

Python + dbt + snowflake



Теория <https://ivan-shamaev.ru/dbt-clickhouse-tutorial-run-model-data/>

Курс <https://www.udemy.com/course/the-dbt-bootcamp-transform-your-data-using-data-build-tool/>

Подготовка Snowflake

1. Регистрация аккаунта ч\з ВПН
2. Создать WH (вкладка админ) TRANSFORM

Warehouses

Search: [] Type: All Size: All Status: All 2 Warehouses Columns: []

NAME	SI...	STATUS	RUNNING	QUEUED	OWNER	RESUMED
TRANSFORM	S	Started	2	0	ACCOUNTADMIN	5 minutes ago
SYSTEM\$STREAMLIT_NOTEBO...	XS	Suspended	0	0	—	1 hour ago
TRANSFORM_DEAD	S	Suspended	0	0	ACCOUNTADMIN	1 hour ago

3. Создать DB RAW и ANALYTICS

The screenshot shows the Snowflake web interface. On the left, the navigation menu includes 'Create', 'Home', 'Search', 'Projects', 'Data', 'Databases', 'Add Data', 'Data Products', 'AI & ML', and 'Monitoring'. The 'Databases' section is highlighted. In the center, a tree view shows the hierarchy: ANALYTICS > RAW > GLOBALMART > Tables > ORDERS. On the right, a table titled 'Databases' lists 4 databases:

NAME	SOURCE	OWNER	CREATED
ANALYTICS	Local	ACCOUNTADMIN	1 hour ago
RAW	Local	ACCOUNTADMIN	59 minut...
SNOWFLAKE	Share	—	1 hour ago
SNOWFLAKE_SAMPLE_DATA	Share	ACCOUNTADMIN	1 hour ago

4. В RAW создать схему GLOBALMART

5. В GLOBALMART создать таблицы orders

The screenshot shows the Snowflake web interface for the 'RAW / GLOBALMART / ORDERS' table. The table is owned by 'ACCOUNTADMIN' and was created 46 minutes ago. The 'Columns' tab is selected, showing 8 columns:

NAME	TYPE
CUSTOMERID	Varchar
ORDERCOSTPRICE	Float
ORDERDATE	Varchar
ORDERID	Varchar
ORDERSELLINGPRICE	Float
PRODUCTID	Varchar
SHIPDATE	Varchar
SHIPMODE	Varchar

и загрузить данные, выбрав в качестве WH TRANSFORM. Таким же образом создаем и загружаем таблицы customer и product

RAW / GLOBALMART / CUSTOMER

Table ACCOUNTADMIN just now 0 0.0B

Table Details **Columns** Data Preview Copy History

5 Columns

NAME ↑	TYPE
COUNTRY	Varchar
CUSTOMERID	Varchar
SEGMENT	Varchar
STATE	Varchar
SUDTOMERNAME	Varchar

RAW / GLOBALMART / PRODUCT

Table ACCOUNTADMIN just now 0 0.0B

Table Details **Columns** Data Preview Copy History

4 Columns

NAME ↑	TYPE
CATEGORY	Varchar
PRODUCTID	Varchar
PRODUCTNAME	Varchar
SUBCATEGORY	Varchar

Настройка DBT

https://quickstarts.snowflake.com/guide/data_engineering_with_snowpark_python_and_dbt/#0

```
> python3 -m venv dbt-env # Создаем env
> source dbt-env/bin/activate # активируем его
> python -m pip install dbt-core dbt-snowflake # установка dbt пакетов
> dbt --version
> dbt init SnowflakeDBTIntro # инициализация проекта dbt
> dbt run # пробный запуск
```

dbt run -s model_name - запускает выбранный модуль

dbt run -s +model_name - запускает выбранный модуль и зависимые от него модули

dbt docs generate - сгенерировать документацию

dbt test -s raw_customer - запуск тестов на модель из yaml файла в папке models

dbt test -s test_raw_orders_selling_price_is_positive - запуск теста из папки tests

dbt test -s source:globalmart - запускает тесты на источник, которые прописаны в yaml файле в папке models

dbt docs serve --port 8001 --host "192.168.0.3" - запустить сгенерированную доку в браузере

dbt seed — это команда для загрузки CSV-файлов (так называемых «семян») в хранилище данных в инструменте dbt.(т е в папку seed добавляем файл и запускаем команду)

В dbt существует 2 вида тестирования:

1. Тестирование схемы - проверяет качество централизованных данных(уникальность, ссылочная целостность, nulls и тд)
2. Тестирование значения данных

Types of Tests

1. singular - a specific test for a specific model
2. generic tests - scalable tests where you write a few lines of html code and then testing a model or a column
 - unique - every value in a column of a model is unique
 - not null - every value in a column of a model is not null
 - accepted values - every value in the column exists in a given lists
 - relationships - each value in a column exists in a column of another table

Использование jinja

```
-- объявление переменной
{% set tabletype = "orderstable" -%}
{% set category = "Furniture" -%}

select
    orderid,
    '{{ tabletype }}' as tablesourcesource,
    case when category = '{{category}}' then orderprofit
    end as {{category}}_orderprofit
from {{ ref('stg_orders') }}
```

```
-- цикл и условный оператор
{% set categorys = ["Furniture", "Office", "Technology"] -%}

select
    orderid,
    {% for category in categorys %}
    sum(case when category = '{{category}}' then orderprofit end) as
    {{category}}_orderprofit
    {% if not loop.last %}, {% endif %}
    {% endfor %}
from {{ ref('stg_orders') }}
group by 1
```

Установка пакетов

hub.getdbt.com

```
> dbt deps # обновление пакетов, сохраненных в файле .yaml
```