

**Задача 1.** Для таблиц audiobooks, audio\_cards, и listenings, с которыми работали на уроке

-- 1 вывести сколько пользователей добавили книгу 'Coraline', сколько пользователей прослушало больше 10%.

-- 2 по каждой операционной системе и тайтлу книги вывести кол-во пользователей, сумму прослушивания в часах. не учитывать тестовые прослушивания.

-- 3 найти книгу, которую слушает больше всего людей

-- 4 найти книгу, которую чаще всего дослушивают до конца

-- 5 найти среднее время между первым и вторым прослушиванием у пользователей. удалять тестовые чтения не нужно.

-- 6 для каждого тайтла книги вывести трех пользователей с минимальным временем между первым и последним прослушиваниями книги. удалять тестовые чтения нужно. Отсортировать по книге и разнице времени

**Задача 2.** В файле с клиентской информацией по одному клиенту может быть несколько строк, мы считаем актуальной строку, которая была последней (если сложно определить последнюю строку, оставляем любую строку).

Строки не являются полными дублями.

Написать запрос, который будет оставлять только одну строку для каждого ID\_клиента.

Структура таблицы (**Clients**), пример данных:

ID_клиента	Фамилия	Имя	Отчество	Дата рождения	Место работы	Должность	Еще какие-то поля
100	Иванова	Мария	Сергеевна	1970.01.01	ООО Березка	Бухгалтер	...
101	Сидоров	Иван	Иванович	1950.01.01	пусто	пусто	...
100	Петрова	Мария	Сергеевна	1970.01.01	ООО Березка	Бухгалтер	...
101	Сидоров	Иван	Иванович	1950.01.01	ОАО Привет	Инженер	...

Поле ID\_клиента должно быть уникальным, т.е. мы не можем вставить в таблицу две строки с одним ID\_клиента. Фамилия, Место работы и т.п. могут изменяться.

Приведенные в примере данные получены в исходных данных (находятся в файле или временной таблице).

Написать запрос, который возвращает по каждому клиенту только одну строку.

**Задача 3.**

Дана таблица Orders, Необходимо написать запрос для команды маркетинга, который выявит всех клиентов, кто купил iPhone, но не купил AirPods, чтобы отправить им маркетинговое предложение.

Таблица и пример данных даны ниже:

```
CREATE TABLE orders
(
  Customer_name
  Order_day DATE FORMAT 'DD-MMM-YY',
  Order_Id VARCHAR(10),
  Prod_Name VARCHAR(10),
  Qty INTEGER,
  Price INTEGER
)

Sample data:

Mahesh | Jan 1st | 01 | iPhone | 1 | 10 |
Mahesh | Jan 1st | 01 | iPad | 1 | 20 |
Mahesh | Jan 1st | 01 | AirPods | 1 | 20 |

Wayne | Jan 1st | 02 | iPhone | 1 | 10 |
Wayne | Jan 1st | 02 | shirt | 1 | 20 |
Wayne | Jan 1st | 02 | shoe | 1 | 20 |

-- By month, count of customers who only ordered iPhones but not AirPods in the same order.
```

#### Задача 4\*.

Дана таблица с историей подписки. Есть начало и конече периода подписки и статус. Нужно получить результат, как sample output . Условие для типа подписки так же дано.

customer_id	membership_start_date	membership_end_date	membership_status
114	2015-01-01	2015-02-15	Free
114	2015-02-15	2015-03-15	Paid
114	2015-03-15	2015-04-01	Non-member
114	2015-04-01	2015-10-01	Paid
114	2015-10-01	2016-01-01	Paid

Sample output:

customer_id	change_date	event
114	2015-01-01	WarmStart
114	2015-02-15	Convert
114	2015-03-15	Cancel
114	2015-04-01	ColdStart
114	2015-10-01	Renewal
114	2016-01-01	Cancel

Given the following labels of various transitions:

```
Free -> Paid: "Convert"
Paid -> Free: "ReverseConvert"
Paid -> Non-member: "Cancel"
Free -> Non-member: "Cancel"
Non-member -> Paid: "ColdStart"
Non-member -> Free: "WarmStart"
Paid -> Paid: "Renewal"
Free -> Free: "Renewal"
```