

Table of Contents

- [1. solution](#)
 - [1.1. create table](#)
 - [1.2. ChatGPT support](#)
 - [1.2.1. request 1](#)
 - [1.2.2. answer 1](#)
 - [1.2.3. request 2](#)
 - [1.2.4. answer 2](#)
 - [1.2.5. answer 3 - combination of request 1 and request 2](#)
 - [1.2.6. request 4](#)
 - [1.2.7. answer 4](#)
 - [1.3. test answer 1 - failed](#)
 - [1.4. test answer 3 - success](#)
 - [1.5. test answer 4 - success](#)
- [2. final solution](#)
 - [2.1. explanation](#)

Есть таблица TABLE APPLICATIONS, содержащая все заявки на кредиты со всех банков, то есть каждый раз, когда клиент приходит в какой-нибудь банк, в таблице появляется строка.

Список полей таблицы:

- Applicat_ID – уникальный идентификатор заявки
- Applicat_Date – Дата заявки с точностью до секунды
- Bank_Name – название банка
- Passport_Number – номер паспорта клиента

Задание. Посещение банков.

- Вывести паспорта всех клиентов, у которых в жизни была следующая ситуация: клиент пришел в Альфабанк, а до этого никогда не посещал ХоумКредит.

1. solution

Другими словами это задание: Вывести уникальные passport_number у которых есть запись Bank_Name равная 'Альфабанк' но у других записей с этим passport_number, у которых Applicat_Date меньше, Bank_Name != 'ХоумКредит'.

Нужно составить SQL select запрос.

Here is a SQL table

1.1. create table

```
DROP TABLE table_applications;

CREATE TABLE table_applications (
  Applicat_ID INTEGER PRIMARY KEY, -- уникальный идентификатор посещения
  Applicat_Date INTEGER, -- epoch -- Дата посещения
  Bank_Name TEXT, -- название банка
  Passport_Number VARCHAR(30) -- номер паспорта клиента
);

INSERT INTO table_applications VALUES(NULL, 2147483640, 'Альфабанк', '999999900');
INSERT INTO table_applications VALUES(NULL, 2147483645, 'ХоумКредит', '999999900');

INSERT INTO table_applications VALUES(NULL, 2147483639, 'ХоумКредит', '0733223445');
INSERT INTO table_applications VALUES(NULL, 2147483640, 'ХоумКредит', '0733223445');
INSERT INTO table_applications VALUES(NULL, 2147483641, 'Альфабанк', '0733223445');
INSERT INTO table_applications VALUES(NULL, 2147483645, 'ХоумКредит', '0733223445');

INSERT INTO table_applications VALUES(NULL, 2147483641, 'Альфабанк', '1133223411');
INSERT INTO table_applications VALUES(NULL, 2147483647, 'Альфабанк', '1133223411');

SELECT * FROM table_applications;
```

1.2. ChatGPT support

1.2.1. request 1

```
Here is a SQL table
CREATE TABLE table_applications (
  Applicat_ID INTEGER PRIMARY KEY, -- уникальный идентификатор посещения
  Applicat_Date INTEGER, -- epoch -- Дата посещения
  Bank_Name TEXT, -- название банка
  Passport_Number VARCHAR(30) -- номер паспорта клиента
);

Задание: Вывести паспорта всех клиентов, которые приходили в Альфабанк, а до этого никогда не
посещали
ХоумКредит.

Другими словами это задание: Вывести уникальные passport_number у которых есть запись
Bank_Name равная 'Альфабанк' но у других записей с этим
passport_number, у которых Applicat_Date меньше, Bank_Name != 'ХоумКредит'.

Составь SQL select запрос. Объяснить как если бы ты его составлял поэтапно с логической цепочкой
рассуждений.
```

1.2.2. answer 1

Чтобы выполнить данное задание, нужно последовательно следовать данным требованиям с использованием подзапросов:

1. Начнем с выбора всех уникальных номеров паспортов (passport_number), у которых есть запись с `Bank_Name` равным 'Альфабанк':

```
SELECT DISTINCT Passport_Number
FROM table_applications
WHERE Bank_Name = 'Альфабанк';
```

1. Далее нам нужно проверить, что для этих номеров паспортов нет записей, у которых `Bank_Name` равно 'ХоумКредит' и `Applicat_Date` меньше, чем у записей для 'Альфабанк'. Для этого используем подзапрос в операторе NOT IN:

```
SELECT DISTINCT Passport_Number
FROM table_applications
WHERE Bank_Name = 'Альфабанк'
AND Passport_Number NOT IN (
    SELECT Passport_Number
    FROM table_applications
    WHERE Bank_Name = 'ХоумКредит'
    AND Applicat_Date <
        (SELECT MIN(Applicat_Date)
         FROM table_applications
         WHERE Bank_Name = 'Альфабанк'
         AND Passport_Number = table_applications.Passport_Number)
);
```

```sql

```

Этот запрос позволит вывести уникальные номера паспортов клиентов, которые посещали Альфабанк, но не посещали ХоумКредит.

Таким образом, вы поэтапно создаете логическую цепочку запросов, чтобы выбрать нужную информацию из базы данных.

1.2.3. request 2

What approach can you suggest to build this SELECT request, dont build it yourself but explain how would you select approach to build it?

1.2.4. answer 2

To build the SELECT request to achieve the desired outcome, you would need to use a combination of subqueries, joins, and the GROUP BY clause. Here is an outline of the approach:

1. Use a subquery to select the passport numbers of clients who visited 'Альфабанк'.
2. Join the table with itself based on the passport numbers to find other records with the same passport number.
3. Filter the joined results to include only those where the Applicat_Date is less than the visited date at 'Альфабанк' and the Bank_Name is not 'ХоумКредит'.
4. Group the results by passport number to get unique passport numbers.

By following this approach, you can identify and retrieve the passport numbers of clients who visited 'Альфабанк' but never visited 'ХоумКредит' before.

1.2.5. answer 3 - combination of request 1 and request 2

1.2.6. request 4

Упрости этот запрос, чтобы SELECT вы выполнял подзапросы, чтобы не было подциклов.

1.2.7. answer 4

Для упрощения запроса и избежания подциклов можно использовать JOIN вместо подзапросов. Вот упрощенный запрос:

```
SELECT DISTINCT t1.Passport_Number FROM table_applications t1 LEFT JOIN table_applications t2 ON
t1.Passport_Number = t2.Passport_Number AND t2.Applicat_Date < t1.Applicat_Date AND t2.Bank_Name =
'ХоумКредит' WHERE t1.Bank_Name = 'Альфабанк' AND t2.Passport_Number IS NULL;
```

Этот запрос использует LEFT JOIN для соединения таблицы table_applications саму с собой по условиям, указанным в JOIN. Затем он фильтрует строки, где соединение не удалось (т.е. где нет соответствующей записи в таблице t2), и выводит уникальные Passport_Number из заявок в Альфабанк, которые не имеют соответствующих заявок в ХоумКредит с более ранней датой.

1.3. test answer 1 - failed

```
SELECT DISTINCT Passport_Number
FROM table_applications
WHERE Bank_Name = 'Альфабанк'
AND Passport_Number NOT IN (
```

```

SELECT Passport_Number
FROM table_applications
WHERE Bank_Name = 'ХоумКредит'
AND Applicat_Date <
(SELECT MIN(Applicat_Date)
FROM table_applications
WHERE Bank_Name = 'Альфабанк'
AND Passport_Number = table_applications.Passport_Number)
);

```

Passport_Number
999999900
733223445
1133223411

Fail. Не верно.

1.4. test answer 3 - success

```

SELECT DISTINCT t1.Passport_Number
FROM table_applications t1
WHERE t1.Bank_Name = 'Альфабанк'
AND NOT EXISTS (
SELECT 1
FROM table_applications t2
WHERE t2.Passport_Number = t1.Passport_Number
AND t2.Applicat_Date < t1.Applicat_Date
AND t2.Bank_Name = 'ХоумКредит'
);

```

Success! Успех!

Select 1 - возвращает константу

```

SELECT 1
FROM table_applications t2
WHERE t2.Bank_Name = 'ХоумКредит'

```

1
1
1
1

1.5. test answer 4 - success

```

SELECT DISTINCT t1.Passport_Number
FROM table_applications t1
LEFT JOIN table_applications t2
ON t1.Passport_Number = t2.Passport_Number
AND t2.Applicat_Date < t1.Applicat_Date
AND t2.Bank_Name = 'ХоумКредит'
WHERE t1.Bank_Name = 'Альфабанк'
AND t2.Passport_Number IS NULL;

```

Passport_Number
999999900
1133223411

```

SELECT DISTINCT *
FROM table_applications t1
LEFT JOIN table_applications t2
ON t1.Passport_Number = t2.Passport_Number
AND t2.Applicat_Date < t1.Applicat_Date
AND t2.Bank_Name = 'ХоумКредит'
-- WHERE t1.Bank_Name = 'Альфабанк'
-- AND t2.Passport_Number IS NULL;

```

Applicat_ID	Applicat_Date	Bank_Name	Passport_Number	Applicat_ID	Applicat_Date	Bank_Name	Passport_Nu
1	2147483640	Альфабанк	999999900				
2	2147483645	ХоумКредит	999999900				
3	2147483639	ХоумКредит	733223445				
4	2147483640	ХоумКредит	733223445	3	2147483639	ХоумКредит	733223445
5	2147483641	Альфабанк	733223445	3	2147483639	ХоумКредит	733223445
5	2147483641	Альфабанк	733223445	4	2147483640	ХоумКредит	733223445
6	2147483645	ХоумКредит	733223445	3	2147483639	ХоумКредит	733223445

Applicat_ID	Applicat_Date	Bank_Name	Passport_Number	Applicat_ID	Applicat_Date	Bank_Name	Passport_Nu
6	2147483645	ХоумКредит	733223445	4	2147483640	ХоумКредит	73327
7	2147483641	Альфабанк	1133223411				
8	2147483647	Альфабанк	1133223411				

Попробуем оптимизировать запрос, чтобы уменьшить талицу t2

```
SELECT DISTINCT t1.Passport_Number
FROM table_applications t1
LEFT JOIN (
  SELECT Passport_Number, MIN(Applicat_Date) AS Max_Applicat_Date
  FROM table_applications
  WHERE Bank_Name = 'ХоумКредит'
  GROUP BY Passport_Number
) t2 ON t1.Passport_Number = t2.Passport_Number AND t2.Max_Applicat_Date < t1.Applicat_Date
WHERE t1.Bank_Name = 'Альфабанк' AND t2.Passport_Number IS NULL;
```

Passport_Number
999999900
1133223411

Подзапрос из test6:

```
SELECT Passport_Number, MIN(Applicat_Date) AS Max_Applicat_Date
FROM table_applications
WHERE Bank_Name = 'ХоумКредит'
GROUP BY Passport_Number
```

Passport_Number	Max_Applicat_Date
733223445	2147483639
999999900	2147483645

Как видим t2 стал меньше, что уменьшело вычисления в JOIN операции

2. final solution

Хотя индексы не могут быть использоваты в подзапросах, этот подзапрос агрегирует записи в MIN(Applicat_Date) пред JOIN, за счет чего выполняется максимально эффективно.

```
SELECT DISTINCT t1.Passport_Number
FROM table_applications t1
LEFT JOIN (
  SELECT Passport_Number, MIN(Applicat_Date) AS Max_Applicat_Date
  FROM table_applications
  WHERE Bank_Name = 'ХоумКредит'
  GROUP BY Passport_Number
) t2 ON t1.Passport_Number = t2.Passport_Number AND t2.Max_Applicat_Date < t1.Applicat_Date
WHERE t1.Bank_Name = 'Альфабанк' AND t2.Passport_Number IS NULL;
```

Passport_Number
999999900
1133223411

2.1. explanation

Это запрос выполняет LEFT JOIN на записях из оригинальной таблицы и подзапроса, который выбирает запись Bank_Name = 'ХоумКредит' с MIN(Applicat_Date). Запись добавленная в правую часть, означает, что клиент посещал банк 'ХоумКредит' до этого.

Объединение выбирает записи без правой части, чтобы отобрать клиентов не посещавших 'ХоумКредит' до посещения 'Альфабанк'.

Последним этапом происходит выбор уникальных записей по Passport_number.