Tài liệu hướng dẫn, vận hành và cấu hình Postgres, Airflow, Superset

I. Tài liệu hướng dẫn sử dụng và vận hành

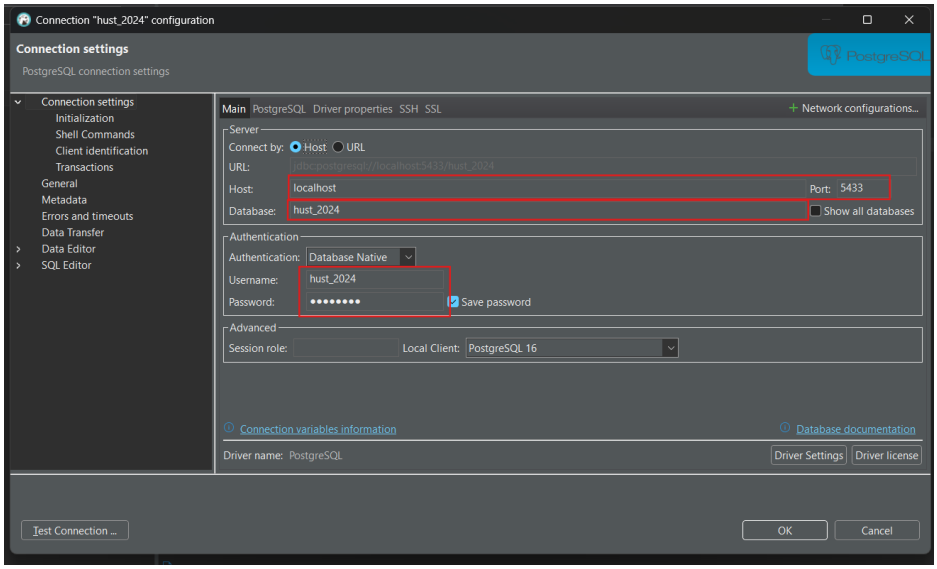
1. Postgres

a. Hướng dẫn kết nối đến Postgres từ Dbeaver

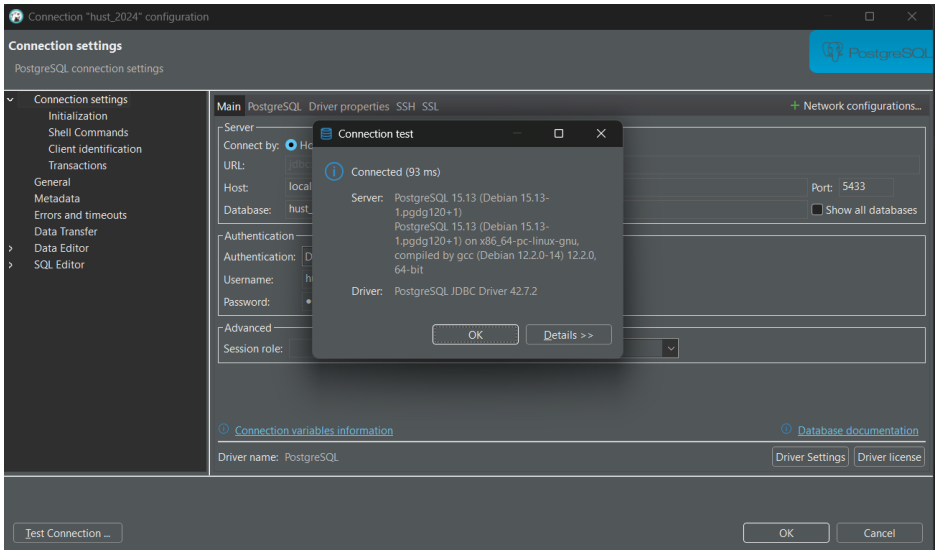
B1: Download Dbeaver: [Download | DBeaver Community]

B2: Tạo connect đến Postgres:

- Chọn biểu tượng New Database Connection ở thanh công cụ hoặc ấn tổ hợp CTRL + SHIFT + N
- Ở hộp công cụ Connect To Databasse chọn ALL và nhập postgres vào thanh tìm kiếm -> chọn biểu tượng Postgres
- Nhập các thông tin kết nối vào các phần tương ứng:
  Thông tin kết nối mặc định: Host: localhost:5433
  Database: hust_2024
  User/Pas: hust_2024/admin123



- Chọn test Connection -> Hiện ra thông báo connected là đúng



B3: Tạo các schema và các bảng tương ứng với với dữ liệu cần kéo về
Lưu ý:

- Có thể thêm các trường kỹ thuật như create_date để lưu thời gian kéo dữ liệu về để dễ xử lý sau này).
- Có thể chọn đánh partition các trường date hoặc timestamp để có tối ưu truy vấn và tránh phân mảnh dữ liệu nếu thực hiện delete nhiều lần
- Đánh index hợp lý để tối ưu truy vấn nếu dữ liệu lớn (> 1M record/bảng)

b. Hướng dẫn kết nối đến Postgres từ Airflow
B1: Import thư viện Psycopg2 vào các Dags làm việc với database
B2: Khởi tạo cấu hình kết nối:

```
conn = psycopg2.connect(
     host='host.docker.internal',
     port=5433,
     user='hust_2024',
     password='admin123',
     database='hust_2024'
)
cursor = conn.cursor()
```

B3: Thực hiện các câu lệnh SQL:

```
cursor.execute(""" SELECT *
              FROM financial_ratios """)
```

B4: Đóng kết nối:

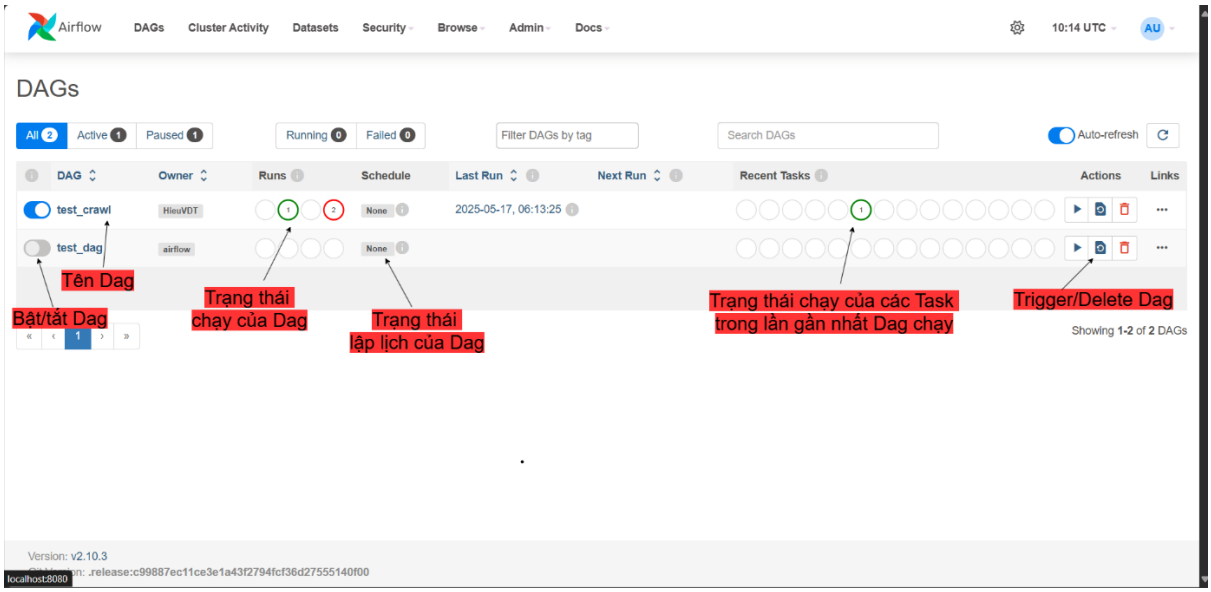
```
cursor.close()
conn.close()
```

c. Hướng dẫn kết nối đến Postgres từ Superset

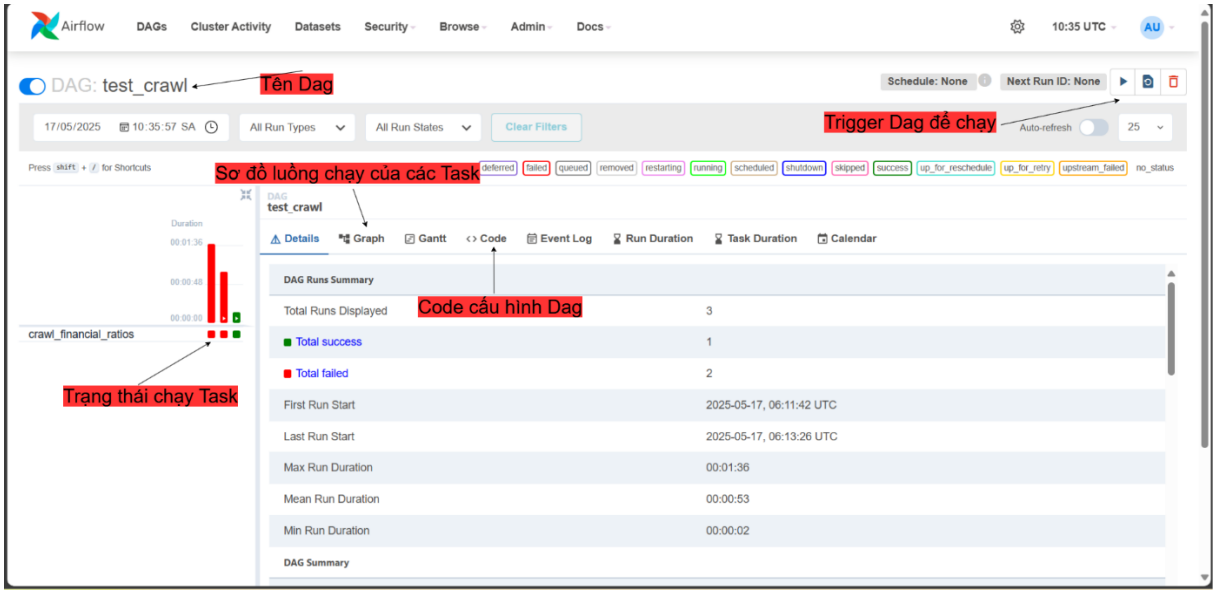
2. Airflow

- Cấu hình các Dags giống 2 Dags mẫu đã tạo trong foulder ./airflow/dags/
- Nếu muốn import thêm thư viện mới để viết Dags thì thêm vào file **requirements.txt**
- Sử dụng Airflow UI
    - Truy cập vào **http://localhost:8080/** sử dụng user/pass: admin/admin

- Thông tin các phần trong UI:



- Thông tin các phần trong Dags UI



- **Chú ý:**
  - Không sửa những phần đã comment không sửa
  - Chia nhỏ công việc thành các Task nhỏ để dễ kiểm soát
  - Sử dụng logging để dễ quản lý và debug
  - Đối với các Task crawl dữ liệu nên cấu hình thời gian sleep 3s đối với các luồng crawl trong cùng 1 task và sleep 30-40s giữa các task, đối với các task nặng như crawl dữ liệu đánh giá hoặc chi tiết sản phẩm nên chạy theo batch, sau mỗi batch sẽ sleep 30s để tránh bị block

3. **DBT**

B1: Cài đặt các thư viện trong file DBT/requirements.txt: **pip install -r requirements.txt**

B2: Điều chỉnh lại thông tin kết nối đến database ở các file **profiles.yml**

B3: Chạy câu lệnh: **dbt debug** để kiểm tra các thông tin đã đúng chưa

B3: Thêm các bảng dữ liệu cần sử dụng vào file **DBT/models/marts/source.yml theo mẫu có sẵn**

B4: Tạo các models DBT ở trong **DBT/models** theo mẫu models **DBT/models/marts/financial_ratios__fa.sql**

B5: Thêm thông tin các bảng trả về vào file **DBT/models/marts/schema.yml** theo mẫu có sẵn

B6: Khởi chạy tất cả các models bằng câu lệnh: **dbt run**

Hoặc chạy models tùy chọn bằng câu lệnh: **dbt run –select file_model.sql**

Câu lệnh	Ý nghĩa
dbt debug	Kiểm tra kết nối, cấu hình, môi trường đã sẵn sàng chưa
dbt run	Chạy tất cả các models đã tạo
dbt run --select file_models.sql	Chạy models tùy chọn
dbt test	Chạy các test đã định nghĩa
dbt test --select file_models.sql	Chạy test cho model tùy ch
dbt docs generate	Sinh tài liệu tự động cho các models đã tạo (metadata, mô tả, lineage)
dbt docs serve	Mở web tài liệu các models đã tạo

4. **Superset**

II. Tài liệu hướng dẫn cấu hình công cụ

Chú ý:

- Cài đặt Docker Desktop: Get Docker Desktop | Docker Docs
- Mở Docker Destop trước khi khởi tạo các công cụ
- Chỉ được chỉnh sửa các file được nhắc đến trong hướng dẫn, nếu chỉnh sửa các file khác mà dẫn đến lỗi mình sẽ **không chịu trách nhiệm**

1. **Postgres**

B1: Chỉnh sửa thông tin kết nối của postgres trong file **postgres/.env**

B1: Chuyển vị trí đến nơi lưu các file cấu hình postgres

B3: Chạy câu lệnh

- Docker-compose up -d –build
- Docker-compose up -d

B4: Mở Dbeaver kết nối thử đến postgres với hướng dẫn ở phần I.1.a

2. **Airflow**

B1: Chỉnh sửa thông tin kết nối db lưu log, web server trong file **airflow/.env**

B2: Thêm, sửa các thư viện cần dùng trong file **airflow/requirements.txt**

B3: Chạy câu lệnh

- Docker-compose up -d –build
- Docker-compose up -d

B4: Truy cập web DAGs - Airflow với hướng dẫn ở I.2

3. **DBT**

   Chỉ chỉnh sửa thông tin kết nối đến database trong file **profiles.yml** và các file định nghĩa **source, schema** trong models. Tuyệt đối **không chỉnh sửa** file **dbt_project.yml**

4. **Superset**

   B1: Chỉnh sửa thông tin kết nối Superset UI trong file superset/.env

   B2: Chạy câu lệnh

   - Docker-compose up -d –build
   - Docker-compose up -d

   B3: Truy cập web Superset với user/pass được cấu hình (mặc định là admin/admin)