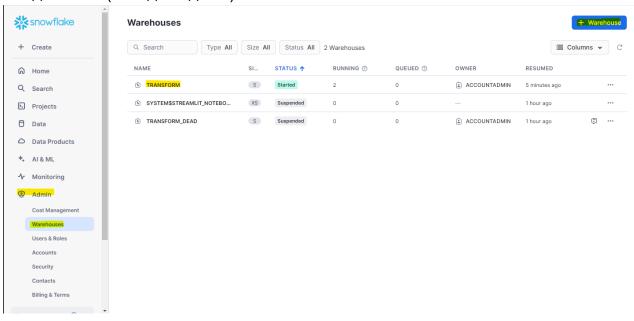
Python + dbt + snowflake



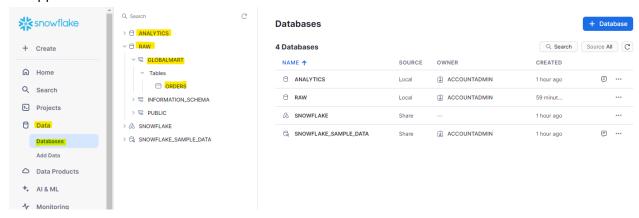
Теория https://ivan-shamaev.ru/dbt-clickhouse-tutorial-run-model-data/ Курс https://www.udemy.com/course/the-dbt-bootcamp-transform-your-data-using-data-build-tool/

Подготовка Snowflake

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



3. Создать DB RAW и ANALYTICS

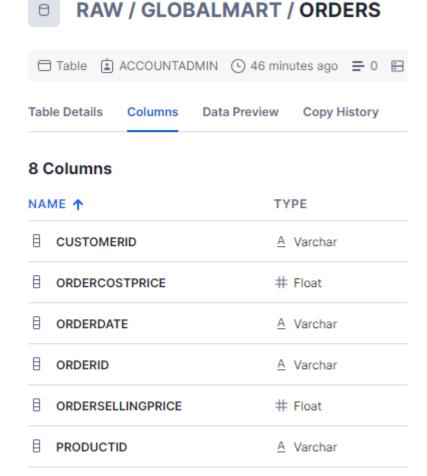


4. В RAW создать схему GLOBALMART

SHIPDATE

SHIPMODE

5. В GLOBALMART создать таблицы orders



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

A Varchar

A Varchar

RAW / GLOBALMART / CUSTOMER

| | Table 🛓 | ACCOUNTAI | DMIN () just no | w = 0 @ | ∃ 0.0B | |
|-----------------------|---------|--------------|-----------------|---------|-----------|--|
| Table Details Columns | | Data Preview | Copy His | tory | | |
| 5 (| Columns | | | | | |
| NA | МЕ ↑ | TYPE | | | | |
| B | COUNTRY | ′ | | | A Varchar | |
| B | сиѕтоме | RID | | | A Varchar | |
| B | SEGMENT | | | | A Varchar | |
| B | STATE | | | | A Varchar | |
| 8 | SUDTOME | RNAME | | | A Varchar | |

RAW / GLOBALMART / PRODUCT

| Table | accountai | DMIN (5) Just no | w = 0 🖶 0.0E | 5 | | | | | | |
|--------------|-----------|------------------|--------------|---------|--|--|--|--|--|--|
| Table Detail | s Columns | Data Preview | Copy History | | | | | | | |
| 4 Columns | | | | | | | | | | |
| NAME ↑ TY | | | | | | | | | | |
| B CATEG | ORY | | <u>A</u> | Varchar | | | | | | |
| ∃ PRODU | ICTID | | A | Varchar | | | | | | |
| ∃ PRODU | ICTNAME | | A | Varchar | | | | | | |
| B SUBCA | TEGORY | | <u>A</u> | 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 файлик 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 tablesource,
   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 .yml
```