**Отчет по ДЗ 04 Внедрение и жизнь с Data Vault**

**1. Добавление новой строки с новым бизнес-ключом**

1. Добавил новую строку в «source\_customers.csv» с новым id

105,Donald,Trump,d.trump345@usa.gov

2. Выполнил:

«««bash

dbt seed

dbt run

«««

3. Результаты:

- В «stg\_customers» появилась новая запись

- В «hub\_customer» добавился новый хаб с этим бизнес-ключом

- В «sat\_customer» добавилась новая запись с данными клиента

**2. Добавление строки с существующим бизнес-ключом, но другими свойствами**

1. Добавил запись в « source\_customers.csv « с существующим бизнес ключем, но другими именами

106,Donald1,Trump1,d.trump345@usa.gov

2. Выполнил:

«««bash

dbt seed

dbt run

«««

3. Результаты:

- В «stg\_customers» появилась новая версия записи

- В «hub\_customer» изменений нет (ключ уже существует)

- В «sat\_customer» добавилась новая версия записи с обновленными данными

**3. Удаление строки из CSV**

1. Удалил одну запись из «source\_customers.csv»

2. Выполнил:

«««bash

dbt seed

dbt run

«««

3. Результаты:

- В «stg\_customers» запись исчезла

- В «hub\_customer» запись осталась (Data Vault сохраняет историю)

- В «sat\_customer» последняя версия записи осталась без изменений

**4. Витрина динамики количества заказов по неделям**

Создал модель «marts.orders\_weekly\_trend»:

«««sql

with weekly\_orders as (

select

date\_trunc('week', order\_date) as week\_start,

count(\*) as order\_count

from {{ ref('sat\_order') }}

group by 1

order by 1

)

select

week\_start,

order\_count,

lag(order\_count) over (order by week\_start) as prev\_week\_count,

order\_count - lag(order\_count) over (order by week\_start) as week\_over\_week\_change,

(order\_count::float / lag(order\_count) over (order by week\_start) - 1) \* 100 as week\_over\_week\_change\_pct

from weekly\_orders

«««

Результат показывает:

- Количество заказов по неделям

- Изменение количества заказов по сравнению с предыдущей неделей

- Процентное изменение

**5. Витрина клиентов по количеству завершенных заказов**

Создал модель «marts.customers\_completed\_orders»:

«««sql

with completed\_orders as (

    select

        hc.customer\_pk,

        count(distinct ho.order\_pk) as completed\_orders\_count

    from {{ ref('hub\_customer') }} hc

    join {{ ref('link\_customer\_order') }} lco on hc.customer\_pk = lco.customer\_pk

    join {{ ref('hub\_order') }} ho on lco.order\_pk = ho.order\_pk

    join {{ ref('sat\_order') }} sod on ho.order\_pk = sod.order\_pk

    where sod.status = 'completed'

    group by 1

)

select

    co.customer\_pk,

    sc.first\_name,

    sc.last\_name,

    sc.email,

    co.completed\_orders\_count

from completed\_orders co

join {{ ref('stg\_customers') }} sc on co.customer\_pk = sc.customer\_pk

order by co.completed\_orders\_count desc

«««

Результат:

- Список клиентов, отсортированный по количеству завершенных заказов

- Для каждого клиента показаны его данные и количество заказов

**Отчет о выполнении**

В ходе выполнения домашнего задания:

1. Успешно развернул проект Data Vault с использованием dbt и Docker

2. Исследовал поведение модели при:

- Добавлении новых бизнес-ключей

- Обновлении атрибутов существующих ключей

- Удалении записей из источников

3. Создал две аналитические витрины:

- Динамика заказов по неделям

- Рейтинг клиентов по завершенным заказам