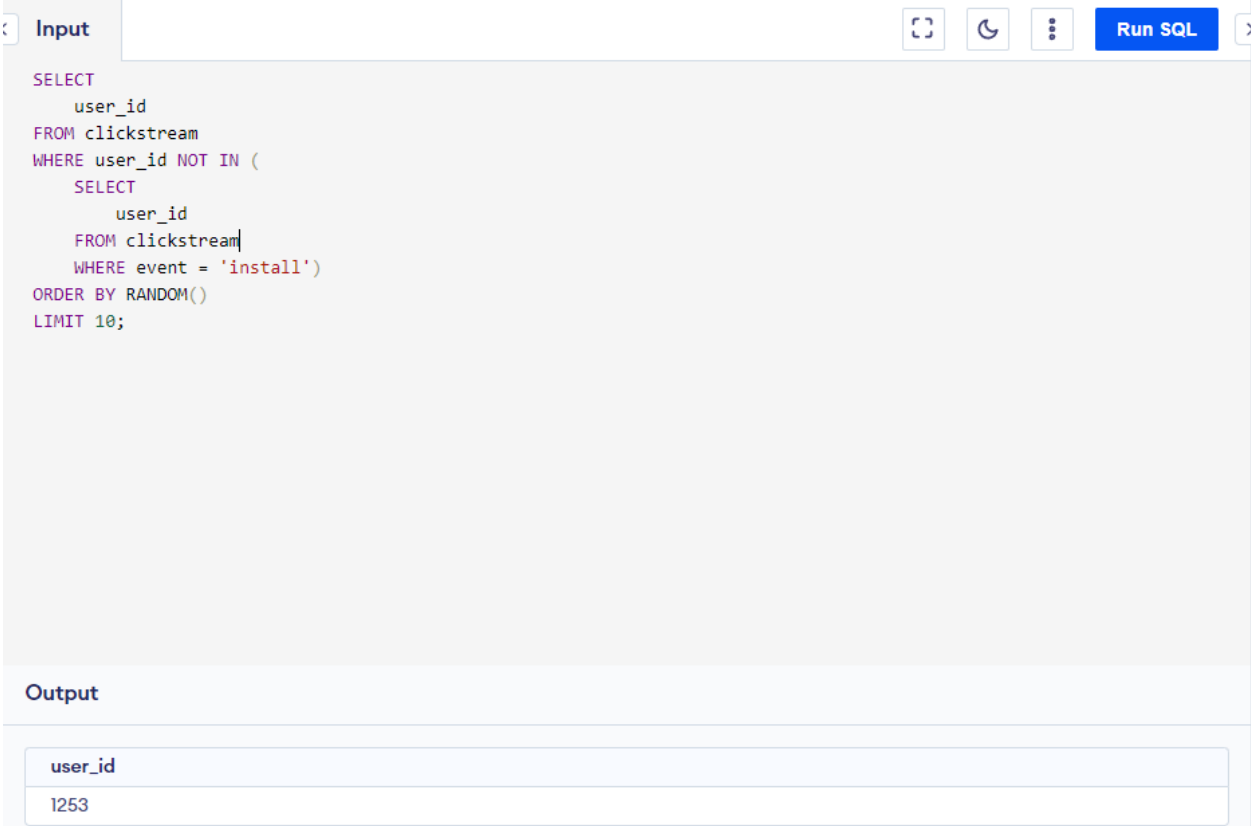
At the top right of the editor are icons for full screen, refresh, and a menu, along with a blue "Run SQL" button. Below the query editor is an "Output" section, which currently displays a single row with the value "SUM(cnt)".

Учитывая, что автор не указал, что именно считается дублями, я сделал так, как мне показалось логичным. За дубли мы примем полностью повторяющиеся строки, кроме даты. Чтобы можно было найти повторяющиеся операции у одно и того же пользователя, к примеру.

Посчитаем все значения по группировке, какие у нас только есть, обозначим это как *cnt*. А заключим это всё во временную таблицу *s1*, чтобы к ней можно было обратиться. Ну и как итог просуммируем дубли (если *cnt* > 1, то это дубль).

1.8

Найдите 10 случайных пользователей, по которым никогда не приходило событие установки.



The screenshot shows a SQL IDE interface with an 'Input' tab and a 'Run SQL' button. The SQL query is as follows:

```
SELECT
  user_id
FROM clickstream
WHERE user_id NOT IN (
  SELECT
    user_id
  FROM clickstream
  WHERE event = 'install')
ORDER BY RANDOM()
LIMIT 10;
```

Below the query editor, the 'Output' section displays a table with one column, 'user_id', and one row containing the value '1253'.

user_id
1253

Используя подселект получим таблицу пользователей, отвечающую заданному условию. Ну а за случайность вывода пользователей из этой таблицы отвечает *RANDOM*.

Задание 2

Составьте запрос, который поможет определить, какой из студентов получил больше двух пятерок и две двойки.

Task2

name	score
иванов	5
иванов	5
иванов	5
иванов	2
иванов	5
иванов	2
иванов	5
иванов	5
петров	5
петров	5
петров	2

Input

Run SQL

```
WITH
t5 AS (
  SELECT name, COUNT(score) AS count_5
  FROM task2
  WHERE score = 5
  GROUP BY name
),
t2 AS (
  SELECT name, COUNT(score) AS count_2
  FROM task2
  WHERE score = 2
  GROUP BY name
)

SELECT
  t5.name,
  t5.count_5,
  t2.count_2
FROM t5
JOIN t2 ON t5.name = t2.name
WHERE count_5 > 2 AND count_2 = 2
```

Output

name	count_5	count_2
иванов	6	2

Предварительно создав переменные с посчитанными оценками (отдельно для пятёрок и отдельно для двоек), объединяем их и задаём условие фильтрации через *WHERE*, чтобы получить необходимый ответ.

Задание 3

Напишите запрос, который поможет определить, сколько дублей (одинаковые значения) содержится в таблице **task2** из задания выше.

Задание решил выполнить по аналогии с заданием 1.7.

Наш промежуточный итог:

name	score	cnt
иванов	2	2
иванов	5	6
петров	2	1
петров	5	2

И итоговый запрос:

Input

WITH

c1 AS(SELECT

name,

score,

COUNT(*) AS cnt

FROM task2

GROUP BY name, score

HAVING 'count' > 1)

SELECT

SUM(cnt) as 'всего'

FROM c1

Output

всего
11

Задание 4

Есть таблица **Users** некой социальной сети. Нужно написать запрос, который найдет все случаи, в которых один юзер подписан на второго, а второй не подписан на него в ответ.

ID	ID_fol
1	2
1	3
2	1
3	2
3	1

Создадим отдельную таблицу *cross_sub* с взаимными подписками и вычтем её при помощи оператора *EXCEPT* из нашей исходной таблицы.

Input

Run SQL

```
WITH
cross_sub AS (SELECT
    u1.ID,
    u1.ID_fol,
    u2.ID_fol AS ID_fol_2
FROM users AS u1
INNER JOIN users AS u2 ON u1.ID_fol = u2.ID
WHERE u1.ID = ID_fol_2
)

SELECT *
FROM users

EXCEPT

SELECT ID, ID_fol]
FROM cross_sub
```

Output

ID	ID_fol
3	2