ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
(ГАОУ ВО МГПУ)
Институт цифрового образования
Департамент информатики, управления и технологий

Практическая работа № 2.1

по дисциплине «Платформы Data Engineering»

Выполнил:
студент группы БД251м

Направление подготовки/Специальность
38.04.05 - Бизнес-информатика
St_62
(Ф.И.О.)

Проверил:
Кандидат технических наук, доцент
(ученая степень, звание)
Босенко Тимур Муртазович
(Ф.И.О.)

Москва 2025

Архитектура DWH

Ключевые фрагменты кода

```
-- models/staging/stg_orders.sql
-- Эта модель читает данные из исходной таблицы stg.orders,
-- приводит их к нужным типам и исправляет ошибку с почтовым кодом.
-- Все последующие модели будут ссылаться на эту, а не на исходную таблицу.

SELECT
-- Приводим все к нижнему регистру для консистентности в dbt
"order_id",
("order_date")::date as order_date,
("ship_date")::date as ship_date,
"ship_mode",
"customer_id",
"customer_name",
"segment",
"country",
"city",
"state",
-- Исправляем проблему с Burlington прямо здесь, один раз и навсегда
CASE
WHEN "city" = 'Burlington' AND "postal_code" IS NULL THEN '05401'
ELSE "postal_code"
END as postal_code,
"region",
"product_id",
"category",
"subcategory" as sub_category, -- переименовываем для соответствия
"product_name",
"sales",
"quantity",
"discount",
"profit"
FROM {{ source('stg', 'orders') }}


-- Создает таблицу фактов, объединяя все измерения
SELECT
-- Суррогатные ключи из измерений
cd.cust_id,
pd.prod_id,
sd.ship_id,
gd.geo_id,
-- Ключи для календаря
to_char(o.order_date, 'yyyymmdd')::int AS order_date_id,
to_char(o.ship_date, 'yyyymmdd')::int AS ship_date_id,
```

```sql
-- Бизнес-ключ и метрики
o.order_id,
o.sales,
o.profit,
o.quantity,
o.discount
FROM {{ ref('stg_orders') }} AS o
LEFT JOIN {{ ref('customer_dim') }} AS cd ON o.customer_id = cd.customer_id
LEFT JOIN {{ ref('product_dim') }} AS pd ON o.product_id = pd.product_id
LEFT JOIN {{ ref('shipping_dim') }} AS sd ON o.ship_mode = sd.ship_mode
LEFT JOIN {{ ref('geo_dim') }} AS gd ON o.postal_code = gd.postal_code AND o.city =
gd.city AND o.state = gd.state
```

```sql
-- Топ-5 штатов по суммарным продажам
select
g.state,
sum(f.sales) as total_sales,
count(distinct f.order_id) as number_of_orders
from {{ ref('sales_fact') }} as f
left join {{ ref('geo_dim') }} as g
on f.geo_id = g.geo_id
where g.state is not null
group by g.state
order by total_sales desc, state asc
limit 5
```

```yaml
# Путь к файлу: models/marts/schema.yml

version: 2

models:
  - name: shipping_dim
    columns:
      - name: ship_id
        tests:
          - unique
          - not_null

  - name: customer_dim
    columns:
      - name: cust_id
        tests:
          - unique
          - not_null

  - name: geo_dim
    columns:
      - name: geo_id
        tests:
          - unique
          - not_null

  - name: product_dim
    columns:
      - name: prod_id
        tests:
          - unique
          - not_null

  - name: sales_fact
    columns:
      - name: cust_id
        tests:
          - relationships:
              arguments:
                to: ref('customer_dim')
                field: cust_id

  - name: mart_geo_sales
    columns:
      - name: state
        tests:
          - unique
          - not_null
```

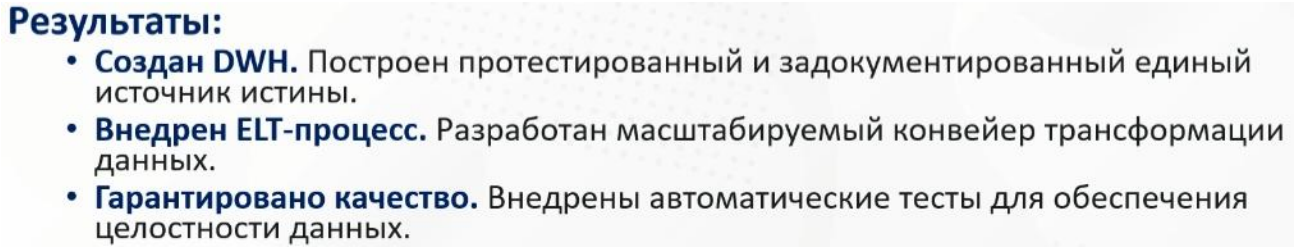
Результаты

```
(dbt-env) dev@dev-vm:~/Downloads/pde_magistr/superstore_dwh$ dbt run
13:31:03  Running with dbt=1.10.11
13:31:03  Registered adapter: postgres=1.9.1
13:31:03  Found 8 models, 11 data tests, 1 source, 435 macros
13:31:03
13:31:03  Concurrency: 4 threads (target='dev')
13:31:03
13:31:03  1 of 8 START sql table model dw_test.calendar_dim ..... [RUN]
13:31:03  2 of 8 START sql view model stg.stg_orders ..... [RUN]
13:31:04  2 of 8 OK created sql view model stg.stg_orders ..... [CREATE VIEW in 0.15s]
13:31:04  3 of 8 START sql table model dw_test.customer_dim ..... [RUN]
13:31:04  4 of 8 START sql table model dw_test.geo_dim ..... [RUN]
13:31:04  5 of 8 START sql table model dw_test.product_dim ..... [RUN]
13:31:04  1 of 8 OK created sql table model dw_test.calendar_dim ..... [SELECT 7670 in 0.18s]
13:31:04  6 of 8 START sql table model dw_test.shipping_dim ..... [RUN]
13:31:04  3 of 8 OK created sql table model dw_test.customer_dim ..... [SELECT 1093 in 0.31s]
13:31:04  4 of 8 OK created sql table model dw_test.geo_dim ..... [SELECT 932 in 0.32s]
13:31:04  5 of 8 OK created sql table model dw_test.product_dim ..... [SELECT 4644 in 0.33s]
13:31:04  6 of 8 OK created sql table model dw_test.shipping_dim ..... [SELECT 4 in 0.29s]
13:31:04  7 of 8 START sql table model dw_test.sales_fact ..... [RUN]
13:31:04  7 of 8 OK created sql table model dw_test.sales_fact ..... [SELECT 25967 in 0.20s]
13:31:04  8 of 8 START sql table model dw_test.mart_geo_sales ..... [RUN]
13:31:04  8 of 8 OK created sql table model dw_test.mart_geo_sales ..... [SELECT 5 in 0.09s]
13:31:04
13:31:04  Finished running 7 table models, 1 view model in 0 hours 0 minutes and 0.98 seconds (0.98s).
13:31:04
13:31:04  Completed successfully
13:31:04
13:31:04  Done. PASS=8 WARN=0 ERROR=0 SKIP=0 NO-OP=0 TOTAL=8
```

```
(dbt-env) dev@dev-vm:~/Downloads/pde_magistr/superstore_dwh$ dbt test
13:32:42  Running with dbt=1.10.11
13:32:43  Registered adapter: postgres=1.9.1
13:32:43  Found 8 models, 11 data tests, 1 source, 435 macros
13:32:43
13:32:43  Concurrency: 4 threads (target='dev')
13:32:43
13:32:43  1 of 11 START test not_null_customer_dim_cust_id ..... [RUN]
13:32:43  2 of 11 START test not_null_geo_dim_geo_id ..... [RUN]
13:32:43  3 of 11 START test not_null_mart_geo_sales_state ..... [RUN]
13:32:43  4 of 11 START test not_null_product_dim_prod_id ..... [RUN]
13:32:43  2 of 11 PASS not_null_geo_dim_geo_id ..... [PASS in 0.16s]
13:32:43  4 of 11 PASS not_null_product_dim_prod_id ..... [PASS in 0.09s]
13:32:43  5 of 11 START test not_null_shipping_dim_ship_id ..... [RUN]
13:32:43  3 of 11 PASS not_null_mart_geo_sales_state ..... [PASS in 0.15s]
13:32:43  1 of 11 PASS not_null_customer_dim_cust_id ..... [PASS in 0.19s]
13:32:43  7 of 11 START test unique_customer_dim_cust_id ..... [RUN]
13:32:43  6 of 11 START test relationships_sales_fact_cust_id__cust_id__ref_customer_dim_  [RUN]
13:32:43  8 of 11 START test unique_geo_dim_geo_id ..... [RUN]
13:32:44  5 of 11 PASS not_null_shipping_dim_ship_id ..... [PASS in 0.15s]
13:32:44  7 of 11 PASS unique_customer_dim_cust_id ..... [PASS in 0.13s]
13:32:44  9 of 11 START test unique_mart_geo_sales_state ..... [RUN]
13:32:44  10 of 11 START test unique_product_dim_prod_id ..... [RUN]
13:32:44  6 of 11 PASS relationships_sales_fact_cust_id__cust_id__ref_customer_dim_ ..... [PASS in 0.16s]
13:32:44  8 of 11 PASS unique_geo_dim_geo_id ..... [PASS in 0.14s]
13:32:44  11 of 11 START test unique_shipping_dim_ship_id ..... [RUN]
13:32:44  9 of 11 PASS unique_mart_geo_sales_state ..... [PASS in 0.08s]
13:32:44  10 of 11 PASS unique_product_dim_prod_id ..... [PASS in 0.05s]
13:32:44  11 of 11 PASS unique_shipping_dim_ship_id ..... [PASS in 0.06s]
13:32:44
13:32:44  Finished running 11 data tests in 0 hours 0 minutes and 0.63 seconds (0.63s).
13:32:44
13:32:44  Completed successfully
13:32:44
13:32:44  Done. PASS=11 WARN=0 ERROR=0 SKIP=0 NO-OP=0 TOTAL=11
```

Таблица: dw_test.mart_geo_sales

	state	total_sales	number_of_orders
0	California	1.206441e+06	1051
1	New York	7.456531e+05	562
2	Texas	4.415029e+05	507
3	Washington	3.857865e+05	285
4	Pennsylvania	3.229228e+05	318



Выводы

Преимущества dbt для реализации DWH по сравнению с написанием DDL/DML скриптов вручную:

- Встроенные тесты (not_null, unique) и кастомные тесты обеспечивают контроль целостности и ловят ошибки рано.
- Можно ссылаться на SQL-запросы и модели в следующей работе, вместо предварительной обработки сырых данных в каждом проекте.
- Автогенерация документации (dbt docs) и граф зависимостей повышают прозрачность архитектуры.