Иванушкин Иван Александрович

anvi.inlae@gmail.com

DB BROWSER SQLite

Дано

Имеется таблица проводок по счетам, отражающая факты начисления или списания денежных средств по счетам клиента. Состоит из столбцов:

BUSINESS_DT - дата проводки

ACCOUNT_DEBIT_ID - идентификатор счета дебита, с которого списываются средства

ACCOUNT_CREDIT_ID - идентификатор счета кредита, на который начисляются средства

POSTING_AMT - сумма проводки в рублях. т.е. сумма средств, которая списывается со счета дебита и зачисляется на счет кредита. Значение всегда положительное.

Заполненная таблица:

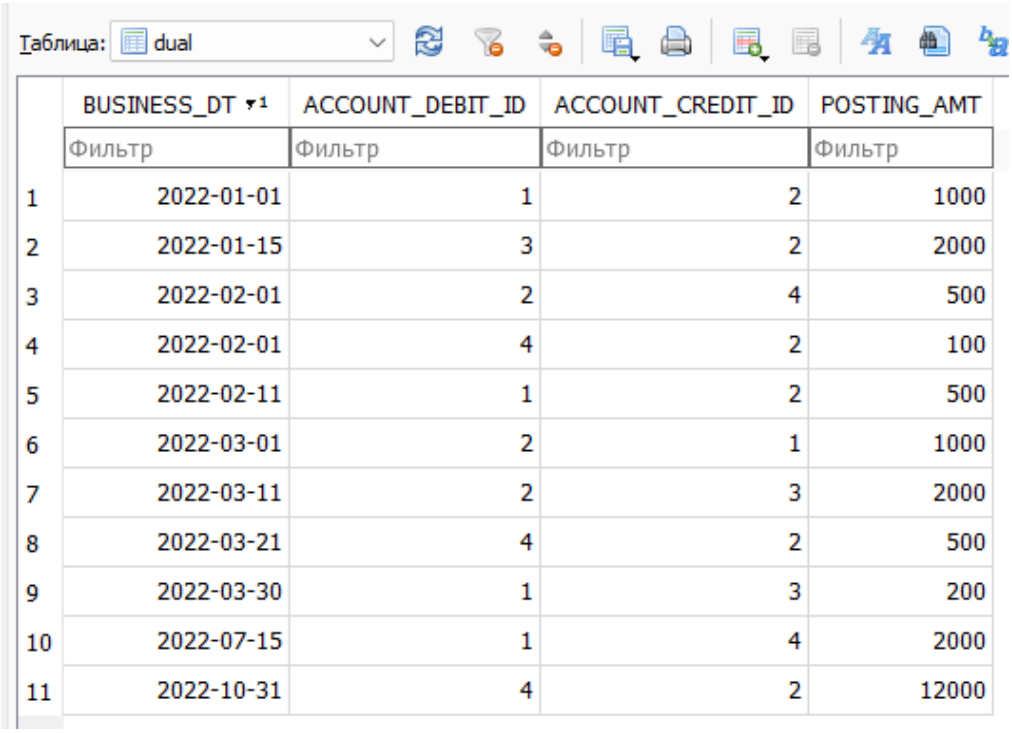
Таблица: dual

	BUSINESS_DT	ACCOUNT_DEBIT_ID	ACCOUNT_CREDIT_ID	POSTING_AMT
	Фильтр	Фильтр	Фильтр	Фильтр
1	2022-01-01	1	2	1000
2	2022-01-15	3	2	2000
3	2022-02-01	2	4	500
4	2022-02-01	4	2	100
5	2022-02-11	1	2	500
6	2022-03-01	2	1	1000
7	2022-03-11	2	3	2000
8	2022-03-21	4	2	500
9	2022-03-30	1	3	200
10	2022-07-15	1	4	2000
11	2022-10-31	4	2	12000

Задание 1

Написать SQL-запрос, который на основании таблицы проводок рассчитает остатки по счетам на текущую дату. Начальные остатки по всем счетам полагаем равными нулю. Таблица должна состоять из следующих столбцов:

Account_ID – идентификатор счета

Current_Date – дата, на которую посчитан остаток

Account_Balance – остаток на счете, может быть отрицательным.

Решение

1. Вычислил в табличных выражениях debit и credit сумму ушедших и зачисленных денег соответственно.

1.1. Использовал функцию sum и партицию, чтоб найти участие каждого уникального счёта в проводках без уменьшения строк.

2. Под «*остатки по счетам на текущую дату*» допустил, что *дата текущая* = сейчас (now). Но можно подставить любую. В таблицу добавил даты из будущего для проверки.

3. В основном запросе вычел сумму зачислений от суммы убытка, соединив табличные выражения через ID счёта. Ниже будет скриншот для удобочитаемости кода.

```
WITH
debit (ACCOUNT_DEBIT_ID, D_BALANCE) AS (
SELECT DISTINCT ACCOUNT_DEBIT_ID,  SUM(POSTING_AMT) OVER (PARTITION BY ACCOUNT_DEBIT_ID) as D_BALANCE
FROM dual
WHERE BUSINESS_DT <= DATE('now') ),


credit (ACCOUNT_CREDIT_ID, C_BALANCE) AS (
SELECT DISTINCT ACCOUNT_CREDIT_ID,  SUM(POSTING_AMT) OVER (PARTITION BY  ACCOUNT_CREDIT_ID) as C_BALANCE
FROM dual
WHERE BUSINESS_DT <= DATE('now') )



SELECT ACCOUNT_DEBIT_ID AS 'Account_ID',  C_BALANCE - D_BALANCE AS 'Account_Balance', DATE('now') as 'Current_Date'
FROM debit INNER JOIN credit ON ACCOUNT_DEBIT_ID = ACCOUNT_CREDIT_ID
```

Скриншот из программы

SQL 1

```
WITH
debit (ACCOUNT_DEBIT_ID, D_BALANCE) AS (
SELECT DISTINCT ACCOUNT_DEBIT_ID,  SUM(POSTING_AMT) OVER (PARTITION BY  ACCOUNT_DEBIT_ID) as D_BALANCE
FROM dual
WHERE BUSINESS_DT <= DATE('now') ),

credit (ACCOUNT_CREDIT_ID, C_BALANCE) AS (
SELECT DISTINCT ACCOUNT_CREDIT_ID,  SUM(POSTING_AMT) OVER (PARTITION BY  ACCOUNT_CREDIT_ID) as C_BALANCE
FROM dual
WHERE BUSINESS_DT <= DATE('now') )


SELECT ACCOUNT_DEBIT_ID AS 'Account_ID',  C_BALANCE - D_BALANCE AS 'Account_Balance', DATE('now') as 'Current_Date'
FROM debit INNER JOIN credit ON ACCOUNT_DEBIT_ID = ACCOUNT_CREDIT_ID
```

	Account_ID	Account_Balance	Current_Date
1	1	-2700	2022-07-30
2	2	600	2022-07-30
3	3	200	2022-07-30
4	4	1900	2022-07-30

Задание 2

Написать SQL-запрос, который выводит таблицу остатков за весь период, за который нам известны проводки (например, с января по март). Начальные остатки по всем счетам полагаем равными нулю. Таблица должна состоять из следующих столбцов:

Account_ID – идентификатор счета

Business_From_DT – дата, начиная с которой сформировался указанный остаток на счете.

Business_To_DT – дата, по которую на счете находится указанный остаток.

Если остаток действует по текущий момент (дата следующего изменения остатка не известна), то поле следует заполнить датой 9999-12-31.

Account_Balance – остаток на счете Account_ID в период с Business_From_DT по Business_To_DT.

Например, на счете 2 был остаток в 1000 рублей, начиная с 01.01.2020 по 04.01.2020 включительно (5го числа остаток уже поменялся).

Решение

Задача выглядит идентично первой.

1. В табличном выражении debit_date нахожу первую дату появления каждого счёта в колонке ACCOUNT_DEBIT_ID (из него переводили).

2. В табличном выражении credit_date нахожу первую дату появления каждого счёта в колонке ACCOUNT_CREDIT_ID (на него переводили).

я не нашёл способа для себя, как можно было бы сделать первые 2 действия рациональнее

3. В основном запросе для вывода Business_From_DT нахожу минимум из возвращаемых значений дат из табличных выражений debit_date и credit_date.

4. В основном запросе для вывода Business_To_DT нахожу максимум из возвращаемых значений дат из табличных выражений debit_date и credit_date.

5. Расчёт баланса счёта идентичен коду из первого задания.

Добавленный код:

```
debit_date(deb_id, deb_from, deb_to) AS (
SELECT DISTINCT ACCOUNT_DEBIT_ID , MIN(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_DEBIT_ID) AS 'deb_from',
                            MAX(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_DEBIT_ID) AS 'deb_to'
FROM dual  ),

credit_date(cre_id, cre_from, cre_to) AS (
SELECT DISTINCT ACCOUNT_CREDIT_ID , MIN(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_CREDIT_ID) AS 'cre_from',
                            MAX(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_CREDIT_ID) AS 'cre_to'
FROM dual  )
```

Скриншот из программы:

```
16  debit_date(deb_id, deb_from, deb_to) AS (
17  SELECT DISTINCT ACCOUNT_DEBIT_ID , MIN(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_DEBIT_ID) AS 'deb_from',
18                                     MAX(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_DEBIT_ID) AS 'deb_to'
19  FROM dual
20  ),
21
22  credit_date(cre_id, cre_from, cre_to) AS
23  (
24  SELECT DISTINCT ACCOUNT_CREDIT_ID , MIN(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_CREDIT_ID) AS 'cre_from',
25                                     MAX(BUSINESS_DT) OVER (PARTITION BY ACCOUNT_CREDIT_ID) AS 'cre_to'
26  FROM dual
27  )
28
29
30  SELECT ACCOUNT_DEBIT_ID AS 'Account_ID', MIN(deb_from, cre_from) AS 'Business_From_DT',
31                                          MAX(deb_to, cre_to) AS 'Business_To_DT',
32                                              C_BALANCE - D_BALANCE AS 'Account_Balance'
33  FROM debit INNER JOIN credit ON ACCOUNT_DEBIT_ID = ACCOUNT_CREDIT_ID
34                  INNER JOIN debit_date ON ACCOUNT_DEBIT_ID = deb_id
35                      INNER JOIN credit_date ON ACCOUNT_CREDIT_ID = cre_id
36
```

	Account_ID	Business_From_DT	Business_To_DT	Account_Balance
1	1	2022-01-01	2022-07-15	-2700
2	2	2022-01-01	2022-10-31	600
3	3	2022-01-15	2022-03-30	200
4	4	2022-02-01	2022-10-31	1900