Задача 1 (Здесь по сути два задания).

Таблица представлена следующими полями:  
1) key  
2) id  
3) phone  
4) mail  
  
Тестовые данные:  
  
1;12345;89997776655;[test@mail.ru](mailto:test@mail.ru)  
2;54321;87778885566;[two@mail.ru](mailto:two@mail.ru)  
3;98765;87776664577;three@mail  
4; 66678;87778885566;[four@mail.ru](mailto:four@mail.ru)  
5; 34567;84547895566;[four@mail.ru](mailto:four@mail.ru)  
6; 34567;89087545678;[five@mail.ru](mailto:five@mail.ru)

drop table if exists t1 ;

CREATE TABLE if not exists t1 (

key int,

id int,

phone varchar,

mail varchar);

На основании заданного поля (это может быть id, phone, mail) получить все "связанные данные"

Например:  
если задать поиск по условию phone = 87778885566;  
  
Результат должен быть следующим:  
  
2;54321;87778885566;[two@mail.ru](mailto:two@mail.ru)  
4; 66678;87778885566;[four@mail.ru](mailto:four@mail.ru)  
5; 34567;84547895566;[four@mail.ru](mailto:four@mail.ru)  
6; 34567;89087545678;[five@mail.ru](mailto:five@mail.ru)

Задание следует отдельно выполнить на python и на SQL (по сути два задания). Диалект SQL можно использовать любой.

Задача 2.

1. Таблица **CLIENTS\_TABLE** (по клиентам)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *идентификатор клиента* | *ФИО клиента* | *ДР клиента* | *пол клиента* | *остальные параметры* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLIENT\_ID** | **CLIENT\_NAME** | **BIRTHDAY** | **GENDER** | **…** |
| … | … | … | … | … |

1. Таблица **LOANS\_TABLE** (по договорам)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *номер договора* | *идентификатор клиента* | *дата договора* | *сумма по договору* | *остальные параметры* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LOAN\_ID** | **CLIENT\_ID** | **LOAN\_DATE** | **LOAN\_AMOUNT** | **…** |
| … | … | … | … | … |

Каждый клиент может обращаться в компанию несколько раз, соответственно в базе может храниться информация по нескольким договорам на одного клиента.

Договор, оформленный клиентом у нас впервые, будем называть **первым** договором; договор, оформленный после – **вторым**; далее – **третьим**; и так далее.

Необходимо **написать SQL запрос** к базе для представления его результатов в сводной таблице вида:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Количество **1** договоров, оформленных в 2020 | Количество **2** договоров, оформленных в 2020 | Количество **3** договоров, оформленных в 2020 | Количество **4** договоров, оформленных в 2020 | … |
| Мужчины |  |  |  |  |  |
| Женщины |  |  |  |  |  |

Диалект SQL можно использовать любой.

Тестовые данные:

MSSQL

drop table LOANS\_TABLE;

CREATE TABLE LOANS\_TABLE (

LOAN\_ID int,

CLIENT\_ID int,

LOAN\_DATE date,

LOAN\_AMOUNT float);

drop table CLIENTS\_TABLE;

CREATE TABLE CLIENTS\_TABLE (

CLIENT\_ID int,

CLIENT\_NAME NVARCHAR(20),

BIRTHDAY date,

GENDER NVARCHAR(20));

INSERT INTO CLIENTS\_TABLE

VALUES

(1, 'bob', '20200115', 'male'),

(2, 'rocky', '20200215', 'female'),

(3, 'like', '20200215', 'female'),

(4, 'ricky', '20200215', 'male');

INSERT INTO LOANS\_TABLE

VALUES

(1, 1, '20200115', 10000),

(2, 2, '20200215', 20000),

(3, 3, '20200315', 30000),

(4, 4, '20200415', 40000),

(5, 1, '20200116', 15000),

(6, 2, '20200315', 35000),

(7, 3, '20200315', 5000),

(8, 1, '20200115', 1500),

(9, 2, '20200115', 500),

(10, 1, '20200115', 1500);

MYSQL

drop table if EXISTS LOANS\_TABLE;

CREATE TABLE if not exists LOANS\_TABLE (

LOAN\_ID int,

CLIENT\_ID int,

LOAN\_DATE date,

LOAN\_AMOUNT float);

drop table if EXISTS CLIENTS\_TABLE;

CREATE TABLE if not exists CLIENTS\_TABLE (

CLIENT\_ID int,

CLIENT\_NAME NVARCHAR(20),

BIRTHDAY date,

GENDER NVARCHAR(20));

postgreSQL

drop table if EXISTS loans\_table;

CREATE TABLE if not exists loans\_table (

LOAN\_ID int,

CLIENT\_ID int,

LOAN\_DATE date,

LOAN\_AMOUNT float);

drop table if EXISTS clients\_table;

CREATE TABLE if not exists clients\_table (

CLIENT\_ID int,

CLIENT\_NAME VARCHAR(20),

BIRTHDAY date,

GENDER VARCHAR(20));