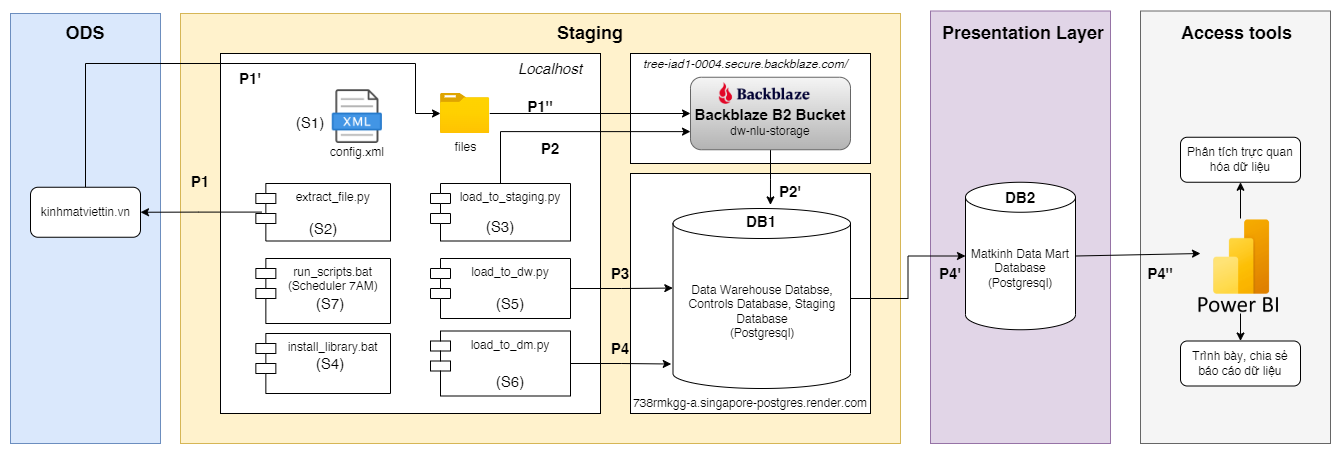
## Current DW Architecture Overview



Description:

(P1): Quá trình Extract dữ liệu:

- P1: Tiến hành cào dữ liệu từ nguồn trang web tương ứng file config.

- P1’: File .csv sau khi cào được lưu vào folder trên máy local theo file config chuẩn bị đưa lên bucket lưu trữ file tập trung trên B2.

- P1’’: Đưa file .csv lên bucket dw-nlu-storage từ file config.xml (S1) theo file config và ghi log trên Database dw (DB1).

(P2): Quá trình load dữ liệu vào staging trong Database dw (DB1)

- P2: Load file config.xml (S1) kết nối lấy url file .csv tương ứng file config.

- P2’: Import file vào bảng staging tương ứng file config.

(P3): Quá trình load dữ liệu vào Data Warehouse dw (DB1)

- P3: Insert into từ dữ liệu, đồng thời transform dữ liệu từ bảng staging vào dw..

(P4): Quá trình load dữ liệu lên Presentation Layer

- P4’: Dữ liệu Dim, Fact được đến Database Data Mart (DB2)

- P4’’: Data trên Data Mart (DB2) được đưa đến PowerBi để phân tích, báo cáo.

(S1): File config.xml chứa thông tin để kết nối các Database Server và Cloud Storage Server B2.

(S2): Script thực hiện process extract dữ liệu P1.

(S3): Script thực hiện process load dữ liệu vào staging P2.

(S4): Script giúp kiểm tra và cài đặt các thư viện cần thiết, chạy 1 lần đầu duy nhất khi bắt đầu với dự án.

(S5): Script thực hiện process load dữ liệu vào dw P3.

(S6): Script thực hiện process load dữ liệu vào dm P4.

(S7): Script thực hiện lần lượt các process, thứ tự lần lượt: P1 (Extract file) 🡪 P2 (Load to staging) 🡪 P3 (Load to dw) 🡪 P4 (Load to dm). Script được lập lịch chạy tự động với task scheduler hằng ngày vào 7AM.

(DB1): Database postgresql dw chứa các bảng file\_logs, file\_config, staging, dw.

(DB2): Database postgresql dm chứa các bảng dim, fact, report.

## Functional Specification

### 2.1. Impact Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Criteria** | **Implementer** | **Note** |
| 1 | Extract file |  | Quá trình cào dữ liệu từ trang web nguồn về file .csv, sau đó upload file lên backblaze b2. Ghi nhận log |
| 2 | Load to staging area |  | Quá trình truy cập file .csv tương ứng từ B2, insert dữ liệu vào bảng staging, đồng thời transform dữ liệu trong database dw |
| 3 | Load to Data Warehouse core |  | Quá trình chuyển dữ liệu từ staging sang dw |

### 2.2. Assumptions:

**Extract file module:**

* Người thực hiện:
* Thực hiện khi chạy script (S2) extract\_file.py, lập lịch task scheduler trên localhost chạy 6AM hằng ngày và thực hiện ghi log.
* Datasources:

+ Nguồn trang web lấy dữ liệu theo file config trong bảng file\_config trong Database dw (DB1): *dpg-ct2jaspu0jms738rmkgg-a.singapore-postgres.render.com.*

+ File .csv được lưu tạm ở localhost sau đó được lưu trữ ở B2 Bucket dw-nlu-storage: *tree-iad1-0004.secure.backblaze.com/*

+ File dữ liệu được lưu dưới dạng .csv.

+ Tên file theo cấu trúc: **data\_{table\_staging\_destiantion}\_YYYY-MM-DD\_{source}.csv**

**Load to staging module:**

* Người thực hiện:
* Thực hiện khi chạy script (S3) load\_to\_staging.py, lập lịch task scheduler trên localhost chạy 8AM hằng ngày và thực hiện ghi log.
* Datasources:

+ File .csv được lưu trữ ở B2 Bucket dw-nlu-storage: *tree-iad1-0004.secure.backblaze.com/file/dw-nlu-storage/*

+ File dữ liệu được lưu dưới dạng .csv.

+ Bảng Staging tương ứng file config được lưu trong Database dw (DB1): *dpg-ct2jaspu0jms738rmkgg-a.singapore-postgres.render.com.*

**Load to dw module:**

* Người thực hiện:
* Thực hiện khi chạy script (S3) load\_to\_dw.py, lập lịch task scheduler trên localhost chạy 10AM hằng ngày và thực hiện ghi log.
* Datasources:

+ Bảng staging tương ứng id config được lưu trong Database staging (DB1): *dpg-ct2jaspu0jms738rmkgg-a.singapore-postgres.render.com.*

+ Bảng dw được lưu trong Database dw (DB1): *dpg-ct2jaspu0jms738rmkgg-a.singapore-postgres.render.com.*

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **File Script name** | **Features** | **Note** |
| S2 | extract\_file.py | Thực hiện process extract dữ liệu P1. |  |
|  |  |  |  |

## Exception Handling

* **Normal Condition Checking**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Criteria** | **Value** | **Action** |
| 1 | Process status | **RUNNING** | Đang trong quá trình chạy tiến trình |
| 2 | File status | **ES** (Extract Successfully) | Có thể sẵn sàng load to staging |
| **LS** (Load to staging Successfully) | Có thể sẵn sàng load to Data Warehouse dw |
| **LWS** (Load to dw successfully) | Có thể sẵn sàng load to Datamart |
| **EF** (Extract Fail) | Lỗi trong quá trình cào dữ liệu, extract file lỗi. |

* **Exception/Error**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Type** | **Exception** | **Handling** |
| 1 | Process error  (Sysstem interrupt) | **[Extract]**  Lỗi trong quá trình cào dữ liệu  *(Mất kết nối, website chặn)* | Gửi mail thông báo. Kiểm tra kết nối. Chạy lại script extract\_file.py |
| **[Load to staging]**  Không có file .csv trên B2  *(Lỗi extract file hoặc lỗi kết nối B2)* | Gửi mail thông báo.  Thử chạy lại extract\_file.py |
| **[Có tiền trình đã/đang chạy]** | Gửi mail thông báo |
| 2 | Database connection error | **[Lỗi kết nối Database]**  *(Mất kết nối Internet)* | Gửi mail thông báo |

## How to run job manually

### 4.1. Tạo biến môi trường chứa mail nhận thông báo của chương trình

* Giả sử email muốn nhận thông báo của bạn là [your\_mail@gmail.com](mailto:your_mail@gmail.com)
* Mở cmd chạy lệnh:

setx MY\_EMAIL\_DW\_VAR “[your\_mail@gmail.com](mailto:your_mail@gmail.com)”

### 4.2. Chạy 1 file duy thực hiện các tất cả giai đoạn 1 lúc

Thứ tự lần lượt: P1 (Extract file) 🡪 P2 (Load to staging) 🡪 P3 (Load to dw) 🡪 P4 (Load to dm). Script được lập lịch chạy tự động với task scheduler hằng ngày vào 7AM.

* Xét điều kiện máy đã có python và kết nối internet
* Mở cmd di chuyển đến thư mục chứa các file script
* Chạy file install\_library.bat nếu lần đầu chạy dự án.

|  |
| --- |
| #1. Kết nối đến database Controls (Postgres)  # Thông tin kết nối trong file config.xml  #2. Sử dụng câu query để kiểm tra file config vừa chọn file config  SELECT \*  FROM file\_config  #3. Chạy file run\_scripts.bat với 3 tham số <id\_config> <địa chỉ cụ thể file config.xml> <Ngày> (Optional, mặc định là ngày hôm nay, định dạng YYYY-MM-DD)  run\_scripts.bat 1 E:\Documents\NLU\DataWarehouse\project\config.xml 2024-12-12 |

### 4.3. Chạy riêng lẻ từng giai đoạn: Extract file

* Xét điều kiện máy đã có python và kết nối internet
* Mở cmd di chuyển đến thư mục chứa các file script
* Chạy file install\_library.bat nếu lần đầu chạy dự án.

|  |
| --- |
| #1. Kết nối đến database Controls (Postgres)  # Thông tin kết nối trong file config.xml  #2. Sử dụng câu query để kiểm tra file config vừa chọn file config  SELECT \*  FROM file\_config  #3. Chạy file extract\_file.py với 3 tham số <id\_config> <địa chỉ cụ thể file config.xml> <Ngày> (Optional, mặc định là ngày hôm nay, định dạng YYYY-MM-DD)  python extract\_file.py 1 E:\Documents\NLU\DataWarehouse\project\config.xml 2024-12-12 |

### 4.4. Chạy riêng lẻ từng giai đoạn: Load to staging

* Xét điều kiện máy đã có python và kết nối internet
* Mở cmd di chuyển đến thư mục chứa các file script
* Chạy file install\_library.bat nếu lần đầu chạy dự án.

|  |
| --- |
| #1. Kết nối đến database Controls (Postgres)  # Thông tin kết nối trong file config.xml  #2. Sử dụng câu query để kiểm tra file config vừa chọn file config  SELECT \*  FROM file\_config  #3. Chạy file load\_to\_staging.py với 3 tham số <id\_config> <địa chỉ cụ thể file config.xml> <Ngày> (Optional, mặc định là ngày hôm nay, định dạng YYYY-MM-DD)  python load\_to\_staging.py 1 E:\Documents\NLU\DataWarehouse\project\config.xml |

### 4.5. Chạy riêng lẻ từng giai đoạn: Load to dw

* Xét điều kiện máy đã có python và kết nối internet
* Mở cmd di chuyển đến thư mục chứa các file script
* Chạy file install\_library.bat nếu lần đầu chạy dự án.

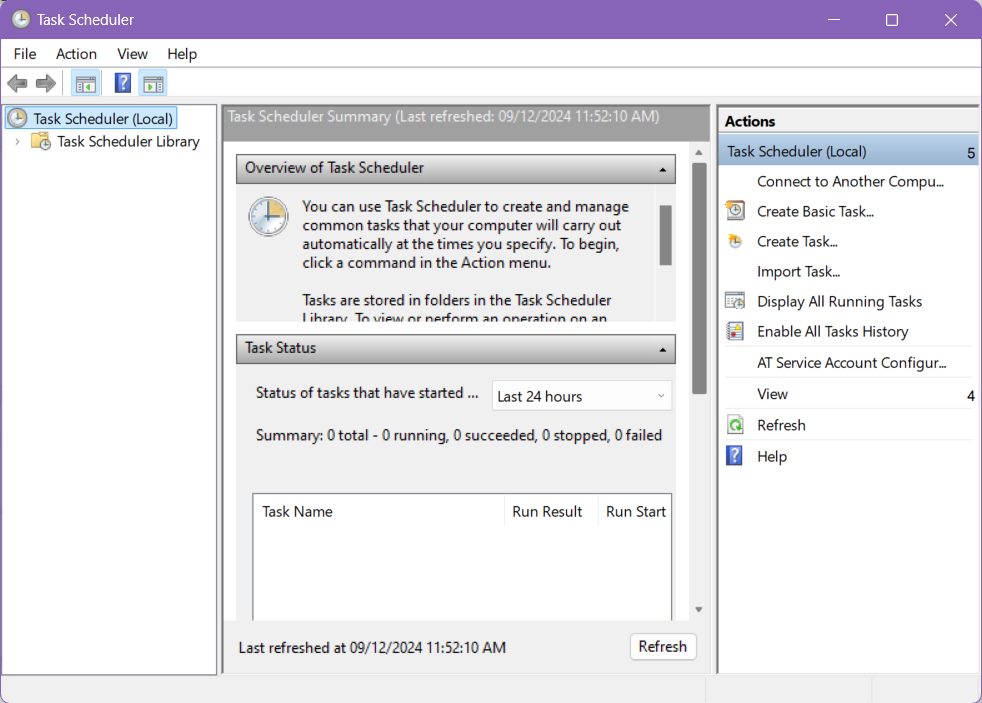
|  |
| --- |
| #1. Kết nối đến database Controls (Postgres)  # Thông tin kết nối trong file config.xml  #2. Sử dụng câu query để kiểm tra file config vừa chọn file config  SELECT \*  FROM file\_config  #3. Chạy file load\_to\_dw.py với 3 tham số <id\_config> <địa chỉ cụ thể file config.xml> <Ngày> (Optional, mặc định là ngày hôm nay, định dạng YYYY-MM-DD)  python load\_to\_dw.py 1 E:\Documents\NLU\DataWarehouse\project\config.xml |

## Scheduling

Sử dụng Task Scheduler trên Window

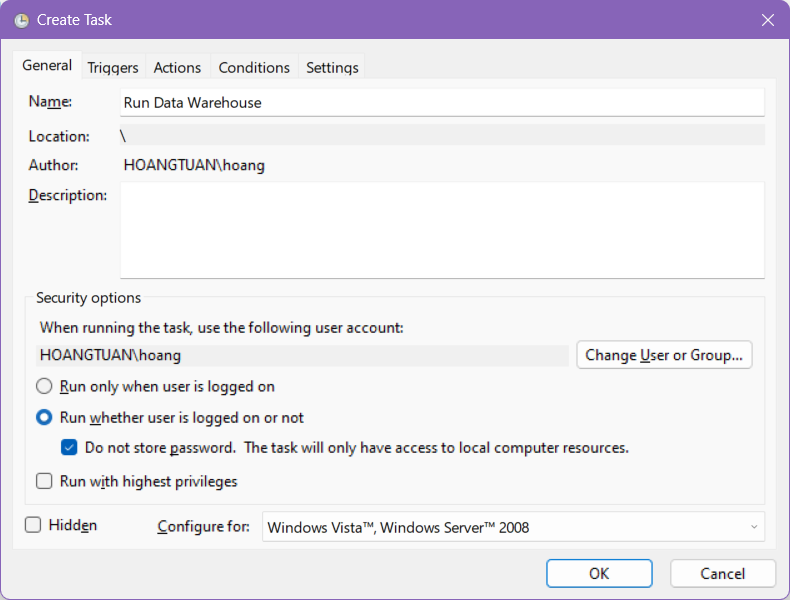
* Mở Task Scheduler: Nhấn **Win + S**, tìm kiếm Task Scheduler, và mở ứng dụng.
* Tạo task mới:

**1. Trong Task Scheduler, nhấn Create Task.**

****

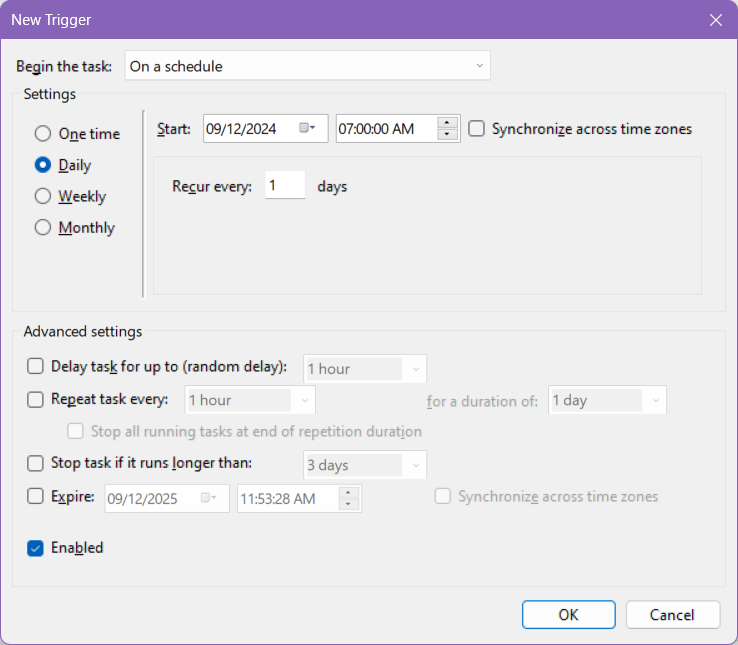
**2. Tab General:**

* Nhập tên task, ví dụ: Run Data Warehouse.
* Chọn **Run whether user is logged on or not**.
* Tick **Do not store password** (tùy chọn nếu không cần truy cập mạng).



**3.** **Tab Triggers**:

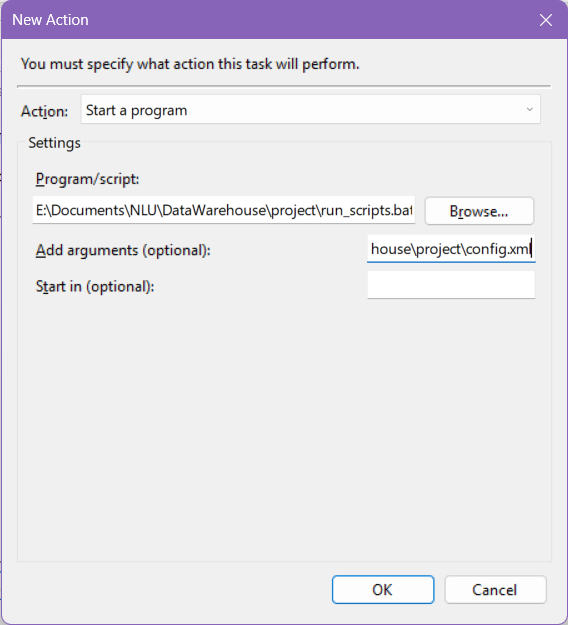
* Nhấn **New** để thêm trigger.
* Chọn loại lịch (ví dụ: hàng ngày, hàng giờ, hoặc giờ cụ thể).



**4**. **Tab Actions**:

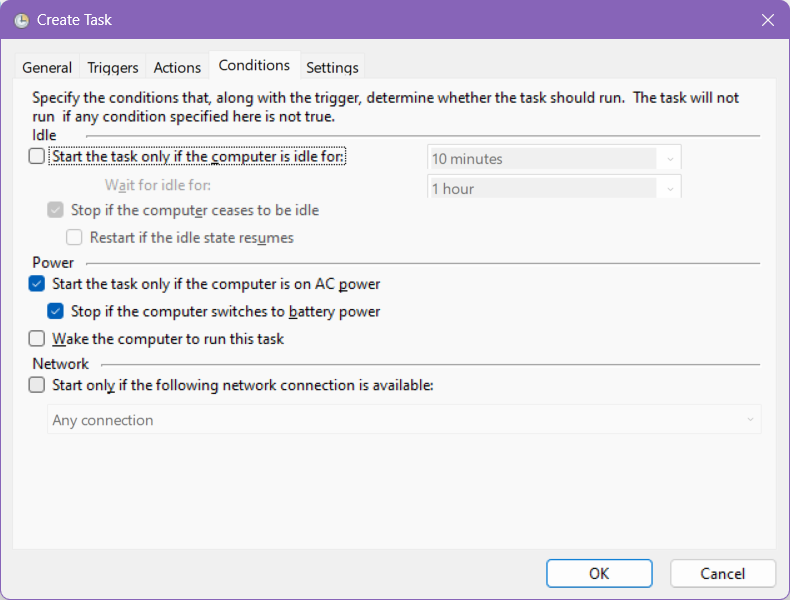
* Nhấn **New**.
* **Action:** Chọn **Start a Program**.
* **Program/script:** Chọn đường dẫn tới python.exe, ví dụ: E:\Documents\NLU\DataWarehouse\project\extract\_file.py
* **Add arguments (optional):** Nhập đường dẫn file script và các tham số id\_config, đường dẫn chính xác file config.xml, ví dụ:

1 E:\Documents\NLU\DataWarehouse\project\config.xml



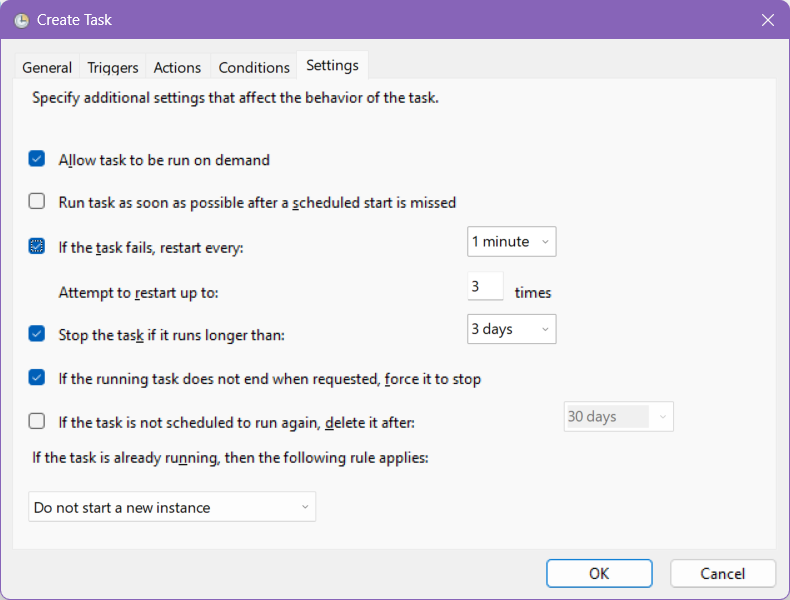
**5.Tab Conditions:**

* Bỏ tick **Start the task only if the computer is on AC power** nếu không muốn điều kiện này.



**6.Tab Settings**:

* Tick **Allow task to be run on demand**.
* Tick **If the task fails, restart every...** để tự động khởi động lại nếu xảy ra lỗi.

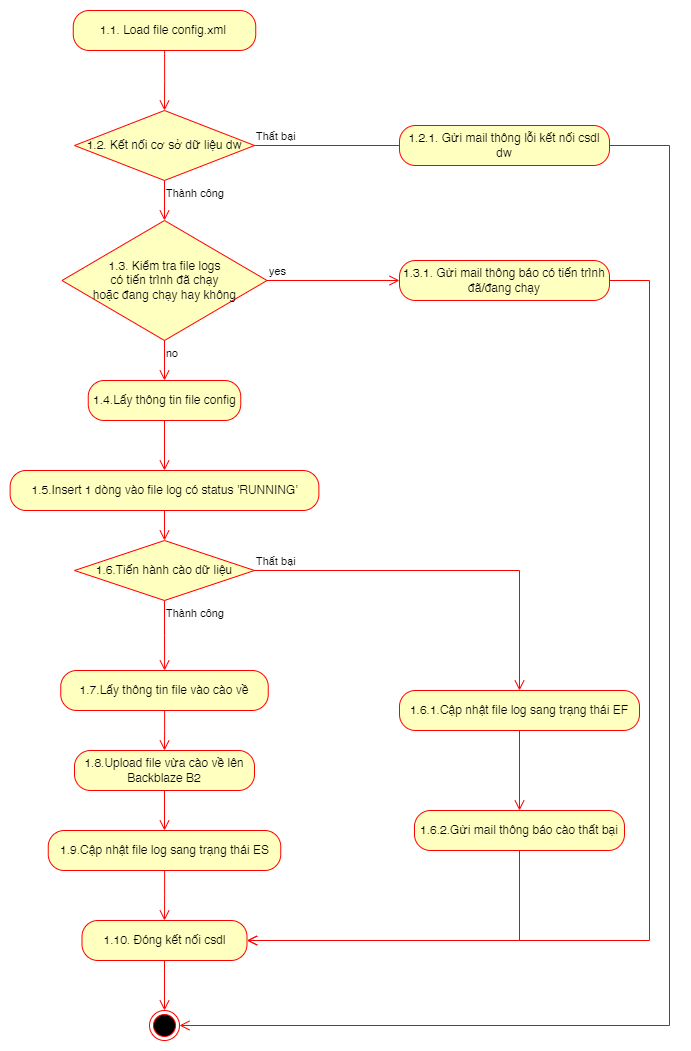


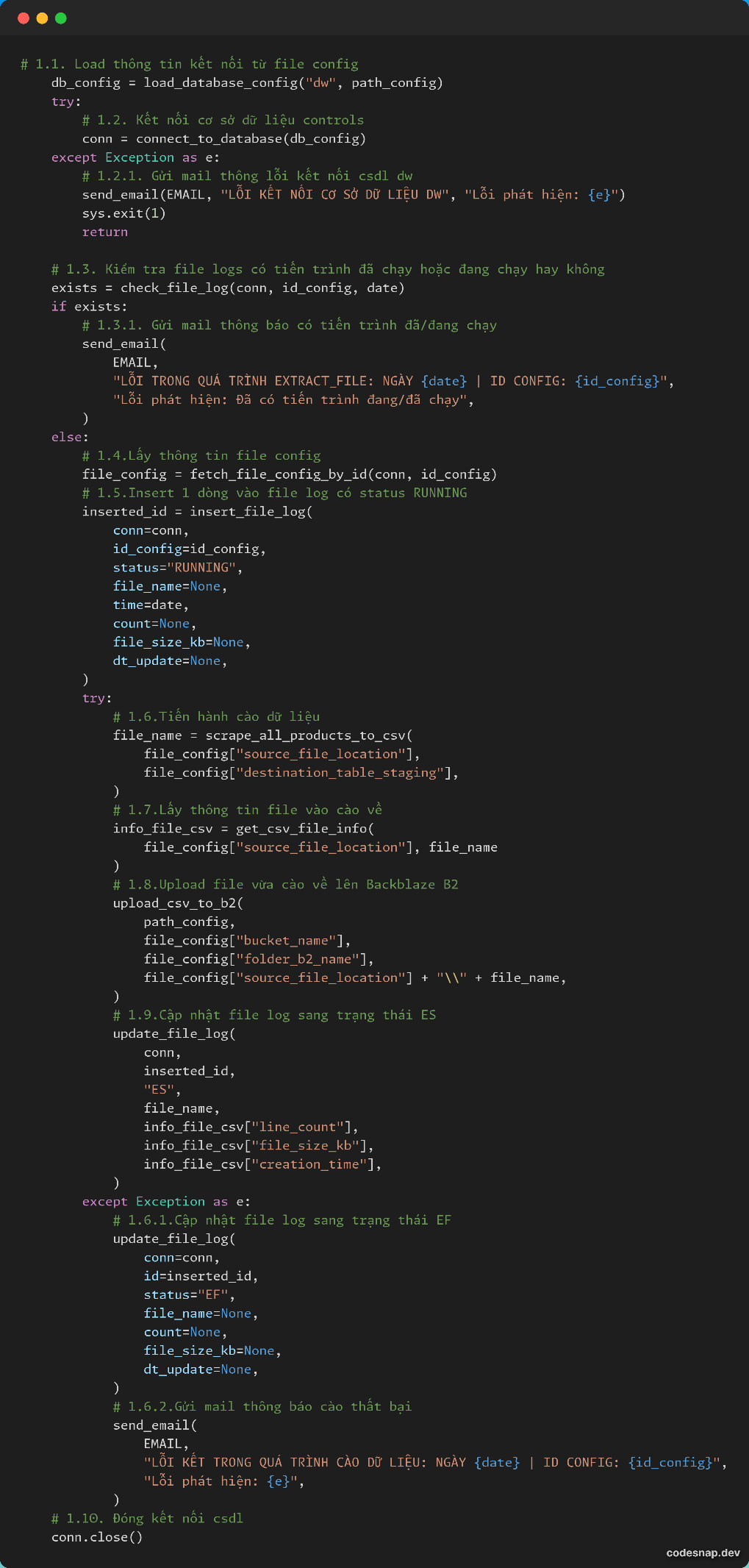
|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Subject** | **Schedule** | **Note** |
| 1 | Chạy với run\_scripts.bat | 7AM | Khi chạy sẽ tự động chạy các process theo thứ tự P1 (Extract file) 🡪 P2 (Load to staging) 🡪 P3 (Load to dw) 🡪 P4 (Load to dm). |

*\*Có thể tùy chỉnh chạy từng giai đoạn riêng lẻ*

## Logical Workflow

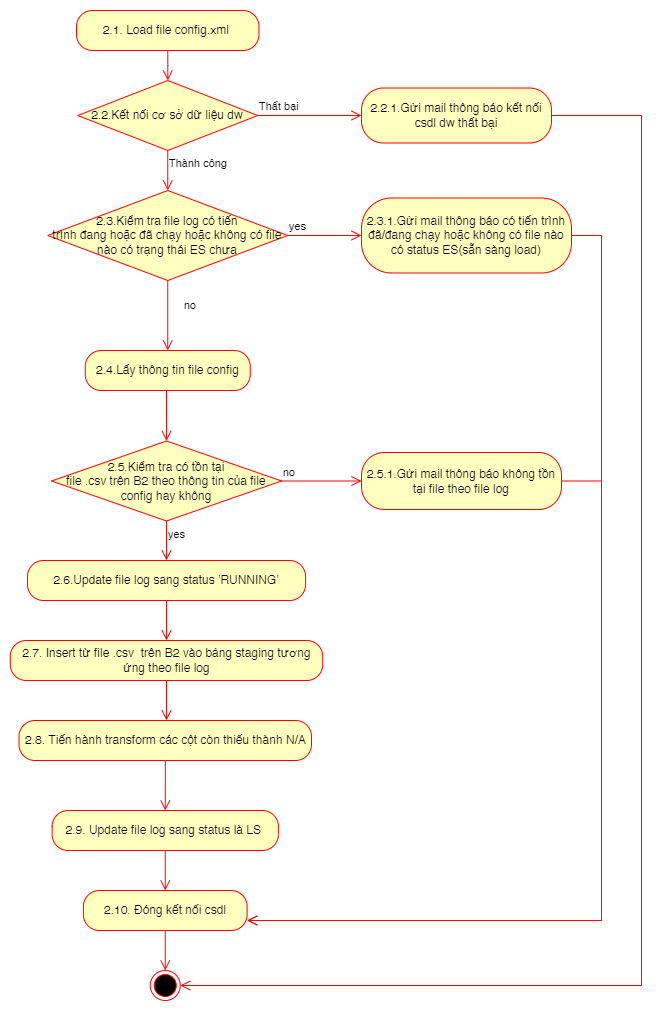
### 6.1. Extract file process





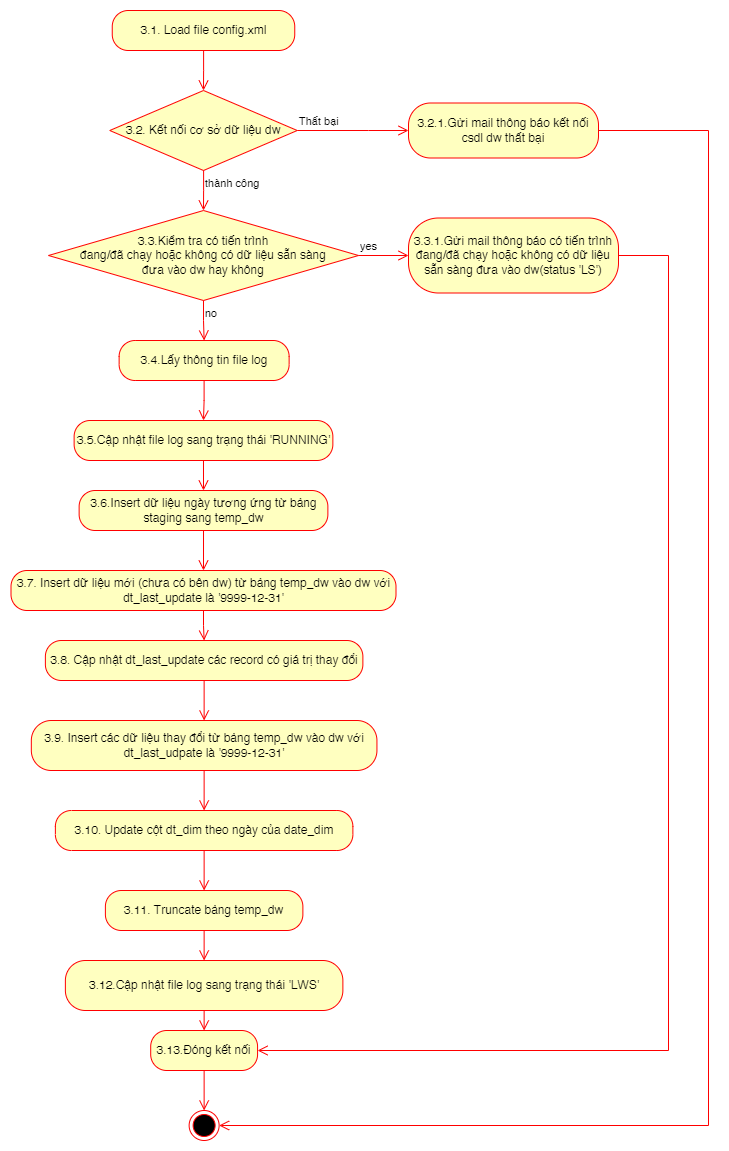
|  |
| --- |
| * 1. **load\_database\_config** |
| **Refer:**  **Purpose:** Lấy thông tin để kết nối database từ file config.xml đường dẫn chính xác file là input nhập vào.  **Input:** db\_name (tên database), config\_path(đường dẫn chính xác của file config.xml).  **Output:** 1 dictionary chứa thông tin kết nối database tương ứng. |
| * 1. **connect\_to\_database** |
| **Refer:**  **Purpose:** Kết nối với database Postgres.  **Input:** db\_config(1 dictionary chứa các thông tin kết nối).  **Output:** 1 biến Connection. |
| * 1. **check\_file\_log** |
| **Refer:**  **Purpose:** Kiểm tra file logs có tiến trình đã chạy hoặc đang chạy hay không.  **Input:** conn (1 connection kết nối csdl), id\_config (id của file config), date(ngày định dạng YYYY-MM-DD).  **Output:** True/False. |
| * 1. **fetch\_file\_config\_by\_id** |
| **Refer:**  **Purpose:** Lấy thông tin file config.  **Input:** conn (1 connection kết nối csdl), id\_config (id của file config).  **Output:** 1 dictionary chứa thông tin của file config. |
|  |
|  |
|  |
|  |

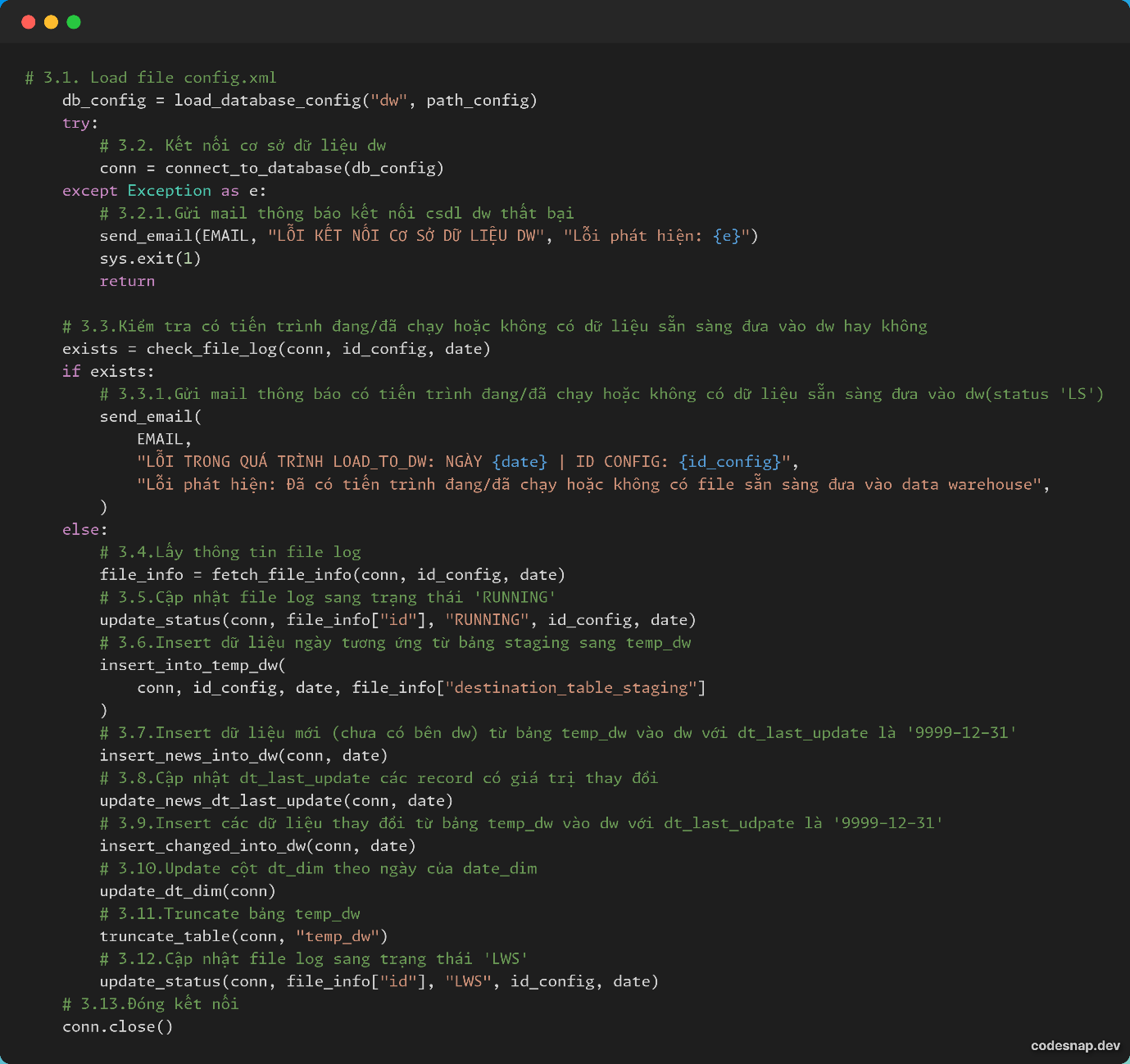
### 6.2. Load to staging process



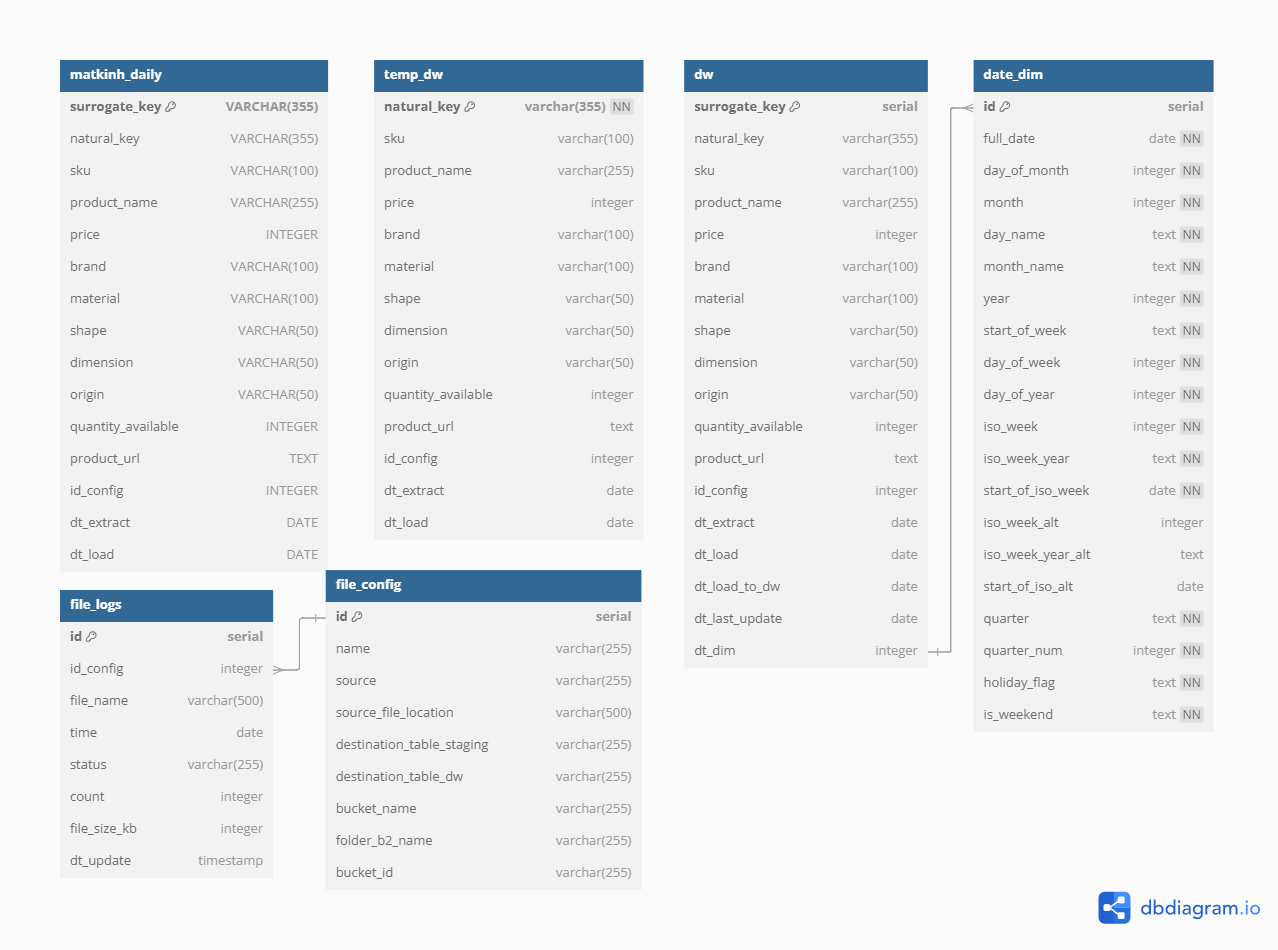


### 6.3. Load to dw process





## Database Structure



*(Database digram trong dw)*