

ПРОЕКТНОЕ ЗАДАНИЕ № 2.2

Анализ и восстановление данных витрины dm.loan_holiday_info

Условие:

Витрина dm.loan_holiday_info содержит информацию о кредитных каникулах, сделках и продуктах банка. Были выявлены пропуски по определенным датам эффективности (effective_from_date).

Цель:

- Проанализировать текущее состояние витрины и её источников.
- Выявить, по каким датам отсутствуют записи.
- Определить, требуется ли полная или частичная перегрузка.
- Выполнить загрузку недостающих данных.
- Перегрузить витрину и проверить корректность.

1. Анализ текущего состояния

Кол-во строк в таблицах:

```
SELECT count(*) FROM dm.loan_holiday_info /;
```

```
-- Результат:  
-- 10040
```

```
SELECT count(*) FROM rd.deal_info;
```

```
-- Результат:  
-- 10000
```

```
SELECT count(*) FROM rd.loan_holiday /;
```

```
-- Результат:  
-- 10000
```

```
SELECT count(*) FROM rd.product;
```

```
-- Результат:  
-- 10000
```

Количества сопоставимы, но возможны расхождения по датам.

2. Группировка по effective_from_date

- Витрина

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM dm.loan_holiday_info
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-08-11  3502
-- 2023-03-15  3500
-- 2023-01-01  3000
```

- Сделки

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.deal_info
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-08-11  3500
-- 2023-01-01  3000
```

- Каникулы

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.loan_holiday
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-08-11  3500
-- 2023-03-15  3500
-- 2023-01-01  3000
```

- Продукты

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.product
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-03-15  3500
```

Выявленные проблемы:

- В rd.deal_info отсутствуют строки на дату 2023-03-15.
 - В rd.product отсутствуют строки на даты 2023-01-01 и 2023-08-11.
-

3. Загрузка недостающих данных

Для восстановления данных выполним дозагрузку недостающих записей из временных таблиц rd.tmp_deal_info и rd.tmp_product

- 3.1 Дозагрузка данных в rd.deal_info

Анализ временной таблицы rd.tmp_deal_info показал наличие недостающих данных по дате эффективности 2023-03-15:

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.tmp_deal_info
GROUP BY effective_from_date;
```

-- Результат:

-- 2023-03-15 3500

Выполнена частичная дозагрузка данных:

```
INSERT INTO rd.deal_info (
    deal_rk, deal_num, deal_name, deal_sum, client_rk, account_rk, agreement_rk,
    deal_start_date, department_rk, product_rk, deal_type_cd, effective_from_date,
    effective_to_date
)
SELECT
    deal_rk, deal_num, deal_name, deal_sum, client_rk, account_rk, agreement_rk,
    deal_start_date, department_rk, product_rk, deal_type_cd, effective_from_date,
    effective_to_date
FROM rd.tmp_deal_info
WHERE effective_from_date = DATE '2023-03-15';
```

Повторная проверка после загрузки:

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.deal_info
GROUP BY effective_from_date;
```

-- Результат:

-- 2023-08-11 3500

-- 2023-03-15 3500

-- 2023-01-01 3000

Все три ключевые даты теперь представлены в таблице rd.deal_info

- 3.2 Дозагрузка данных в rd.product

Предварительный анализ временной таблицы rd.tmp_product показал наличие всех трех необходимых дат:

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM rd.tmp_product
GROUP BY effective_from_date;
```

-- Результат:

-- 2023-08-11 3500

-- 2023-03-15 3500

-- 2023-01-01 3000

В основную таблицу rd.product были загружены недостающие строки (даты 2023-01-01 и 2023-08-11):

```
INSERT INTO rd.product (  
    product_rk, product_name, effective_from_date, effective_to_date)  
SELECT  
    product_rk, product_name, effective_from_date, effective_to_date  
FROM rd.tmp_product  
WHERE effective_from_date IN ('2023-01-01', '2023-08-11');
```

Повторная проверка после загрузки:

```
SELECT effective_from_date, COUNT(*) AS cnt  
FROM rd.product  
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-08-11  3500  
-- 2023-03-15  3500  
-- 2023-01-01  3000
```

Все три ключевые даты теперь представлены в таблице rd.product

4. Перегрузка витрины dm.loan_holiday_info

Для обновления витрины была выполнена процедура dm.fill_loan_holiday_info, которая перезаполняет таблицу на основе актуальных данных из источников rd.deal_info, rd.loan_holiday и rd.product.

```
CREATE OR REPLACE PROCEDURE dm.fill_loan_holiday_info()  
LANGUAGE plpgsql  
AS $$  
BEGIN
```

```
    TRUNCATE TABLE dm.loan_holiday_info;
```

```
    INSERT INTO dm.loan_holiday_info (  
        deal_rk,
```

```
        effective_from_date,  
        effective_to_date,
```

```
        agreement_rk,  
        client_rk,
```

```
        department_rk,  
        product_rk,
```

```
        product_name,  
        deal_type_cd,
```

```
        deal_start_date,  
        deal_name,
```

```
        deal_number,  
        deal_sum,
```

```
loan_holiday_type_cd,  
loan_holiday_start_date,  
loan_holiday_finish_date,  
loan_holiday_fact_finish_date,  
loan_holiday_finish_flg,  
loan_holiday_last_possible_date  
)
```

SELECT

```
d.deal_rk,  
lh.effective_from_date,  
lh.effective_to_date,  
d.agreement_rk,  
d.client_rk,  
d.department_rk,  
d.product_rk,  
p.product_name,  
d.deal_type_cd,  
d.deal_start_date,  
d.deal_name,  
d.deal_num,  
d.deal_sum,  
lh.loan_holiday_type_cd,  
lh.loan_holiday_start_date,  
lh.loan_holiday_finish_date,  
lh.loan_holiday_fact_finish_date,  
lh.loan_holiday_finish_flg,  
lh.loan_holiday_last_possible_date
```

FROM rd.deal_info d

LEFT JOIN rd.loan_holiday lh

ON d.deal_rk = lh.deal_rk

AND d.effective_from_date = lh.effective_from_date

LEFT JOIN rd.product p

ON p.product_rk = d.product_rk

AND p.effective_from_date = d.effective_from_date;

RAISE NOTICE 'Загрузка в dm.loan_holiday_info завершена.';

END;

\$\$;

Запуск процедуры:

CALL dm.fill_loan_holiday_info();

5. Проверка результатов

Для валидации загрузки был выполнен подсчет строк по effective_from_date в витрине:

```
SELECT effective_from_date, COUNT(*) AS cnt
FROM dm.loan_holiday_info
GROUP BY effective_from_date;
```

-- Результат:

```
-- 2023-08-11  3522
-- 2023-03-15  3510
-- 2023-01-01  3008
```

Перегрузка прошла успешно - прирост данных подтверждён, данные по отсутствующим датам восстановлены.

6. Проверка на дубликаты в витрине dm.loan_holiday_info

```
SELECT *
FROM dm.loan_holiday_info
GROUP BY
    deal_rk,
    effective_from_date,
    effective_to_date,
    agreement_rk,
    account_rk,
    client_rk,
    department_rk,
    product_rk,
    product_name,
    deal_type_cd,
    deal_start_date,
    deal_name,
    deal_number,
    deal_sum,
    loan_holiday_type_cd,
    loan_holiday_start_date,
    loan_holiday_finish_date,
    loan_holiday_fact_finish_date,
    loan_holiday_finish_flg,
    loan_holiday_last_possible_date
HAVING COUNT(*) > 1;
```

Результат:

В результате проверки выявлено 8 полных дубликатов строк в витрине m.loan_holiday_info.

Так как в задании отсутствуют инструкции по их удалению, было принято решение **оставить дубликаты без изменений.**

Выводы:

1. Причина отсутствия данных:

Установлено, что пропуски в витрине dm.loan_holiday_info были вызваны **неполной загрузкой данных в источники** - таблицы rd.deal_info и rd.product - по датам эффективности:

2023-01-01, 2023-03-15, 2023-08-11.

2. Способ восстановления данных:

Для устранения проблемы была выполнена **частичная дозагрузка недостающих строк** из временных таблиц rd.tmp_deal_info и rd.tmp_product, с последующей вставкой только по необходимым датам.

3. Перегрузка витрины:

После обновления источников была **успешно выполнена перегрузка витрины** dm.loan_holiday_info с помощью хранимой процедуры fill_loan_holiday_info. Объем данных в витрине стал соответствовать ожидаемому.