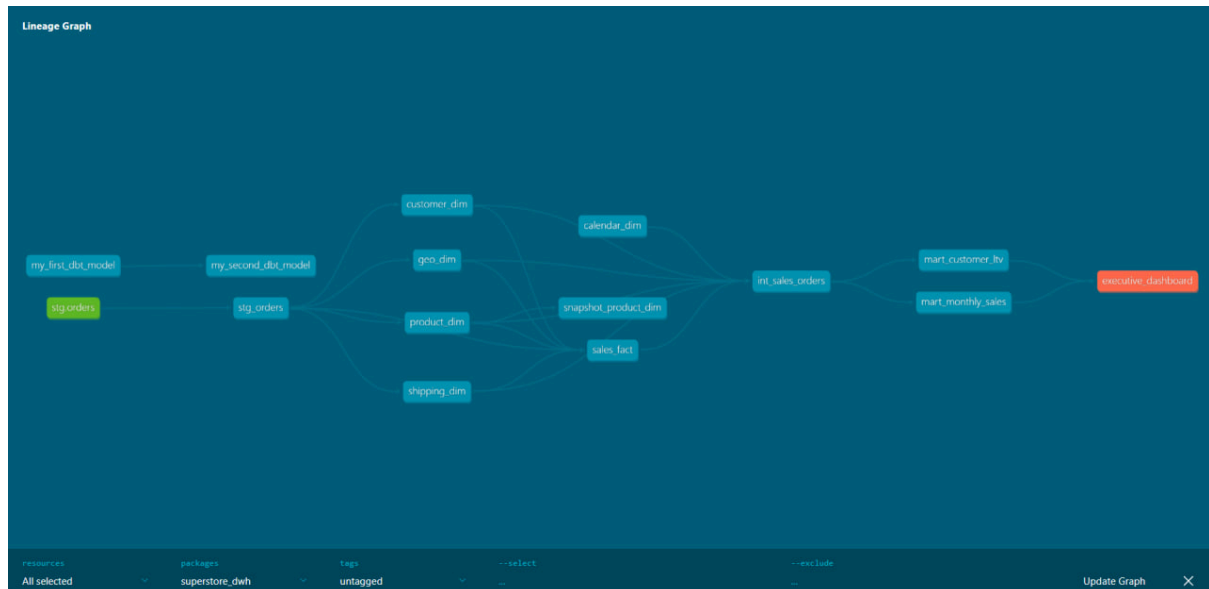


Проект: superstore\_dwh

Выполнил(а): Бурлов Василий Тимофеевич

Группа: БД-251м

**Скриншот графа зависимостей (lineage) из dbt docs, на котором видны связи от int\_sales\_orders к вашим витринам:**



**Код промежуточной модели int\_sales\_orders.sql:**

```
-- models/intermediate/int_sales_orders.sql
-- Эта модель объединяет факты со всеми измерениями, создавая
широкую, денормализованную таблицу для легкого использования в
витринах.
```

```
SELECT
```

```
-- Ключи
f.order_id,
```

```
-- Измерения из customer_dim
```

```
c.customer_id,
c.customer_name,
```

```
-- Измерения из product_dim
```

```
p.product_id,
p.product_name,
p.category,
p.sub_category,
p.segment,
```

```
-- Измерения из geo_dim
```

```

g.city,
g.state,
-- Измерения из shipping_dim
s.ship_mode,
-- Даты из calendar_dim (с правильными псевдонимами)
cal_order.date as order_date,
cal_ship.date as ship_date,
-- Метрики из sales_fact
f.sales,
f.profit,
f.quantity,
f.discount
FROM {{ ref('sales_fact') }} AS f
LEFT JOIN {{ ref('customer_dim') }} AS c ON f.cust_id =
c.cust_id
LEFT JOIN {{ ref('product_dim') }} AS p ON f.prod_id =
p.prod_id
LEFT JOIN {{ ref('shipping_dim') }} AS s ON f.ship_id =
s.ship_id
LEFT JOIN {{ ref('geo_dim') }} AS g ON f.geo_id = g.geo_id
-- ИСПРАВЛЕНО: Добавляем псевдонимы, так как календарьиспользуется
дважды

LEFT JOIN {{ ref('calendar_dim') }} AS cal_order ON
f.order_date_id = cal_order.dateid
LEFT JOIN {{ ref('calendar_dim') }} AS cal_ship ON
f.ship_date_id = cal_ship.dateid

```

### **Код вашей индивидуальной mart-модели:**

```

-- models/marts/mart_monthly_running_profit.sql
{{ config(
    materialized='view',
    schema='dw'
)}}

```

```

WITH monthly_profit AS (
    SELECT
        date_trunc('month', order_date)::date AS sales_month,
        SUM(profit) AS monthly_profit
    FROM {{ ref('int_orders_pivoted') }}
    GROUP BY 1
),
running_profit AS (
    SELECT
        sales_month,
        monthly_profit,
        SUM(monthly_profit) OVER (ORDER BY sales_month) AS
cumulative_profit
    FROM monthly_profit
)
SELECT *
FROM running_profit

```

**Код кастомного теста test\_is\_positive.sql и пример его применения:**

```

-- tests/generic/test_is_positive.sql
{% test is_positive(model, column_name) %}
SELECT *
FROM {{ model }}
WHERE {{ column_name }} < 0
{% endtest %}

```

**Код snapshot\_product\_dim.sql:**

```

-- snapshots/snapshot_product_dim.sql
{% snapshot snapshot_product_dim %}
{{
    config(
        target_schema='dw_snapshots',
        strategy='check',

```

```

unique_key='prod_id',
check_cols=['segment', 'category'],
)
}}
SELECT prod_id, product_id, segment, category FROM {{
ref('product_dim') }}
{% endsnapshot %}

```

## Скриншот успешного выполнения dbt run, dbt test и dbt snapshot:

```

(dbt-env) C:\Users\User\Downloads\Магистратура\учеба\Платформы Data Engineering\pde_magistr\superstore_dwh>dbt run --select mart_monthly_running_profit
20:35:46 Running with dbt=1.10.13
20:35:46 Registered adapter: postgres=1.9.1
20:35:49 Found 14 models, 1 snapshot, 20 data tests, 1 source, 1 exposure, 449 macros
20:35:49 Concurrency: 1 threads (target='dev')
20:35:49 1 of 1 START sql view model dw.mart_monthly_running_profit ..... [RUN]
20:35:50 1 of 1 OK created sql view model dw.mart_monthly_running_profit ..... [CREATE VIEW in 0.28s]
20:35:50 Finished running 1 view model in 0 hours 0 minutes and 1.36 seconds (1.36s).
20:35:50 Completed successfully
20:35:50 Done. PASS=1 WARN=0 ERROR=0 SKIP=0 NO-OP=0 TOTAL=1

```

```

(dbt-env) C:\Users\User\Downloads\Магистратура\учеба\Платформы Data Engineering\pde_magistr\superstore_dwh>dbt test --select mart_monthly_running_profit
20:36:24 Running with dbt=1.10.13
20:36:25 Registered adapter: postgres=1.9.1
20:36:26 Found 14 models, 1 snapshot, 20 data tests, 1 source, 1 exposure, 449 macros
20:36:26 Concurrency: 1 threads (target='dev')
20:36:26 1 of 4 START test is_positive_mart_monthly_running_profit_cumulative_profit .... [RUN]
20:36:28 1 of 4 PASS is_positive_mart_monthly_running_profit_cumulative_profit ..... [PASS in 0.27s]
20:36:28 2 of 4 START test is_positive_mart_monthly_running_profit_monthly_profit ..... [RUN]
20:36:28 2 of 4 FAIL 3 is_positive_mart_monthly_running_profit_monthly_profit ..... [FAIL 3 in 0.20s]
20:36:28 3 of 4 START test not_null_mart_monthly_running_profit_cumulative_profit ..... [RUN]
20:36:28 3 of 4 PASS not_null_mart_monthly_running_profit_cumulative_profit ..... [PASS in 0.24s]
20:36:28 4 of 4 START test not_null_mart_monthly_running_profit_monthly_profit ..... [RUN]
20:36:28 4 of 4 PASS not_null_mart_monthly_running_profit_monthly_profit ..... [PASS in 0.20s]
20:36:28 Finished running 4 data tests in 0 hours 0 minutes and 1.97 seconds (1.97s).
20:36:29 Completed with 1 error, 0 partial successes, and 0 warnings:
20:36:29 Failure in test is_positive_mart_monthly_running_profit_monthly_profit (models\marts\schema.yml)
20:36:29 Got 3 results, configured to fail if != 0
20:36:29   compiled code at target\compiled\superstore_dwh\models\marts\schema.yml\is_positive_mart_monthly_running_profit_monthly_profit.sql
20:36:29 Done. PASS=3 WARN=0 ERROR=1 SKIP=0 NO-OP=0 TOTAL=4

```

```

(dbt-env) C:\Users\User\Downloads\Магистратура\учеба\Платформы Data Engineering\pde_magistr\superstore_dwh> dbt snapshot
18:59:12 Running with dbt=1.10.13
18:59:13 Registered adapter: postgres=1.9.1
18:59:14 Found 12 models, 1 snapshot, 15 data tests, 1 source, 1 exposure, 449 macros
18:59:14 Concurrency: 1 threads (target='dev')
18:59:14 1 of 1 START snapshot dw_snapshots.snapshot_product_dim ..... [RUN]
18:59:16 1 of 1 OK snapshotted dw_snapshots.snapshot_product_dim ..... [SELECT 4344 in 0.46s]
18:59:16 Finished running 1 snapshot in 0 hours 0 minutes and 1.54 seconds (1.54s).
18:59:16 Completed successfully
18:59:16 Done. PASS=1 WARN=0 ERROR=0 SKIP=0 NO-OP=0 TOTAL=1

```

## Скриншот с данными из моей индивидуальной mart-модели (вариант 5):

The screenshot shows a data management interface with a SQL editor at the top and a results table below. The SQL query is: `SELECT * FROM dw.mart_monthly_running_profit ORDER BY sales_month;`. The results table has four columns: `sales_month`, `monthly_profit`, and `cumulative_profit`. The first 10 rows of data are displayed, showing a progression from January 2016 to October 2016. A right-hand panel shows a selected value of '2016-01-01'.

	sales_month	monthly_profit	cumulative_profit
1	2016-01-01	6 272,6964	6 272,6964
2	2016-02-01	1 880,7989	8 153,4953
3	2016-03-01	3 071,8974	11 225,3927
4	2016-04-01	8 598,457	19 823,8497
5	2016-05-01	6 359,4265	26 183,2762
6	2016-06-01	11 026,6281	37 209,9043
7	2016-07-01	-281,8726	36 928,0317
8	2016-08-01	12 627,8536	49 555,8853
9	2016-09-01	22 448,2889	72 004,1742
10	2016-10-01	8 209,7283	80 213,9025

48 строк получено - 0.1s (0.0s получ.), 2025-10-27 в 23:44:00