

## ПРОЕКТНОЕ ЗАДАНИЕ № 2.3

### Корректировка данных в rd.account\_balance

#### Условие:

Витрина dm.account\_balance\_turnover отображает изменение баланса счетов по дням. Она строится на основе следующих источников:

- rd.account — информация по счетам клиентов;
- rd.account\_balance — информация с номерами счетов и суммами на начало и конец дня;
- dm.dict\_currency — справочник валют.

В ходе проверки были выявлены несоответствия между:

- account\_out\_sum текущего дня и
- account\_in\_sum следующего дня.

#### Цель:

1. Определить корректное значение поля account\_in\_sum.
2. Определить корректное значение поля account\_out\_sum.
3. Обновить данные в rd.account\_balance, используя полученные правила.
4. Написать процедуру загрузки данных в витрину dm.account\_balance\_turnover.
5. Провести проверку после обновлений.

---

#### 1. Проверка и корректировка account\_in\_sum

Если account\_in\_sum не совпадает с account\_out\_sum предыдущего дня, то корректным считается account\_out\_sum.

#### SELECT

```
curr.account_rk,  
curr.effective_date,  
curr.account_in_sum AS wrong_in_sum,  
prev.account_out_sum AS correct_in_sum  
FROM rd.account_balance curr  
JOIN rd.account_balance prev  
  ON curr.account_rk = prev.account_rk  
  AND curr.effective_date = prev.effective_date + INTERVAL '1 day'  
WHERE curr.account_in_sum IS DISTINCT FROM prev.account_out_sum;
```

SELECT curr.account_rk, curr.effective_date, curr.account_in_su					Введите SQL выражение чтобы отфильтровать				
Таблица	123 account_rk	effective_date	123 wrong_in_sum	123 correct_in_sum					
Текст	1	2 000 536	2023-01-16	55 993	55 978				
	2	2 000 536	2023-01-19	171 708	171 701				
	3	2 003 113	2023-01-11	70 885	70 870				
	4	2 003 113	2023-01-14	226 321	226 302				
	5	2 003 113	2023-01-17	309 347	309 342				
	6	2 003 113	2023-01-20	434 905	434 894				
	7	2 003 113	2023-01-23	591 646	591 629				
	8	2 003 278	2023-01-16	52 457	52 451				
	9	2 003 278	2023-01-19	134 200	134 195				
	10	2 003 703	2023-01-22	89 005	88 992				
	11	2 005 911	2023-01-29	100 664	100 649				

## 2. Проверка и корректировка account\_out\_sum

Если account\_out\_sum предыдущего дня не совпадает с account\_in\_sum текущего дня, то корректным считается account\_in\_sum.

**SELECT**

prev.account\_rk,  
prev.effective\_date,  
prev.account\_out\_sum AS wrong\_out\_sum,  
curr.account\_in\_sum AS correct\_out\_sum

**FROM** rd.account\_balance prev

**JOIN** rd.account\_balance curr

**ON** curr.account\_rk = prev.account\_rk

**AND** curr.effective\_date = prev.effective\_date + **INTERVAL** '1 day'

**WHERE** prev.account\_out\_sum **IS DISTINCT FROM** curr.account\_in\_sum;

SELECT prev.account_rk, prev.effective_date, prev.account_out					Введите SQL выражение чтобы отфильтровать рез				
Таблица	123 account_rk	effective_date	123 wrong_out_sum	123 correct_out_sum					
Текст	1	2 000 536	2023-01-15	55 978	55 993				
	2	2 000 536	2023-01-18	171 701	171 708				
	3	2 003 113	2023-01-10	70 870	70 885				
	4	2 003 113	2023-01-13	226 302	226 321				
	5	2 003 113	2023-01-16	309 342	309 347				
	6	2 003 113	2023-01-19	434 894	434 905				
	7	2 003 113	2023-01-22	591 629	591 646				
	8	2 003 278	2023-01-15	52 451	52 457				
	9	2 003 278	2023-01-18	134 195	134 200				
	10	2 003 703	2023-01-21	88 992	89 005				
	11	2 005 911	2023-01-28	100 649	100 664				
запись	12	2 005 911	2023-01-31	211 655	211 673				

### 3. Обновление данных в rd.account\_balance

#### *3.1 корректировка account\_in\_sum (по данным предыдущего дня)*

```
UPDATE rd.account_balance curr
SET account_in_sum = prev.account_out_sum
FROM rd.account_balance prev
WHERE curr.account_rk = prev.account_rk
      AND curr.effective_date = prev.effective_date + INTERVAL '1 day'
      AND curr.account_in_sum IS DISTINCT FROM prev.account_out_sum;
```

#### *3.2 корректировка account\_out\_sum (по данным следующего дня)*

```
UPDATE rd.account_balance curr
SET account_out_sum = next.account_in_sum
FROM rd.account_balance next
WHERE curr.account_rk = next.account_rk
      AND curr.effective_date = next.effective_date - INTERVAL '1 day'
      AND curr.account_out_sum IS DISTINCT FROM next.account_in_sum;
```

### 4. Процедура загрузки витрины

```
CREATE OR REPLACE PROCEDURE dm.fill_account_balance_turnover()
LANGUAGE plpgsql
AS $$
BEGIN
    TRUNCATE TABLE dm.account_balance_turnover;

    INSERT INTO dm.account_balance_turnover (
        account_rk,
        currency_name,
        department_rk,
        effective_date,
        account_in_sum,
        account_out_sum
    )
    SELECT
        ab.account_rk,
        dc.currency_name,
        a.department_rk,
        ab.effective_date,
        ab.account_in_sum,
        ab.account_out_sum
    FROM rd.account_balance ab
    JOIN rd.account a
        ON ab.account_rk = a.account_rk
        AND ab.effective_date BETWEEN a.effective_from_date AND
a.effective_to_date
```

```

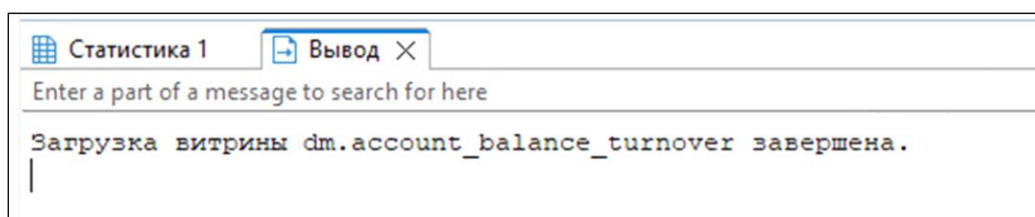
JOIN dm.dict_currency dc
  ON a.currency_cd = dc.currency_cd
  AND ab.effective_date BETWEEN dc.effective_from_date AND
dc.effective_to_date;

RAISE NOTICE 'Загрузка витрины dm.account_balance_turnover
завершена.';
END;
$$;

```

Запуск процедуры:

```
CALL dm.fill_account_balance_turnover();
```



## 5. Проверка после обновлений

### 5.1 Проверка account\_in\_sum

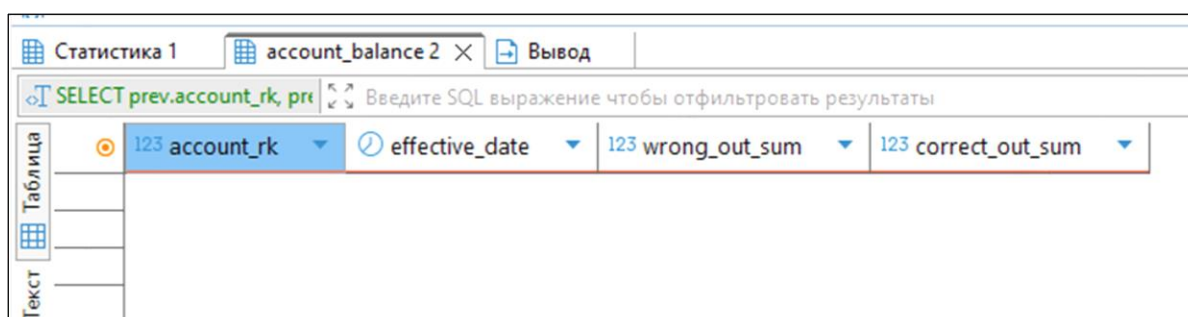
```

SELECT
  curr.account_rk,
  curr.effective_date,
  curr.account_in_sum AS wrong_in_sum,
  prev.account_out_sum AS correct_in_sum
FROM rd.account_balance curr
JOIN rd.account_balance prev
  ON curr.account_rk = prev.account_rk
  AND curr.effective_date = prev.effective_date + INTERVAL '1 day'
WHERE curr.account_in_sum IS DISTINCT FROM prev.account_out_sum;

```

**Результат:**

Нет выведенных строк



## 5.2 Проверка account\_out\_sum

```
SELECT
    prev.account_rk,
    prev.effective_date,
    prev.account_out_sum AS wrong_out_sum,
    curr.account_in_sum AS correct_out_sum
FROM rd.account_balance prev
JOIN rd.account_balance curr
    ON curr.account_rk = prev.account_rk
    AND curr.effective_date = prev.effective_date + INTERVAL '1 day'
WHERE prev.account_out_sum IS DISTINCT FROM curr.account_in_sum;
```

### Результат:

Нет выведенных строк

---

### Выводы:

Обе выборки не вернули строк, что подтверждает корректность значений account\_in\_sum и account\_out\_sum после обновления.