Прототип витрины

=+	~ ° ~		~ SQL			
	account_rk bigint	currency_name text	department_rk bigint	effective_date date	account_in_sum numeric	account_out_sum numeric
1	2943625	RUB	110	2023-01-28	0	25961
2	2943625	RUB	110	2023-01-29	25961	85286
3	2943625	RUB	110	2023-01-30	85286	104895
4	2943625	RUB	110	2023-01-31	104909	115761
5	2943625	RUB	110	2023-02-01	115761	176854
6	2943625	RUB	110	2023-02-02	176854	227253
7	2313904	RUB	97	2023-01-16	0	60561
8	2313904	RUB	97	2023-01-17	60561	126967
9	2313904	RUB	97	2023-01-18	126967	130976
10	2313904	RUB	97	2023-01-19	130990	158993
11	2313904	RUB	97	2023-01-20	158993	218762
12	2313904	RUB	97	2023-01-21	218762	287118
13	2005911	RUB	112	2023-01-26	0	7903
14	2005911	RUB	112	2023-01-27	7903	72538
15	2005911	RUB	112	2023-01-28	72538	100649
16	2005911	RUB	112	2023-01-29	100664	156415
17	2005911	RUB	112	2023-01-30	156415	199465
18	2005911	RUB	112	2023-01-31	199465	211655
19	2005911	RUB	112	2023-02-01	211673	222068
20	2005911	RUB	112	2023-02-02	222068	233226
21	2005911	RUB	112	2023-02-03	233226	246555
77 Tota	2005011 rows: 1000 of 1	RUR 10575 Query co	112 omplete 00:00:00.1	2023-02-04 46 Ln 70, Col	246564	266172

2 запроса на выбор некорректных данных

```
--корректная сумма выходной вчерашний остаток

SELECT ab.account_rk, ab.effective_date, ab.account_out_sum, ab2.effective_date, ab2.account_in_sum, ab.account_out_sum AS correct_sum

FROM rd.account_balance ab

JOIN rd.account_balance ab2

ON ab2.effective_date = ab.effective_date + INTERVAL '1 day'

AND ab.account_rk = ab2.account_rk

WHERE ab2.account_in_sum <> ab.account_out_sum

ORDER BY ab.account_rk, correct_sum

--корректная сумма входной сегодняшний остаток

SELECT ab.account_balance ab

JOIN rd.account_balance ab2

ON ab2.effective_date = ab.effective_date + INTERVAL '1 day'

AND ab.account_rb = ab2.account_rk

WHERE ab2.account_in_sum <> ab2.account_rk

WHERE ab2.account_in_sum <> ab3.account_rk

WHERE ab2.account_rk = ab2.account_rk

WHERE ab2.account_rk, correct_sum
```

Результат первого запроса, где правильный остаток вчерашнего дня

=+		~		~	Î	8	<u>+</u>	~	5QL						
	account_rk bigint		â	effective_date date		account_out_sum numeric		effective_date date	account_in_sum numeric	correct_sum numeric					
1		20	00053	36	2023	-01-15				55978	2023-01-16		5	5993	5597
2		20	0005	36	2023	-01-18				171701	2023-01-19		17	1708	17170
3		20	0031	13	2023	-01-10				70870	2023-01-11		7(0885	7087
4		20	0031	13	2023	-01-13				226302	2023-01-14		22	5321	22630
5		20	0031	13	2023	-01-16				309342	2023-01-17		309	9347	30934
5		20	0031	13	2023	-01-19				434894	2023-01-20		434	4905	43489
7		20	0031	13	2023	-01-22				591629	2023-01-23		59	1646	59162
8		20	00327	78	2023	-01-15				52451	2023-01-16		5	2457	5245
9		20	00327	78	2023	-01-18				134195	2023-01-19		134	4200	13419
10		20	00370	03	2023	-01-21				88992	2023-01-22		8	9005	8899
11		20	0059	11	2023	-01-28				100649	2023-01-29		100	0664	10064
12		20	0059	11	2023	-01-31				211655	2023-02-01		21	1673	21165
13		20	0059	11	2023	-02-03				246555	2023-02-04		24	6564	24655
14		20	00724	18	2023	-01-17				104023	2023-01-18		104	4027	10402
15		20	00724	18	2023	-01-20				263599	2023-01-21		269	3608	26359
16		20	00724	18	2023	-01-23				331243	2023-01-24		33	1252	33124
17		20	00764	10	2023	-01-24				132351	2023-01-25		133	2363	13235
18		20	00764	10	2023	-01-27				177291	2023-01-28		177	7306	17729
19		20	00764	10	2023	-01-30				268530	2023-01-31		26	8541	26853
20		20	00764	10	2023	-02-02				306531	2023-02-03		30	6550	30653
21		20	00810)7	2023	-01-19				99376	2023-01-20		9	9393	9937
22		20	00810)7	2023	-01-22				246090	2023-01-23		24	5092	24609
23		20	00810)7	2023	-01-25				421190	2023-01-26		42	1207	42119
24		20	00810)7	2023-01-28		2023-01-28			550846	2023-01-29		550	0860	55084
25		20	00810)7	2023-01-31		686032			2023-02-01		686041		68603	
26		20	0083	11	2023-01-19				56494	2023-01-20		56501		5649	
27		20	0083	11	2023-01-22		224860			2023-01-23		224	4864	22486	
28		20	0083	11	2023	-01-25				352137	2023-01-26		35	2151	35213
29		20	00832	24	2023	-01-17				191802	2023-01-18		19	1805	19180
30		20	00832	24	2023	-01-20				231433	2023-01-21		23	1442	23143

На скрине видим, что два последних столбца различаются

Результат второго запроса, где правильный остаток сегодняшнего дня

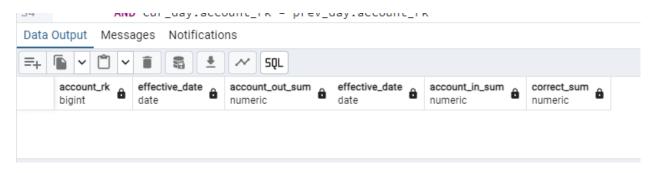
=+	• · · ·		✓ SQL			
	account_rk bigint	effective_date date	account_out_sum numeric	correct_sum numeric	effective_date date	account_in_sum numeric
1	2000536	2023-01-15	55978	55993	2023-01-16	55993
2	2000536	2023-01-18	171701	171708	2023-01-19	171708
3	2003113	2023-01-10	70870	70885	2023-01-11	70885
4	2003113	2023-01-13	226302	226321	2023-01-14	226321
5	2003113	2023-01-16	309342	309347	2023-01-17	309347
6	2003113	2023-01-19	434894	434905	2023-01-20	434905
7	2003113	2023-01-22	591629	591646	2023-01-23	591646
В	2003278	2023-01-15	52451	52457	2023-01-16	52457
9	2003278	2023-01-18	134195	134200	2023-01-19	134200
10	2003703	2023-01-21	88992	89005	2023-01-22	89005
11	2005911	2023-01-28	100649	100664	2023-01-29	100664
12	2005911	2023-01-31	211655	211673	2023-02-01	211673
13	2005911	2023-02-03	246555	246564	2023-02-04	246564
14	2007248	2023-01-17	104023	104027	2023-01-18	104027
15	2007248	2023-01-20	263599	263608	2023-01-21	263608
16	2007248	2023-01-23	331243	331252	2023-01-24	331252
17	2007640	2023-01-24	132351	132363	2023-01-25	132363
18	2007640	2023-01-27	177291	177306	2023-01-28	177306
19	2007640	2023-01-30	268530	268541	2023-01-31	268541
20	2007640	2023-02-02	306531	306550	2023-02-03	306550
21	2008107	2023-01-19	99376	99393	2023-01-20	99393

На этом скрине видно, что корректная сумма отличается от выходной суммы вчерашнего дня

Процедуры изменения для остатков

```
26 --update, где остаток вчерашнего дня является корректным, а входной сегодняшнего дня нет
27 V CREATE OR REPLACE PROCEDURE account_balance_correct_out_sum()
    LANGUAGE plpgsql
29
     AS $$
30
     BEGIN
31
         UPDATE rd.account_balance AS cur_day
32
         SET account_in_sum = prev_day.account_out_sum
33
         FROM rd.account_balance prev_day
         WHERE cur_day.effective_date = prev_day.effective_date + INTERVAL '1 day'
34
35
             AND cur_day.account_rk = prev_day.account_rk
36
             AND cur_day.account_in_sum <> prev_day.account_out_sum;
37
     END;
38
     $$;
39
40
     --update, где остаток сегодняшнего дня является корректным, а выходной вчерашнего дня нет
42 • CREATE OR REPLACE PROCEDURE account_balance_correct_in_sum()
43
    LANGUAGE plpgsql
44
     AS $$
45
    BEGIN
46
         UPDATE rd.account_balance AS prev_day
47
         SET account_out_sum = cur_day.account_in_sum
48
        FROM rd.account_balance cur_day
         WHERE cur_day.effective_date = prev_day.effective_date + INTERVAL '1 day'
49
50
             AND cur_day.account_rk = prev_day.account_rk
51
             AND cur_day.account_in_sum <> prev_day.account_out_sum;
52
     END;
53 $$;
```

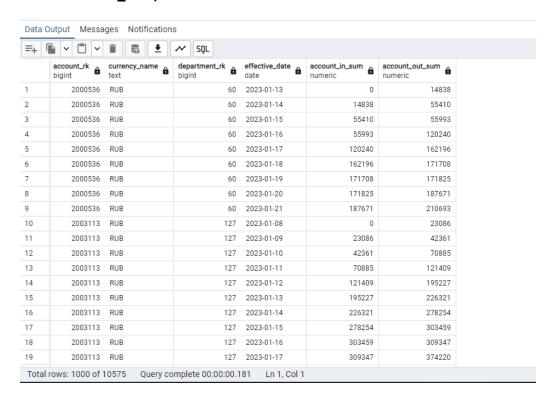
Вызов первой процедуры (результат выборки корректных остатков по прошлому дню)



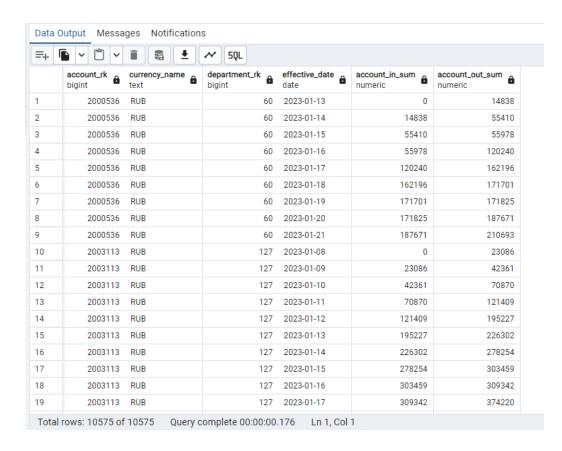
Процедура обновления данных в таблице balance_turnover

```
75
     CREATE OR REPLACE PROCEDURE insert_balance_turnover()
76
     LANGUAGE plpgsql
77
     AS $$
   BEGIN
78
79
         TRUNCATE TABLE dm.account_balance_turnover;
80
81 🕶
        INSERT INTO dm.account_balance_turnover (
            account_rk,
82
83
            currency_name,
            department_rk,
84
85
            effective_date,
86
            account_in_sum,
87
            account_out_sum
88
        )
89
        SELECT a.account_rk,
            COALESCE(dc.currency_name, '-1'::TEXT) AS currency_name,
90
            a.department_rk,
91
            ab.effective_date,
92
            ab.account_in_sum,
93
94
            ab.account_out_sum
95 FROM rd.account a
96 LEFT JOIN rd.account_balance ab ON a.account_rk = ab.account_rk
    LEFT JOIN dm.dict_currency dc ON a.currency_cd = dc.currency_cd;
97
98 END;
99
     $$;
```

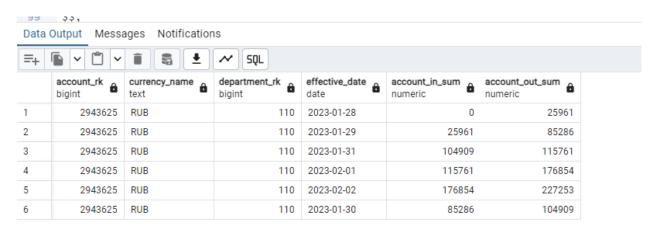
Витрина после сборки с новыми данными (здесь in_sum является корректным, поменяли out sum)



Витрина после сборки с новыми данными (здесь out_sum является корректным, поменяли in_sum)

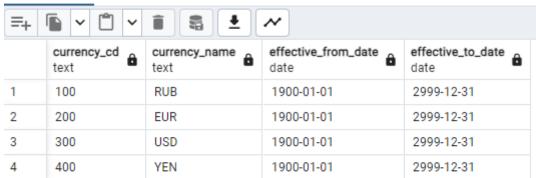


Результат по одному из ключей (испугался, что данные разные, но просто улетела дата вниз, 30 есть и правильное)



До загрузки данных в таблицу dict_currency

Data Output Messages Notifications



Загруженные данные в таблицу dict_currency

Data Output Messages Notifications

=+	=+ □ □<									
	currency_cd [PK] text	currency_name /	effective_from_date /	effective_to_date date						
1	100	RUB	1900-01-01	2999-12-31						
2	200	EUR	1900-01-01	2999-12-31						
3	300	USD	1900-01-01	2999-12-31						
4	400	YEN	1900-01-01	2999-12-31						
5	500	KZT	1900-01-01	2999-12-31						