**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙢🙠🕮🙢🙠**



**ĐỒ ÁN MÔN DATA WAREHOUSE.**

**Đề tài: Báo cáo Laptop.**

**Giảng viên hướng dẫn:** Nguyễn Đức Công Song

**Sinh viên thực hiện:**

Huỳnh Ngọc Quang Minh - 21130445

Huỳnh Thị Mai Phương - 21130149

Trần Quang Vũ - 21130615

Nguyễn Nhật Huy - 21130380

**Nhóm**: 4

**Chuyên ngành:** Công nghệ thông tin

**Tp. Hồ Chí Minh, tháng 11 năm 2024**

**MỤC LỤC**

[1 OVERVIEW 1](#_Toc184755314)

[1.1 Current DW Architecture Overview 1](#_Toc184755315)

[1.2 Propose DW Architecture Overview 1](#_Toc184755316)

[2 REFERENCE 2](#_Toc184755317)

[3 FUNCTIONAL SPECIFICATION 3](#_Toc184755318)

[3.1 Impact Analysis 3](#_Toc184755319)

[3.2 Assumptions 3](#_Toc184755320)

[3.3 Exception Handling 4](#_Toc184755321)

[3.3.1 Normal Condition/ Exception catching 4](#_Toc184755322)

[3.4 How to Run Job manually 8](#_Toc184755323)

[3.4.1 Require 8](#_Toc184755324)

[3.4.2 Tạo các database 8](#_Toc184755325)

[- 3.4.3 Load to Datawarehouse 27](#_Toc184755326)

[3.5 Work flow 28](#_Toc184755327)

[3.5.1 Load to Datawarehouse 28](#_Toc184755328)

[4 Database Architecture 36](#_Toc184755329)

[4.1 Database Controller 36](#_Toc184755330)

[4.2 Staging 40](#_Toc184755331)

[4.3 Data warehouse 45](#_Toc184755332)

[4.4 Data Mart 48](#_Toc184755333)

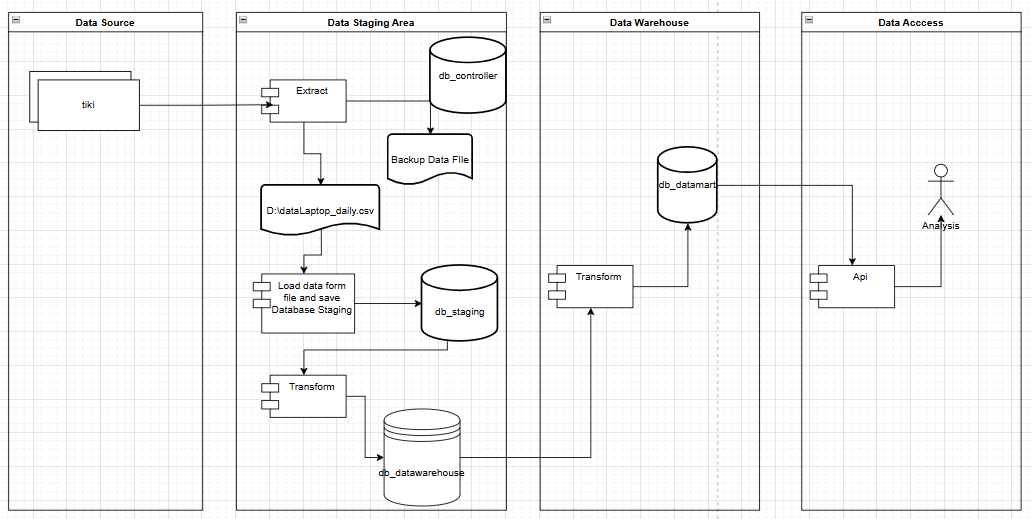
[5 Setup Processes 54](#_Toc184755334)

[5.1 Chuẩn bị. 54](#_Toc184755335)

[5.2 Setup process 54](#_Toc184755336)

# OVERVIEW

## Current DW Architecture Overview



Link: [System Architecture](https://drive.google.com/file/d/10roI7XmV6CnCVxjEaGYMUifz05hFA71R/view?usp=drive_link)

## Propose DW Architecture Overview

* Các process phải thực hiện nhiệm vụ xác định:
* Lấy dữ liệu từ API về máy tính và lưu vào file csv.
* Load data từ file csv vào database staging.
* Tiến hành transform data.
* Thực hiện load dữ liệu từ database staging vào database datawarehouse.
* Phân tích và thống kê dữ liệu sau đó load lên data marts.
* Các process hoạt động độc lập với nhau.
* Dữ liệu được lấy vào lúc 7h sáng mỗi ngày
* Áp dụng cơ chế type 2: add row

# REFERENCE

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Tên tài liệu** | **Mô tả** | **Link** |
| 1 | Tổng hợp | Tài liệu tổng hợp chứa thông tin chi tiết về toàn bộ dự án. | [link](https://drive.google.com/drive/folders/1yBsZHg61Ut-BbRiXBBalJz1oe21XK3qu?usp=sharing) |
| 2 | Data Mapping | Mô tả cách mapping dữ liệu | [link](https://docs.google.com/spreadsheets/d/1JZGHxOiI8izGNTOvHk4isc-vfP5ABxjLwwODv_Gj4Xs/edit?usp=sharing) |
| 3 | Data feed specification | Tài liệu về cấu trúc cơ sở dữ liệu, bao gồm thiết kế bảng, quan hệ, và chi tiết về các trường dữ liệu chính. | [link](https://docs.google.com/document/d/1UQP5ptgnC0K2MKkVcArjenAaQGBruO8fozJmLC-Jwyw/edit?tab=t.0) |

# FUNCTIONAL SPECIFICATION

## Impact Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Criteria** | **Evaluation** | **Tên** |
| 1 | Extract modules | Lấy dữ liệu từ API về máy tính và lưu vào file csv. | Trần Quang Vũ |
| 2 | Load Data Into Staging | Load dữ liệu từ file dữ liệu vào database staging | Huỳnh Thị Mai Phương |
| 3 | Load Data to Datawarehouse | Load dữ liệu từ database staging vào database datawarehouse | Huỳnh Ngọc Quang Minh |
| 4 | Load Data to Data Mart | Load dữ liệu từ database warehouse vào database mart | Nguyễn Nhật Huy |

## Assumptions

Trong quá trình phát triển và triển khai chức năng này, một số giả định quan trọng đã được đưa ra để đảm bảo tính nhất quán và hiệu quả.

*Extract modules:*

* Kết nối thành công đến database
* Load được các giá trị trong bảng config
* Nơi lưu file và nơi backup tồn tại

*Load Data to Datawarehouse:*

* Có data trong database staging
* Có database controller

*Cấu hình hệ thống:*

## Exception Handling

### Normal Condition/ Exception catching

|  |
| --- |
| **[Crawl data]**   * Ghi logs vào db\_controller: nội dung lỗi, thời gian, process nào * Gửi thông báo Email: nội dung lỗi, thời gian process nào   **[Load Data to Staging failed]**   * Ghi logs vào db\_controller: nội dung lỗi, thời gian, process nào * Gửi thông báo Email: nội dung lỗi, thời gian process nào   **[Load Data to Datawarehouse failed]**   * Ghi log vào database controller: nội dung lỗi, thời gian, process nào * Gửi thông báo Email: nội dung lỗi, thời gian chạy chương trình, thời gian kết thúc, process nào   **[Load Data to DataMart failed]**   * Ghi log vào db\_controller: nội dung lỗi, thời gian, process nào * Gửi thông báo Email: nội dung lỗi, thời gian chạy chương trình, thời gian kết thúc, process nào |

* **Normal Condition Checking**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Criteria** | **Value** | **Action** |
| 2 | LoadDataStaging | READY | - Bắt đầu quá trình load dữ liệu.  - Cập nhật trạng thái thành RUNNING.  - Thực thi các bước load dữ liệu từ file vào  staging. |
| RUNNING | * Không chạy process |
| FAILED | * Ghi Log vào db\_controller * Gửi Email “Failed” và thông tin lỗi |
| SUCCESS | * Cập nhật Process staging: SUCCESS * Thêm Process Warehouse vào, với status READY * Ghi Log vào db\_controller * Gửi Email thông báo thành công. |
| 4 | LoadDataWarhouse | RUNNING | * Không chạy process * Ghi Log vào db\_controller * Gửi Email “Process is already running” |
| READY | * Chuyển Process sang status RUNNING * Load dữ liệu sang Datawarehouse |
| FAILED | * Ghi Log vào db\_controller * Gửi Email “Failed” và thông tin lỗi |
| SUCCESS | * Cập nhật Process warehouse sang SUCCESS * Thêm Process datamart vào, với status READY * Ghi Log vào db\_controller * Gửi Email “SUCCESS” với số lượng dữ liệu đã load |
| 5 | LoadDataMart | READY | - Cập nhật trạng thái thành RUNNING  - Gọi phương thức xử lý dữ liệu trong Data Mart |
| RUNNING | - Không chạy process  - Ghi Log trạng thái "Process is already running"  - Gửi Email thông báo |
| FAILED | - Ghi Log lỗi  - Gửi Email thông báo lỗi |
| SUCCESS | - Ghi Log trạng thái "SUCCESS"  -Gửi Email thông báo |

* **Exception/Error**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Type** | **Exception** | **Handling** |
| 1 | Connect error | **[LoadToStaging]**  Không thể kết nối đến db\_controller hoặc db\_staging. | - Gửi email thông báo lỗi kết nối.  -Kết thúc tiến trình. |
| Data error | **[LoadToStaging]**  **-**Dữ liệu không tồn tại  - Dữ liệu không hợp lệ | -Ghi logs lỗi cụ thể |
| Load error | **[LoadToStaging]**  **-**Load dữ liệu vào staging thất bại | -Ghi logs  -Gửi mail thông báo lỗi. |
| 2 | Data error | **[LoadToDatawarehouse]**  Kiểu dữ liệu không khớp. | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
| Configuration Error | **[LoadToDatawarehouse]** Không load được File DB.properties  Dữ liệu trong DB.properties không phù hợp  Không kết nối được database | - Gửi email thông báo lỗi |
| Structure error | **[LoadToDatawarehouse]**  Không có cột dữ liệu | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
|  |  |
| Process error  (System interrupt) | **[LoadToDatawarehouse]** Không có Procedure | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
| 3 | connectConfigException | **[crawl data]** | - Gửi Email  - Dừng chương trình |
| getShortDescriptionTikiException | **[crawl data]** | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
| saveDataToCSVException | **[crawl data]** | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
| linkTikiException | **[crawl data]** | - Ghi lại Log  - Gửi Email  - Dừng chương trình |
| 4 | Connect Error | **[LoadToDatamart]**  Không thể kết nối đến db\_datamart | - Gửi email thông báo lỗi kết nối.  - Kết thúc tiến trình. |
| Data Error | **[LoadToDatamart]**  Dữ liệu không hợp lệ hoặc không tồn tại. | - Ghi log lỗi cụ thể. |
| Load Error | **[LoadToDatamart]**  Không thực hiện được load dữ liệu vào Data Mart. | - Ghi log lỗi.  - Gửi email thông báo lỗi. |
| Configuration Error | **[LoadToDatamart]**  Không load được file DB.properties hoặc dữ liệu cấu hình không phù hợp. | Gửi email thông báo lỗi cấu hình. |
| Structure Error | **[LoadToDatamart]**  Không có cột dữ liệu cần thiết trong Data Mart. | - Ghi log lỗi.  - Gửi email  - Dừng chương trình. |
| Procedure Error | **[LoadToDatamart]**  Procedure load\_to\_datamart không tồn tại hoặc gặp lỗi trong quá trình thực thi. | - Ghi log lỗi cụ thể.  - Gửi email |
| System Interrupt | **[LoadToDatamart]**  Hệ thống bị gián đoạn trong quá trình xử lý. | - Ghi log lỗi.  - Gửi email |

## How to Run Job manually

### Require

* MySQL Server
* Java JDK17

### Tạo các database

* db\_controller

|  |
| --- |
| CREATE DATABASE IF NOT EXISTS db\_controller;  USE db\_controller;  ALTER DATABASE db\_controller CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  DROP TABLE IF EXISTS logs;  DROP TABLE IF EXISTS process;  DROP TABLE IF EXISTS configs;  DROP TABLE IF EXISTS process\_flows;  CREATE TABLE configs (  id INT AUTO\_INCREMENT PRIMARY KEY,  file\_name VARCHAR(255),  source\_path VARCHAR(255),  file\_location VARCHAR(255),  backup\_path VARCHAR(255),  warehouse\_procedure VARCHAR(100),  version VARCHAR(50),  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  insert\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  update\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  );  CREATE TABLE process (  id INT AUTO\_INCREMENT PRIMARY KEY,  config\_id INT,  process\_at VARCHAR(100) COMMENT 'crawl, staging, warehouse, datamart',  status VARCHAR(100) COMMENT 'READY, RUNNING, FAILED, SUCCESS',  begin\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  update\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,    FOREIGN KEY (config\_id) REFERENCES configs(id) ON DELETE SET NULL ON UPDATE CASCADE  );  CREATE TABLE process\_flows (  id INT AUTO\_INCREMENT PRIMARY KEY,  current\_stage VARCHAR(100) NOT NULL,  next\_stage VARCHAR(100),    UNIQUE KEY unique\_flow (current\_stage, next\_stage)  );  CREATE TABLE logs (  id INT AUTO\_INCREMENT PRIMARY KEY,  process\_id INT,  message TEXT,  insert\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  level VARCHAR(100) COMMENT 'info, warn, error, debug',    FOREIGN KEY (process\_id) REFERENCES process(id) ON DELETE SET NULL ON UPDATE CASCADE  );  DROP PROCEDURE IF EXISTS insert\_next\_process;  DELIMITER $$  CREATE PROCEDURE insert\_next\_process(config\_id INT, current\_stage VARCHAR(100))  BEGIN  DECLARE next\_stage VARCHAR(100);  -- Lấy next\_stage từ bảng `process\_flows`  SELECT f.next\_stage  INTO next\_stage  FROM process\_flows f  WHERE f.current\_stage = current\_stage  LIMIT 1;  -- Nếu next\_stage không phải NULL, thêm bản ghi mới vào bảng process  IF next\_stage IS NOT NULL THEN  INSERT INTO process (config\_id, process\_at, status)  VALUES (config\_id, next\_stage, 'READY');  END IF;  END$$  DELIMITER ;  -- Tạo bảng tạm  DROP TABLE IF EXISTS db\_controller.temp\_staging;  CREATE TABLE db\_controller.temp\_staging (  `id` TEXT,  `sku` TEXT,  `product\_name` TEXT,  `short\_description` TEXT,  `price` TEXT,  `list\_price` TEXT,  `original\_price` TEXT,  `discount` TEXT,  `discount\_rate` TEXT,  `all\_time\_quantity\_sold` TEXT,  `rating\_average` TEXT,  `review\_count` TEXT,  `inventory\_status` TEXT,  `stock\_item\_qty` TEXT,  `stock\_item\_max\_sale\_qty` TEXT,  `brand\_id` TEXT,  `brand\_name` TEXT,  `url\_key` TEXT,  `url\_path` TEXT,  `thumbnail\_url` TEXT,  `options` TEXT,  `specifications` TEXT,  `variations` TEXT  );  ALTER TABLE db\_controller.temp\_staging ADD INDEX(id(255));  -- 7. Tạo đường dẫn động cho file  -- theo ngày hiện tại  -- Tạo thủ tục con để tạo tên file động  DELIMITER //  DROP PROCEDURE IF EXISTS GenerateFilePath;  CREATE PROCEDURE GenerateFilePath(IN target\_date DATE, OUT file\_path VARCHAR(500))  BEGIN  DECLARE base\_file\_path VARCHAR(500);  -- Lấy file\_path từ bảng configs  SELECT file\_location INTO base\_file\_path  FROM configs  WHERE is\_active = 1 LIMIT 1;  -- Kiểm tra nếu không có file\_path  IF base\_file\_path IS NULL THEN  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Error: No active file\_location found.'; -- Lỗi không có file\_location  END IF;  -- Tạo tên file động với định dạng 'dataLaptop\_yyyymmdd.csv'  SET file\_path = CONCAT(base\_file\_path, 'dataLaptop\_', DATE\_FORMAT(target\_date, '%Y%m%d'), '.csv');  END //  DELIMITER ;  -- 10. Load dữ liệu từ file lên  -- bảng tạm.  -- Tạo proc để load dữ liệu từ file lên bảng tạm  DELIMITER //  DROP PROCEDURE IF EXISTS LoadDataIntoTempStaging;  CREATE PROCEDURE LoadDataIntoTempStaging(IN target\_date DATE)  BEGIN  DECLARE file\_path VARCHAR(500);  DECLARE load\_sql VARCHAR(1000);  DECLARE csv\_file\_path VARCHAR(500);  -- Gọi thủ tục con GenerateFilePath để tạo tên file  CALL GenerateFilePath(target\_date, csv\_file\_path);  -- Kiểm tra nếu file\_path không được gán giá trị  IF csv\_file\_path IS NULL THEN  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Error: File path generation failed.'; -- Lỗi nếu không tạo được file path  END IF;  -- Tạo câu lệnh SQL động cho LOAD DATA LOCAL INFILE  SET load\_sql = CONCAT(  "LOAD DATA LOCAL INFILE '", csv\_file\_path, "' ",  "INTO TABLE temp\_staging ",  "FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '\"' ",  "LINES TERMINATED BY '\\n' IGNORE 1 ROWS "  );  -- Trả về câu lệnh SQL động  SELECT load\_sql AS dynamic\_sql;  END //  DELIMITER ;  -- 12. Xử lý dữ liệu trong bảng tạm.  -- tạo proc Xử lý dữ liệu trong bảng tạm  DROP PROCEDURE IF EXISTS CleanTempStagingData;  DELIMITER //  CREATE PROCEDURE CleanTempStagingData()  BEGIN  UPDATE db\_controller.temp\_staging  SET  id = COALESCE(id, '0'),  sku = COALESCE(TRIM(sku), 'N/A'),  product\_name = COALESCE(TRIM(product\_name), 'N/A'),  short\_description = COALESCE(TRIM(short\_description), 'N/A'),  price = COALESCE(NULLIF(TRIM(price), ''), '0.00'),  list\_price = COALESCE(NULLIF(TRIM(list\_price), ''), '0.00'),  original\_price = COALESCE(NULLIF(TRIM(original\_price), ''), '0.00'),  discount = COALESCE(NULLIF(TRIM(discount), ''), '0.00'),  discount\_rate = COALESCE(NULLIF(TRIM(discount\_rate), ''), '0.00'),  all\_time\_quantity\_sold = COALESCE(NULLIF(TRIM(all\_time\_quantity\_sold), ''), '0'),  rating\_average = COALESCE(NULLIF(TRIM(rating\_average), ''), '0.00'),  review\_count = COALESCE(NULLIF(TRIM(review\_count), ''), '0'),  inventory\_status = COALESCE(TRIM(inventory\_status), 'N/A'),  stock\_item\_qty = COALESCE(NULLIF(TRIM(stock\_item\_qty), ''), '0'),  stock\_item\_max\_sale\_qty = COALESCE(NULLIF(TRIM(stock\_item\_max\_sale\_qty), ''), '0'),  brand\_id = COALESCE(NULLIF(TRIM(brand\_id), ''), '0'),  brand\_name = COALESCE(TRIM(brand\_name), 'N/A'),  url\_key = COALESCE(LEFT(TRIM(url\_key), 255), 'N/A'),  url\_path = COALESCE(LEFT(TRIM(url\_path), 255), 'N/A'),  thumbnail\_url = COALESCE(TRIM(thumbnail\_url), 'N/A'),  options = COALESCE(CASE  WHEN JSON\_VALID(REPLACE(TRIM(options), "'", '"')) THEN REPLACE(TRIM(options), "'", '"')  ELSE '"N/A"'  END, '"N/A"'),  specifications = COALESCE(CASE  WHEN JSON\_VALID(REPLACE(TRIM(specifications), "'", '"')) THEN REPLACE(TRIM(specifications), "'", '"')  ELSE '"N/A"'  END, '"N/A"'),  variations = COALESCE(CASE  WHEN JSON\_VALID(REPLACE(TRIM(variations), "'", '"')) THEN REPLACE(TRIM(variations), "'", '"')  ELSE '"N/A"'  END, '"N/A"')  WHERE id IS NOT NULL;  END //  DELIMITER ; |

* Tạo db\_staging và các Procedure:

|  |
| --- |
| USE db\_staging;  -- Tạo bảng staging  DROP TABLE IF EXISTS staging\_products;  CREATE TABLE db\_staging.staging\_products (  natural\_key VARCHAR(255),  sku VARCHAR(255),  product\_name VARCHAR(255),  short\_description TEXT,  price DECIMAL(10, 2),  list\_price DECIMAL(10, 2),  original\_price DECIMAL(10, 2),  discount DECIMAL(10, 2),  discount\_rate DECIMAL(5, 2),  all\_time\_quantity\_sold DOUBLE,  rating\_average DECIMAL(3, 2),  review\_count INT,  inventory\_status VARCHAR(50),  stock\_item\_qty INT,  stock\_item\_max\_sale\_qty INT,  brand\_id INT,  brand\_name VARCHAR(255),  url\_key VARCHAR(255),  url\_path VARCHAR(255),  thumbnail\_url VARCHAR(255),  options JSON,  specifications JSON,  variations JSON  );  -- 13. Insert dữ liệu từ bảng tạm vào table staging\_products  -- Tạo proc load dữ liệu từ bảng tạm sang staging  DELIMITER //  DROP PROCEDURE IF EXISTS MoveDataToStagingProducts;  CREATE PROCEDURE MoveDataToStagingProducts()  BEGIN  INSERT INTO db\_staging.staging\_products (  natural\_key,  sku,  product\_name,  short\_description,  price,  list\_price,  original\_price,  discount,  discount\_rate,  all\_time\_quantity\_sold,  rating\_average,  review\_count,  inventory\_status,  stock\_item\_qty,  stock\_item\_max\_sale\_qty,  brand\_id,  brand\_name,  url\_key,  url\_path,  thumbnail\_url,  options,  specifications,  variations  )  SELECT  id AS natural\_key,  sku,  product\_name,  short\_description,  CAST(price AS DECIMAL(10, 2)) AS price,  CAST(list\_price AS DECIMAL(10, 2)) AS list\_price,  CAST(original\_price AS DECIMAL(10, 2)) AS original\_price,  CAST(discount AS DECIMAL(10, 2)) AS discount,  CAST(discount\_rate AS DECIMAL(5, 2)) AS discount\_rate,  CAST(all\_time\_quantity\_sold AS DOUBLE) AS all\_time\_quantity\_sold,  CAST(rating\_average AS DECIMAL(3, 2)) AS rating\_average,  CAST(review\_count AS SIGNED) AS review\_count,  inventory\_status,  CAST(stock\_item\_qty AS SIGNED) AS stock\_item\_qty,  CAST(stock\_item\_max\_sale\_qty AS SIGNED) AS stock\_item\_max\_sale\_qty,  CAST(brand\_id AS SIGNED) AS brand\_id,  brand\_name,  url\_key,  url\_path,  thumbnail\_url,  options,  specifications,  variations  FROM db\_controller.temp\_staging;  END //  DELIMITER ;  -- 17. Xóa dữ liệu trong bảng tạm.  -- Thủ tục xóa dữ liệu trong bảng tạm  DELIMITER $$  CREATE PROCEDURE CleanTempStaging()  BEGIN  -- Xóa toàn bộ dữ liệu trong bảng temp\_staging  DELETE FROM temp\_staging;  -- Thông báo thành công  SELECT 'Data in temp\_staging has been cleared.' AS Message;  END$$  DELIMITER ; |

* Tạo db\_datawarehouse

|  |
| --- |
| CREATE DATABASE IF NOT EXISTS db\_datawarehouse;  USE db\_datawarehouse;  ALTER DATABASE db\_datawarehouse CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  DROP TABLE IF EXISTS dim\_products;  DROP TABLE IF EXISTS dim\_manufacturers;  DROP TABLE IF EXISTS dim\_dates;  CREATE TABLE dim\_dates (  date\_sk INT PRIMARY KEY, -- Khóa chính, sử dụng cho định danh ngày  full\_date DATE, -- Ngày đầy đủ  day\_since\_2005 INT, -- Số ngày kể từ năm 2005  month\_since\_2005 INT, -- Số tháng kể từ năm 2005  day\_of\_week VARCHAR(10), -- Tên ngày trong tuần  calendar\_month VARCHAR(15), -- Tên tháng  calendar\_year INT, -- Năm lịch  calendar\_year\_month VARCHAR(255), -- Định dạng YYYY-MMM  day\_of\_month INT, -- Ngày trong tháng  day\_of\_year INT, -- Ngày trong năm  week\_of\_year\_sunday INT, -- Tuần của năm theo Chủ nhật  year\_week\_sunday VARCHAR(255), -- Định dạng YYYY-Www  week\_sunday\_start DATE, -- Ngày bắt đầu tuần theo Chủ nhật  week\_of\_year\_monday INT, -- Tuần của năm theo Thứ hai  year\_week\_monday VARCHAR(255), -- Định dạng YYYY-Www  week\_monday\_start DATE, -- Ngày bắt đầu tuần theo Thứ hai  quarter\_of\_year VARCHAR(255), -- Quý của năm  quarter\_since\_2005 INT, -- Quý kể từ năm 2005  holiday VARCHAR(255), -- Trạng thái ngày lễ  date\_type VARCHAR(15) -- Kiểu ngày (Weekend/Weekday)  );  LOAD DATA INFILE 'D:\\Workspace\\DataWarehouse\\21130445\_HuynhMinh\\data\\date\_dim.csv'  INTO TABLE dim\_dates  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  -- Table: dim\_manufacturers  CREATE TABLE dim\_manufacturers (  id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  natural\_key INT UNSIGNED NOT NULL,  manufacturer\_name VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci,  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  delete\_date INT,  insert\_date INT,  update\_date INT,    FOREIGN KEY (delete\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (insert\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (update\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL  );  -- Table: dim\_products  CREATE TABLE dim\_products (  id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  natural\_key INT UNSIGNED NOT NULL,  sku\_no VARCHAR(32) NOT NULL,  product\_name VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,  product\_description VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,  image\_url VARCHAR(255),  specifications JSON DEFAULT NULL,  price DECIMAL(10, 2) NOT NULL,  original\_price DECIMAL(10, 2) NOT NULL,  stock INT UNSIGNED DEFAULT '0',  manufacturer\_id INT UNSIGNED,  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  delete\_date INT,  insert\_date INT,  update\_date INT,  expired\_date DATE DEFAULT '9999-12-31',  INDEX idx\_sku\_no (sku\_no),    FOREIGN KEY (manufacturer\_id) REFERENCES dim\_manufacturers(id) ON DELETE SET NULL ON UPDATE CASCADE,  FOREIGN KEY (delete\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (insert\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (update\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL  ) COMMENT = 'SKU'; |

* Note: cần thay đường dẫn của dim\_date
* Tạo Procedure insert data to datawarehouse:

|  |
| --- |
| USE db\_datawarehouse;  DELIMITER $$  DROP PROCEDURE IF EXISTS insert\_data\_to\_datawarehouse;  CREATE PROCEDURE insert\_data\_to\_datawarehouse()  BEGIN  -- Khai báo biến người dùng  DECLARE cur\_date INT;  DECLARE manufacturers\_inserted INT DEFAULT 0;  DECLARE products\_inserted INT DEFAULT 0;  DECLARE total\_inserted INT DEFAULT 0;    -- Gán giá trị cho biến cur\_date  SELECT date\_sk INTO cur\_date  FROM dim\_dates  WHERE full\_date = CURDATE()  LIMIT 1;  -- Thêm các Manufacturer mới vào  INSERT INTO db\_datawarehouse.dim\_manufacturers (natural\_key, manufacturer\_name, is\_active, insert\_date, update\_date)  SELECT DISTINCT sp.brand\_id, sp.brand\_name, 1, cur\_date, cur\_date  FROM db\_staging.staging\_products AS sp  WHERE NOT EXISTS (  SELECT 1 FROM db\_datawarehouse.dim\_manufacturers dm  WHERE dm.natural\_key = sp.brand\_id  );  -- Lấy số lượng bản ghi đã insert  SET manufacturers\_inserted = ROW\_COUNT();  -- Thêm các Product mới vào  INSERT INTO db\_datawarehouse.dim\_products (natural\_key, sku\_no, product\_name, product\_description, image\_url, specifications,  price, original\_price, stock, manufacturer\_id, is\_active, insert\_date, update\_date)  SELECT sp.natural\_key, sp.sku, sp.product\_name, sp.short\_description, sp.thumbnail\_url, sp.specifications, sp.price, sp.original\_price,  sp.stock\_item\_qty, dm.id, 1, cur\_date, cur\_date  FROM db\_staging.staging\_products AS sp  JOIN db\_datawarehouse.dim\_manufacturers AS dm ON dm.manufacturer\_name = sp.brand\_name  WHERE NOT EXISTS (  SELECT 1 FROM db\_datawarehouse.dim\_products AS dp  WHERE dp.natural\_key = sp.natural\_key  );  -- Lấy số lượng bản ghi đã insert  SET products\_inserted = ROW\_COUNT();    SET SQL\_SAFE\_UPDATES = 0;    -- Cập nhật các sản phẩm cũ (đánh dấu hết hiệu lực nếu có sự thay đổi)  UPDATE db\_datawarehouse.dim\_products AS dp  SET dp.is\_active = 0, dp.expired\_date = CURRENT\_DATE, dp.update\_date = cur\_date  WHERE dp.expired\_date = '9999-12-31'  AND dp.is\_active = 1  AND EXISTS (  SELECT 1 FROM db\_staging.staging\_products AS sp  WHERE dp.natural\_key = sp.natural\_key  AND (dp.sku\_no <> sp.sku  OR dp.product\_name <> sp.product\_name  OR dp.product\_description <> sp.short\_description  OR dp.image\_url <> sp.thumbnail\_url  OR dp.specifications <> sp.specifications  OR dp.price <> sp.price  OR dp.original\_price <> sp.original\_price  OR dp.stock <> sp.stock\_item\_qty)  );    SET SQL\_SAFE\_UPDATES = 1;  -- Thêm bản ghi mới với các thay đổi vào dim\_products (SCD Type 2)  INSERT INTO db\_datawarehouse.dim\_products (natural\_key, sku\_no, product\_name, product\_description, image\_url, specifications,  price, original\_price, stock, manufacturer\_id, is\_active, insert\_date, update\_date, expired\_date)  SELECT sp.natural\_key, sp.sku, sp.product\_name, sp.short\_description, sp.thumbnail\_url, sp.specifications, sp.price, sp.original\_price,  sp.stock\_item\_qty, dm.id, 1, cur\_date, cur\_date, '9999-12-31'  FROM db\_staging.staging\_products AS sp  JOIN db\_datawarehouse.dim\_manufacturers AS dm ON dm.manufacturer\_name = sp.brand\_name  WHERE EXISTS (  SELECT 1 FROM db\_datawarehouse.dim\_products AS dp  WHERE dp.natural\_key = sp.natural\_key  AND dp.is\_active = 0 -- Đảm bảo chỉ chèn bản ghi thay đổi khi sản phẩm cũ bị đánh dấu không còn hiệu lực  AND (dp.sku\_no <> sp.sku  OR dp.product\_name <> sp.product\_name  OR dp.product\_description <> sp.short\_description  OR dp.image\_url <> sp.thumbnail\_url  OR dp.specifications <> sp.specifications  OR dp.price <> sp.price  OR dp.original\_price <> sp.original\_price  OR dp.stock <> sp.stock\_item\_qty)  );    SET products\_inserted = products\_inserted + ROW\_COUNT();  -- Tính tổng số bản ghi đã insert  SET total\_inserted = manufacturers\_inserted + products\_inserted;    SELECT total\_inserted;  END $$  DELIMITER ; |

|  |
| --- |
| CREATE DATABASE IF NOT EXISTS db\_datamart;  USE db\_datamart;  ALTER DATABASE db\_datamart CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  DROP TABLE IF EXISTS aggregate\_statistics;  DROP TABLE IF EXISTS dim\_dates;  CREATE TABLE dim\_dates (  date\_sk INT PRIMARY KEY, -- Khóa chính, sử dụng cho định danh ngày  full\_date DATE, -- Ngày đầy đủ  day\_since\_2005 INT, -- Số ngày kể từ năm 2005  month\_since\_2005 INT, -- Số tháng kể từ năm 2005  day\_of\_week VARCHAR(10), -- Tên ngày trong tuần  calendar\_month VARCHAR(15), -- Tên tháng  calendar\_year INT, -- Năm lịch  calendar\_year\_month VARCHAR(255), -- Định dạng YYYY-MMM  day\_of\_month INT, -- Ngày trong tháng  day\_of\_year INT, -- Ngày trong năm  week\_of\_year\_sunday INT, -- Tuần của năm theo Chủ nhật  year\_week\_sunday VARCHAR(255), -- Định dạng YYYY-Www  week\_sunday\_start DATE, -- Ngày bắt đầu tuần theo Chủ nhật  week\_of\_year\_monday INT, -- Tuần của năm theo Thứ hai  year\_week\_monday VARCHAR(255), -- Định dạng YYYY-Www  week\_monday\_start DATE, -- Ngày bắt đầu tuần theo Thứ hai  quarter\_of\_year VARCHAR(255), -- Quý của năm  quarter\_since\_2005 INT, -- Quý kể từ năm 2005  holiday VARCHAR(255), -- Trạng thái ngày lễ  date\_type VARCHAR(15) -- Kiểu ngày (Weekend/Weekday)  );  LOAD DATA INFILE "D:\\DataWarehouse (1)\\21130445\_HuynhMinh\\data\\date\_dim.csv"  INTO TABLE dim\_dates  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  CREATE TABLE aggregate\_statistics (  id INT AUTO\_INCREMENT PRIMARY KEY, -- ID duy nhất cho từng bản ghi  manufacturer\_name VARCHAR(255) NOT NULL, -- Tên nhà sản xuất  total\_products\_sold INT DEFAULT 0, -- Tổng số sản phẩm đã bán  total\_revenue DECIMAL(15, 2) DEFAULT 0.0, -- Tổng doanh thu  total\_profit DECIMAL(15, 2) DEFAULT 0.0, -- Tổng lợi nhuận  total\_inventory INT DEFAULT 0, -- Số lượng tồn kho  best\_selling\_product VARCHAR(255), -- Sản phẩm bán chạy nhất  unsold\_products INT DEFAULT 0, -- Số lượng sản phẩm không bán được  created\_at INT, -- Ngày giờ tổng hợp dữ liệu  updated\_at INT, -- Ngày giờ cập nhật    FOREIGN KEY (created\_at) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (created\_at) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL  );  USE db\_datawarehouse;  DELIMITER $$  DROP PROCEDURE IF EXISTS InsertFactSales;  CREATE PROCEDURE InsertFactSales()  BEGIN  DECLARE cur\_date INT;    -- Lấy date\_sk từ bảng dim\_dates dựa trên ngày hiện tại  SELECT date\_sk INTO cur\_date  FROM dim\_dates  WHERE full\_date = CURDATE()  LIMIT 1;  -- Chèn 30 dòng dữ liệu vào bảng fact\_sales  INSERT INTO fact\_sales (product\_id, quantity\_sold, total\_revenue, total\_cost, insert\_date, update\_date)  VALUES  (1, 50, 5000.00, 3500.00, cur\_date, cur\_date),  (2, 30, 3000.00, 2100.00, cur\_date, cur\_date),  (3, 20, 2000.00, 1500.00, cur\_date, cur\_date),  (4, 40, 4000.00, 2800.00, cur\_date, cur\_date),  (5, 25, 2500.00, 1900.00, cur\_date, cur\_date),  (153, 32, 3800.00, 2650.00, cur\_date, cur\_date),  (214, 45, 4500.00, 3150.00, cur\_date, cur\_date),  (261, 30, 1500.00, 1050.00, cur\_date, cur\_date),  (284, 65, 3500.00, 2450.00, cur\_date, cur\_date),  …  END $$  DELIMITER ;  CALL InsertFactSales();  USE db\_datamart;  DELIMITER $$ |

* Tạo Procedure insert data to datamart:

|  |
| --- |
| DROP PROCEDURE IF EXISTS load\_to\_datamart;  CREATE PROCEDURE load\_to\_datamart()  BEGIN  DECLARE cur\_date INT;  DECLARE done INT DEFAULT 0;  DECLARE manufacturer\_name VARCHAR(255);  DECLARE total\_products\_sold INT;  DECLARE total\_revenue DECIMAL(15, 2);  DECLARE total\_profit DECIMAL(15, 2);  DECLARE total\_inventory INT;  DECLARE best\_selling\_product VARCHAR(255);  DECLARE unsold\_products INT;    -- Declare the cursor for manufacturer statistics  DECLARE manufacturer\_cursor CURSOR FOR  SELECT m.manufacturer\_name,  SUM(fs.quantity\_sold) AS total\_products\_sold,  SUM(fs.total\_revenue) AS total\_revenue,  SUM(fs.profit) AS total\_profit,  SUM(p.stock) AS total\_inventory,  -- Best-selling product per manufacturer  (SELECT p.product\_name  FROM db\_datawarehouse.dim\_products p  JOIN db\_datawarehouse.fact\_sales fs ON p.id = fs.product\_id  WHERE fs.product\_id = p.id  AND p.manufacturer\_id = m.id  GROUP BY p.product\_name  ORDER BY SUM(fs.quantity\_sold) DESC  LIMIT 1) AS best\_selling\_product,  SUM(CASE WHEN fs.quantity\_sold = 0 THEN 1 ELSE 0 END) AS unsold\_products  FROM db\_datawarehouse.dim\_manufacturers m  JOIN db\_datawarehouse.dim\_products p ON m.id = p.manufacturer\_id  LEFT JOIN db\_datawarehouse.fact\_sales fs ON p.id = fs.product\_id  GROUP BY m.id, m.manufacturer\_name; -- Add m.id to GROUP BY    -- Declare a handler for the cursor to stop when done  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;    -- Open the cursor  OPEN manufacturer\_cursor;    -- Loop through the cursor and insert data into the aggregate\_statistics table  read\_loop: LOOP  FETCH manufacturer\_cursor INTO manufacturer\_name, total\_products\_sold, total\_revenue, total\_profit, total\_inventory, best\_selling\_product, unsold\_products;    IF done THEN  LEAVE read\_loop;  END IF;    -- Insert the aggregated statistics into the aggregate\_statistics table  INSERT INTO aggregate\_statistics (manufacturer\_name, total\_products\_sold, total\_revenue, total\_profit, total\_inventory, best\_selling\_product, unsold\_products, created\_at, updated\_at)  VALUES (manufacturer\_name, total\_products\_sold, total\_revenue, total\_profit, total\_inventory, best\_selling\_product, unsold\_products, cur\_date, cur\_date);  END LOOP;    -- Close the cursor  CLOSE manufacturer\_cursor;  END $$  DELIMITER ; |

### 3.4.3 Load to Datawarehouse

* Cấu hình file DB.properties:

|  |
| --- |
| db\_controller.host = 127.0.0.1  db\_controller.port = 3306  db\_controller.username = root  db\_controller.password =  db\_controller.name = db\_controller  db\_datawarehouse.host = 127.0.0.1  db\_datawarehouse.port = 3306  db\_datawarehouse.username = root  db\_datawarehouse.password =  db\_datawarehouse.name = db\_datawarehouse  email = 21130445@st.hcmuaf.edu.vn |

Note: email là email sẽ nhận thông báo

* Chạy file **warehouseloader.exe**

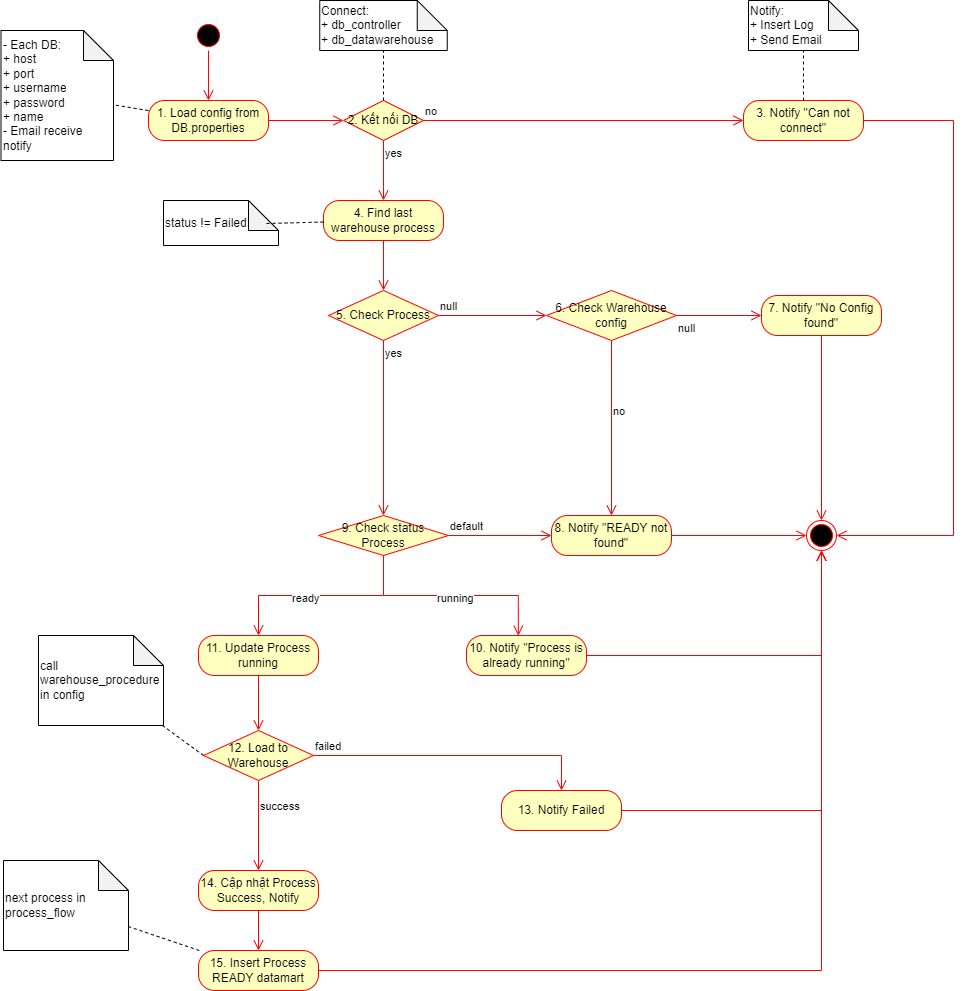
**Note:**

* Cần đảm bảo kết nối được với db\_controller, db\_datawarehouse
* Cần đảm bảo trong db\_staging có dữ liệu
* Cần đảm bảo đã tạo procedure

## Work flow

### 3.5.1 Load to Datawarehouse

Work flow load data to datawarehouse: [Link](https://drive.google.com/file/d/1lvudpV6Z-7d3f8brlyXOi5bwufSXrWtB/view?usp=drive_link)



(1)

*/\*\**

*\* 1. Load config from DB.properties*

*\*/*

static {

File f = null;

try {

f = new File(DBProperties.class.getProtectionDomain().getCodeSource().getLocation().toURI());

String propDir = f.getParentFile().getAbsolutePath() + "\\DB.properties";

File proFile = new File(propDir);

*loadProperties*(proFile);

} catch (IOException | URISyntaxException e) {

f = new File(System.*getProperty*("user.dir"));

String propDir = f.getParentFile().getAbsolutePath() + "\\DB.properties";

File proFile = new File(propDir);

try {

*loadProperties*(proFile);

} catch (IOException ex) {

throw new RuntimeException(ex);

}

}

}

(2)

*/\*\**

*\* 2. Kết nối DB*

*\*/*

private static Jdbi createConnection(String host, String port, String dbName, String username, String password) {

MysqlDataSource dataSource = new MysqlDataSource();

dataSource.setUrl("jdbc:mysql://" + host + ":" + port + "/" + dbName);

dataSource.setUser(username);

dataSource.setPassword(password);

try {

dataSource.setAutoReconnect(true);

dataSource.setUseCompression(true);

Jdbi jdbi = Jdbi.*create*(dataSource);

jdbi.useHandle(handle -> handle.execute("SELECT 1"));

return jdbi;

}

(3)

catch (Exception e) {

*/\*\**

*\* 3. Notify "Can not connect"*

*\*/*

EmailService.*sendEmail*("Load To DataWarehouse Process Failed", "Can not connect to " + dbName, LocalDateTime.*now*());

return null;

}

}

(4)

*/\*\**

*\* 4. Find last warehouse process*

*\*/*

Process process = *processDAO*.findLast("warehouse");

Log log = new Log();

(5)

*/\*\**

*\* 5. Check Process*

*\*/*

if (process == null)

(6)

*/\*\**

*\* 6. Check warehouse config*

*\*/*

if (*configDAO*.findLast("warehouse") == null) {

}

(7)

*/\*\**

*\* 7. Notify "No Config found"*

*\*/*

log.setMessage("No Config found.");

EmailService.*sendEmail*("Load To DataWarehouse Process Failed", "No Config found.", LocalDateTime.*now*());

(8)

*/\*\**

*\* 8. Notify "READY not found"*

*\*/*

log.setMessage("READY warehouse process not found");

(9)

*/\*\**

*\* 9. Check status Process*

*\*/*

switch (process.getStatus().toLowerCase())

(10)

*/\*\**

*\* 10. Notify "Process is already running"*

*\*/*

case "running":

msg = "A warehouse process is already running";

break;

(11)

*/\*\**

*\* 11. Update Process running*

*\*/*

Config config = process.getConfig();

process.setStatus("RUNNING");

*processDAO*.update(process);

(12)

*/\*\**

*\* 12. Load to Warehouse*

*\*/*

totalLoaded = *warehouseDAO*.loadToWarehouse(config.getWarehouseProcedure());

(13)

*/\*\**

*\* 13. Notify Failed*

*\*/*

String msgSpecial = "PROCEDURE " + DBProperties.*dbDatawarehouse* + "." + config.getWarehouseProcedure() + " does not exists";

if (e.getMessage().contains("java.sql.SQLSyntaxErrorException")) {

log.setMessage("Failed! Error: " + msgSpecial);

process.setStatus("READY");

EmailService.*sendEmail*("Error in Warehouse Procedure", "Error: " + msgSpecial, process.getBeginDate().toLocalDateTime());

} else {

log.setMessage("Failed! Error: " + e.getMessage());

process.setStatus("FAILED");

EmailService.*sendEmail*("Error in Warehouse Procedure", "Error: " + e.getMessage(), process.getBeginDate().toLocalDateTime());

}

log.setLevel("warn");

*processDAO*.update(process);

*logDAO*.insert(log);

(14)

*/\*\**

*\* 14. Cập nhật Process SUCCESS, Notify*

*\* Insert Log*

*\*/*

process.setStatus("SUCCESS");

*processDAO*.update(process);

*logDAO*.insert(log);

(15)

*/\*\**

*\* 15. Insert Process READY cho DataMart*

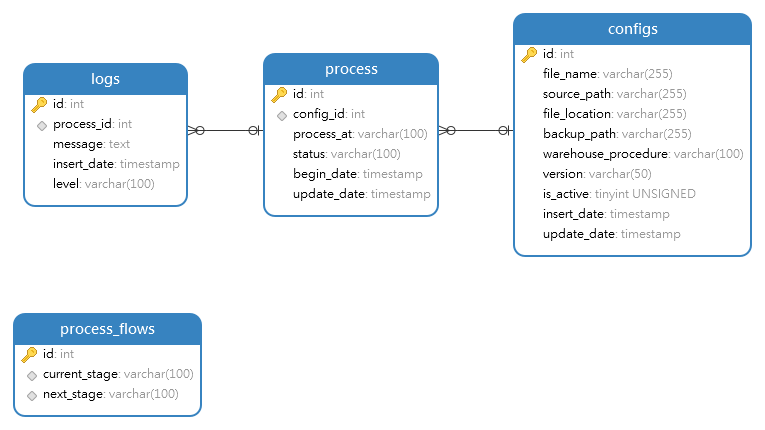
*\* Insert Log*

*\*/*

*processDAO*.insertNextProcess(process.getConfig().getId(), process.getProcessAt())

# Database Architecture

## Database Controller



**configs:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY,  AUTO\_INCREMENT | Mã Config | 1 |
| file\_name | VARCHAR(255) |  | File data | dataLaptop\_daily.csv |
| source\_path | VARCHAR(255) |  | Đường dẫn tới api crawl data | https://tiki.vn/laptop-may-vi-tinh-linh-kien/c1846 |
| file\_location | VARCHAR(255) |  | Đường dẫn tới file data crawled | D:/Workspace/DataWarehouse/data/ |
| backup\_path | VARCHAR(255) |  | Đường dẫn backup file data | D:/backup |
| warehouse\_procedure | VARCHAR(100) |  | Tên Procedure call | insert\_data\_to\_datawarehouse |
| version | VARCHAR(50) |  | Mã phiên bản | 1.0 |
| is\_active | TINYINT(1) | DEFAULT '0' | Còn sử dụng | 1 |
| insert\_date | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày thêm config | 2024-12-09 12:21:17 |
| update\_date | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày cập nhật | 2024-12-09 12:21:17 |

**process:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT | Mã process | 1 |
| config\_id | INT | FOREIGN KEY | Mã config | 1 |
| process\_at | VARCHAR(100) | craw, staging, warehouse, datamart | Chu trình nào | crawl |
| status | VARCHAR(100) | READY, RUNNING, FAILED, SUCCESS | Trạng thái | READY |
| begin\_date | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày bắt đầu process | 2024-12-09 12:21:17 |
| update\_date | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP | Ngày cập nhật | 2024-12-09 12:21:17 |

**process\_flows:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | AUTO\_INCREMENT | Mã process\_flow | 1 |
| current\_stage | VARCHAR(100) | NOT NULL | Trạng thái hiện tại | staging |
| next\_stage | VARCHAR(100) |  | Trạng thái tiếp theo trong chu trình | warehouse |

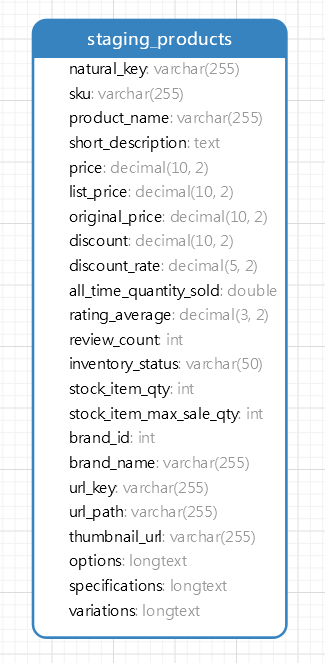
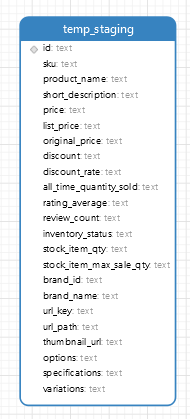
**logs:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY,  AUTO\_INCREMENT | Mã log | 1 |
| process\_id | INT | FOREIGN KEY | Mã process | 1 |
| message | TEXT |  | Tin nhắn | Load thành công |
| insert\_date | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày thêm log | 2024-12-09 12:21:17 |
| level | VARCHAR(100) | info, warn, error, debug | Mức độ của log | info |

|  |
| --- |
| Revisions:  **[2024-12-06]**  Initial |
| CREATE TABLE configs (  id INT AUTO\_INCREMENT PRIMARY KEY,  file\_name VARCHAR(255),  source\_path VARCHAR(255),  file\_location VARCHAR(255),  backup\_path VARCHAR(255),  warehouse\_procedure VARCHAR(100),  version VARCHAR(50),  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  insert\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  update\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  );  CREATE TABLE process (  id INT AUTO\_INCREMENT PRIMARY KEY,  config\_id INT,  process\_at VARCHAR(100) COMMENT 'crawl, staging, warehouse, datamart',  status VARCHAR(100) COMMENT 'READY, RUNNING, FAILED, SUCCESS',  begin\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  update\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,    FOREIGN KEY (config\_id) REFERENCES configs(id) ON DELETE SET NULL ON UPDATE CASCADE  );  CREATE TABLE process\_flows (  id INT AUTO\_INCREMENT PRIMARY KEY,  current\_stage VARCHAR(100) NOT NULL,  next\_stage VARCHAR(100),    UNIQUE KEY unique\_flow (current\_stage, next\_stage)  );  CREATE TABLE logs (  id INT AUTO\_INCREMENT PRIMARY KEY,  process\_id INT,  message TEXT,  insert\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  level VARCHAR(100) COMMENT 'info, warn, error, debug',    FOREIGN KEY (process\_id) REFERENCES process(id) ON DELETE SET NULL ON UPDATE CASCADE  ); |

## Staging

* Table temp\_staging trong db\_controller:
* Table staging\_products trong db\_staging:



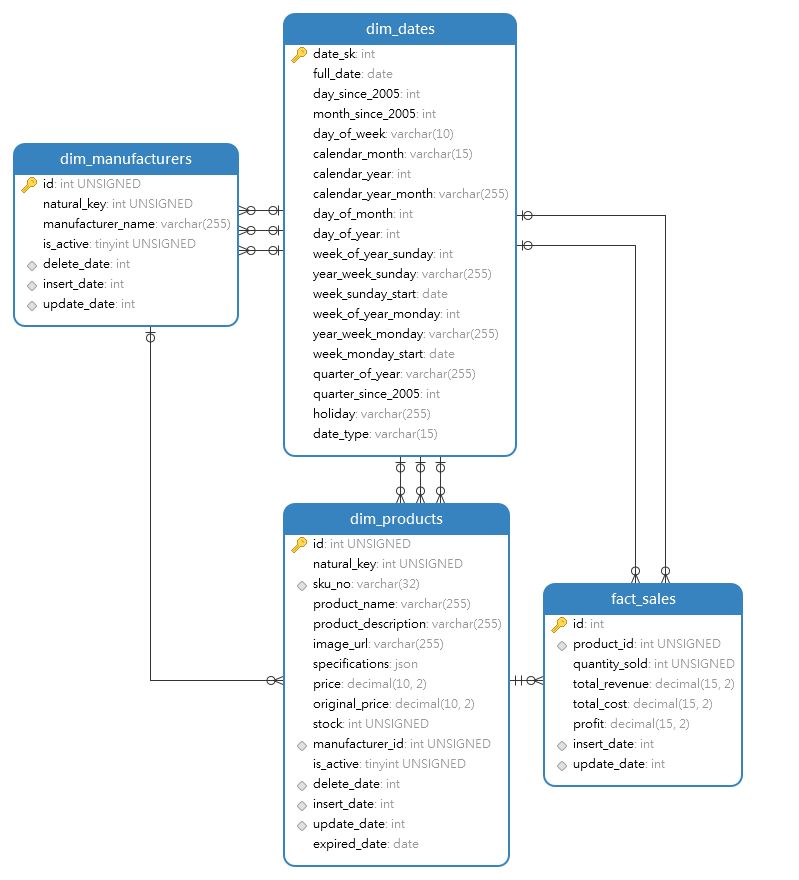
**temp\_staging:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Format** | **Description** | **Example** |
| id | TEXT | Mã sản phẩm | 124742926 |
| sku | TEXT | Mã sku | 4409272649267 |
| product\_name | TEXT | Tên sản phẩm | MacBook Air M1 13 inch 2020 |
| short\_description | TEXT | Mô tả ngắn gọn sản phẩm | MacBook Air (13 inch)Mạnh mẽ vươn tới thành công. Mà nhẹ bỗng.MacBook Air. Với chip M1.MacBook Air. Sức mạnh cho học tập.Vô cùng gọn nhẹ. Vô cùng mạnh mẽ cho việc học.Nội dung quảng cáoMacBook A... |
| price | TEXT | Giá bán hiện tại | 18190000 |
| list\_price | TEXT | Giá nhập | 24990000 |
| original\_price | TEXT | Giá gốc | 24990000 |
| discount | TEXT | Số tiền giảm giá | 6800000 |
| discount\_rate | TEXT | Phần trăm giảm giá | 27 |
| all\_time\_quantity\_sold | TEXT | Số lượng đã bán | 100 |
| rating\_average | TEXT | Đánh giá trung bình | 4.5 |
| review\_count | TEXT | Số lượng đánh giá | 56 |
| inventory\_status | TEXT | Trạng thái sản phẩm | available |
| stock\_item\_qty | TEXT | Số lượng tồn kho | 1000 |
| stock\_item\_max\_sale\_qty | TEXT | Số lượng tối đa có thể bán | 1000 |
| brand\_id | TEXT | Mã thương hiệu | 17827 |
| brand\_name | TEXT | Tên thương hiệu | Apple |
| url\_key | TEXT | Key-search | macbook-air-m1-13-inch-2020-p124742926 |
| url\_path | TEXT | Link sản phẩm | macbook-air-m1-13-inch-2020-p124742926.html?spid=73795623 |
| thumbnail\_url | TEXT | Link ảnh sản phẩm | <https://salt.tikicdn.com/cache/280x280/ts/product/0b/1a/98/b36c67698cc86e194d17ef9e0c6783a5.jpg> |
| options | TEXT | Các biến thể của sản phẩm | {'Màu': ['Space Gray'], 'Dung lượng': ['8GB / 256GB']} |
| specifications | TEXT | Thông tin cấu hình chi tiết | {'Công nghệ âm thanh': ' 3 microphones, Headphones , Loa kép (2 kênh)',(Made in)': 'Trung Quốc', 'RAM': '8GB', 'Độ phân giải': '1600 x 2560 Pixels', 'Kích thước màn hình': '13.3 inch', 'Hệ điều hành': 'macOS', 'Wifi': '802.11ax Wi-Fi 6 wireless networking', 'Sản phẩm có được bảo hành không?': 'Có', 'Hình thức bảo hành': 'Hóa đơn', 'Thời gian bảo hành': '12 Tháng'} |
| variations | TEXT | Chi tiết các biến thể | [{'child\_id': 123859963, 'sku': '9574296074805', 'name': 'Apple MacBook Air 2020 13-inch (Apple M1 - 8GB/ 256GB) - MGN63SA/A - Space Grey', 'original\_price': 24990000, 'price': 18140000, 'thumbnail\_url': 'https://salt.tikicdn.com/cache/280x280/ts/product/0b/1a/98/b36c67698cc86e194d17ef9e0c6783a5.jpg', 'inventory\_status': 'available', 'options': 'Màu: Space Gray, Dung lượng: 8GB / 256GB'}] |

**staging:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Format** | **Description** | **Example** |
| natural\_key | VARCHAR(255) | Mã sản phẩm | 124742926 |
| sku | VARCHAR(255) | Mã sku | 4409272649267 |
| product\_name | VARCHAR(255) | Tên sản phẩm | MacBook Air M1 13 inch 2020 |
| short\_description | TEXT | Mô tả ngắn gọn sản phẩm | MacBook Air (13 inch)Mạnh mẽ vươn tới thành công. Mà nhẹ bỗng.MacBook Air. Với chip M1.MacBook Air. Sức mạnh cho học tập.Vô cùng gọn nhẹ. Vô cùng mạnh mẽ cho việc học.Nội dung quảng cáoMacBook A... |
| price | DECIMAL(10,20) | Giá bán hiện tại | 18190000 |
| list\_price | DECIMAL(10,20) | Giá nhập | 24990000 |
| original\_price | DECIMAL(10,20) | Giá gốc | 24990000 |
| discount | DECIMAL(10,20) | Số tiền giảm giá | 6800000 |
| discount\_rate | DECIMAL(5,2) | Phần trăm giảm giá | 27 |
| all\_time\_quantity\_sold | DOUBLE | Số lượng đã bán | 100 |
| rating\_average | DECIMAL(3,2) | Đánh giá trung bình | 4.5 |
| review\_count | INT(11) | Số lượng đánh giá | 56 |
| inventory\_status | VARCHAR(50) | Trạng thái sản phẩm | available |
| stock\_item\_qty | INT(11) | Số lượng tồn kho | 1000 |
| stock\_item\_max\_sale\_qty | INT(11) | Số lượng tối đa có thể bán | 1000 |
| brand\_id | INT(11) | Mã thương hiệu | 17827 |
| brand\_name | VARCHAR(50) | Tên thương hiệu | Apple |
| url\_key | VARCHAR(50) | Key-search | macbook-air-m1-13-inch-2020-p124742926 |
| url\_path | VARCHAR(50) | Link sản phẩm | macbook-air-m1-13-inch-2020-p124742926.html?spid=73795623 |
| thumbnail\_url | VARCHAR(50) | Link ảnh sản phẩm | <https://salt.tikicdn.com/cache/280x280/ts/product/0b/1a/98/b36c67698cc86e194d17ef9e0c6783a5.jpg> |
| options | JSON | Các biến thể của sản phẩm | {'Màu': ['Space Gray'], 'Dung lượng': ['8GB / 256GB']} |
| specifications | JSON | Thông tin cấu hình chi tiết | {'Công nghệ âm thanh': ' 3 microphones, Headphones , Loa kép (2 kênh)',(Made in)': 'Trung Quốc', 'RAM': '8GB', 'Độ phân giải': '1600 x 2560 Pixels', 'Kích thước màn hình': '13.3 inch', 'Hệ điều hành': 'macOS', 'Wifi': '802.11ax Wi-Fi 6 wireless networking', 'Sản phẩm có được bảo hành không?': 'Có', 'Hình thức bảo hành': 'Hóa đơn', 'Thời gian bảo hành': '12 Tháng'} |
| variations | JSON | Chi tiết các biến thể | [{'child\_id': 123859963, 'sku': '9574296074805', 'name': 'Apple MacBook Air 2020 13-inch (Apple M1 - 8GB/ 256GB) - MGN63SA/A - Space Grey', 'original\_price': 24990000, 'price': 18140000, 'thumbnail\_url': 'https://salt.tikicdn.com/cache/280x280/ts/product/0b/1a/98/b36c67698cc86e194d17ef9e0c6783a5.jpg', 'inventory\_status': 'available', 'options': 'Màu: Space Gray, Dung lượng: 8GB / 256GB'}] |

## Data warehouse



**dim\_manufacturers:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY,  AUTO\_INCREMENT | Mã manufacturer | 1 |
| natural\_key | INT | UNSIGNED NOT NULL | Mã id của dữ liệu gốc | 1 |
| manufacturer\_name | VARCHAR(255) | CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci | Tên manufacturer | Apple |
| is\_active | TINYINT(1) | UNSIGNED DEFAULT '0' | Còn sử dụng | 1 |
| delete\_date | INT | FOREIGN KEY | mã sk | 1 |
| insert\_date | INT | FOREIGN KEY | mã sk | 1 |
| update\_date | INT | FOREIGN KEY | mã sk | 1 |

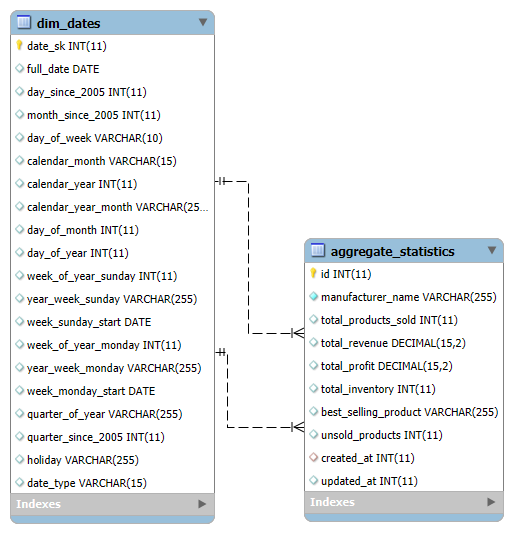
**dim\_products:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY,  AUTO\_INCREMENT | Mã sản phẩm | 1 |
| natural\_key | INT | UNSIGNED NOT NULL | Mã data ban đầu | 1 |
| sku\_no | VARCHAR(32) | NOT NULL | Mã SKU | 4409272649267 |
| product\_name | VARCHAR(255) | CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL | Tên sản phẩm | MacBook Air M1 13 inch 2020 |
| product\_description | VARCHAR(255) | CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL | Mô tả sản phẩm | MacBook Air (13 inch)Mạnh mẽ vươn tới thành công. Mà nhẹ bỗng |
| image\_url | VARCHAR(255) |  | Hình ảnh sản phẩm | http:// |
| specifications | JSON | DEFAULT NULL | Đặc điểm sản phẩm | {"CPU": "8 nhân} |
| price | DECIMAL(10, 2) | NOT NULL | Giá tiền sản phẩm | 18190000.00 |
| original\_price | DECIMAL(10, 2) | NOT NULL | Giá tiền gốc của sản phẩm | 18190000.00 |
| stock | INT | UNSIGNED DEFAULT '0' | Số lượng trong kho | 10 |
| manufacturer\_id | INT | FOREIGN KEY  , UNSIGNED | Mã id manufacturer | 1 |
| is\_active | TINYINT(1) | UNSIGNED DEFAULT '0' | Còn kích hoạt | 1 |
| delete\_date | INT | FOREIGN KEY | mã sk | 1 |
| insert\_date | INT | FOREIGN KEY | mã sk | 1 |
| update\_date | INT | FOREIGN KEY | mã sk | 1 |

**fact\_sales:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Format** | **Edit Rule** | **Description** | **Example** |
| id | INT | PRIMARY KEY,  AUTO\_INCREMENT | Mã sales | 1 |
| product\_id | INT | UNSIGNED NOT NULL | Mã sản phẩm | 1 |
| quantity\_sold | INT | UNSIGNED NOT NULL | Số lượng đã bán | 50 |
| total\_revenue | DECIMAL(15, 2) | NOT NULL | Tổng doanh thu | 5000.00 |
| total\_cost | DECIMAL(15, 2) | NOT NULL | Tổng chi phí | 3500.00 |
| profit | DECIMAL(15, 2) | (total\_revenue - total\_cost) STORED | Tổng lợi nhuận | 1500.00 |
| insert\_date | INT | FOREIGN KEY | mã sk | 1 |
| update\_date | INT | FOREIGN KEY | mã sk | 1 |

## Data Mart



|  |
| --- |
| Revisions:  **[2024-12-06]**  Initial |
| CREATE DATABASE IF NOT EXISTS db\_datawarehouse;  USE db\_datawarehouse;  ALTER DATABASE db\_datawarehouse CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  DROP TABLE IF EXISTS fact\_sales;  DROP TABLE IF EXISTS dim\_products;  DROP TABLE IF EXISTS dim\_manufacturers;  DROP TABLE IF EXISTS dim\_dates;  CREATE TABLE dim\_dates (  date\_sk INT PRIMARY KEY, -- Khóa chính, sử dụng cho định danh ngày  full\_date DATE, -- Ngày đầy đủ  day\_since\_2005 INT, -- Số ngày kể từ năm 2005  month\_since\_2005 INT, -- Số tháng kể từ năm 2005  day\_of\_week VARCHAR(10), -- Tên ngày trong tuần  calendar\_month VARCHAR(15), -- Tên tháng  calendar\_year INT, -- Năm lịch  calendar\_year\_month VARCHAR(255), -- Định dạng YYYY-MMM  day\_of\_month INT, -- Ngày trong tháng  day\_of\_year INT, -- Ngày trong năm  week\_of\_year\_sunday INT, -- Tuần của năm theo Chủ nhật  year\_week\_sunday VARCHAR(255), -- Định dạng YYYY-Www  week\_sunday\_start DATE, -- Ngày bắt đầu tuần theo Chủ nhật  week\_of\_year\_monday INT, -- Tuần của năm theo Thứ hai  year\_week\_monday VARCHAR(255), -- Định dạng YYYY-Www  week\_monday\_start DATE, -- Ngày bắt đầu tuần theo Thứ hai  quarter\_of\_year VARCHAR(255), -- Quý của năm  quarter\_since\_2005 INT, -- Quý kể từ năm 2005  holiday VARCHAR(255), -- Trạng thái ngày lễ  date\_type VARCHAR(15) -- Kiểu ngày (Weekend/Weekday)  );  LOAD DATA INFILE 'D:\\Workspace\\DataWarehouse\\21130445\_HuynhMinh\\data\\date\_dim.csv'  INTO TABLE dim\_dates  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  -- Table: dim\_manufacturers  CREATE TABLE dim\_manufacturers (  id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  natural\_key INT UNSIGNED NOT NULL,  manufacturer\_name VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci,  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  delete\_date INT,  insert\_date INT,  update\_date INT,    FOREIGN KEY (delete\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (insert\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (update\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL  );  -- Table: dim\_products  CREATE TABLE dim\_products (  id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  natural\_key INT UNSIGNED NOT NULL,  sku\_no VARCHAR(32) NOT NULL,  product\_name VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,  product\_description VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,  image\_url VARCHAR(255),  specifications JSON DEFAULT NULL,  price DECIMAL(10, 2) NOT NULL,  original\_price DECIMAL(10, 2) NOT NULL,  stock INT UNSIGNED DEFAULT '0',  manufacturer\_id INT UNSIGNED,  is\_active TINYINT(1) UNSIGNED DEFAULT '0' COMMENT '0: inactive, 1: active',  delete\_date INT,  insert\_date INT,  update\_date INT,  expired\_date DATE DEFAULT '9999-12-31',  INDEX idx\_sku\_no (sku\_no),    FOREIGN KEY (manufacturer\_id) REFERENCES dim\_manufacturers(id) ON DELETE SET NULL ON UPDATE CASCADE,  FOREIGN KEY (delete\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (insert\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (update\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL  ) COMMENT = 'SKU';  CREATE TABLE fact\_sales (  id INT AUTO\_INCREMENT PRIMARY KEY,  product\_id INT UNSIGNED NOT NULL,  quantity\_sold INT UNSIGNED NOT NULL,  total\_revenue DECIMAL(15, 2) NOT NULL,  total\_cost DECIMAL(15, 2) NOT NULL,  profit DECIMAL(15, 2) AS (total\_revenue - total\_cost) STORED,  insert\_date INT,  update\_date INT,    FOREIGN KEY (insert\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (update\_date) REFERENCES dim\_dates(date\_sk) ON DELETE SET NULL,  FOREIGN KEY (product\_id) REFERENCES dim\_products(id) ON DELETE CASCADE  ) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci; |

# Setup Processes

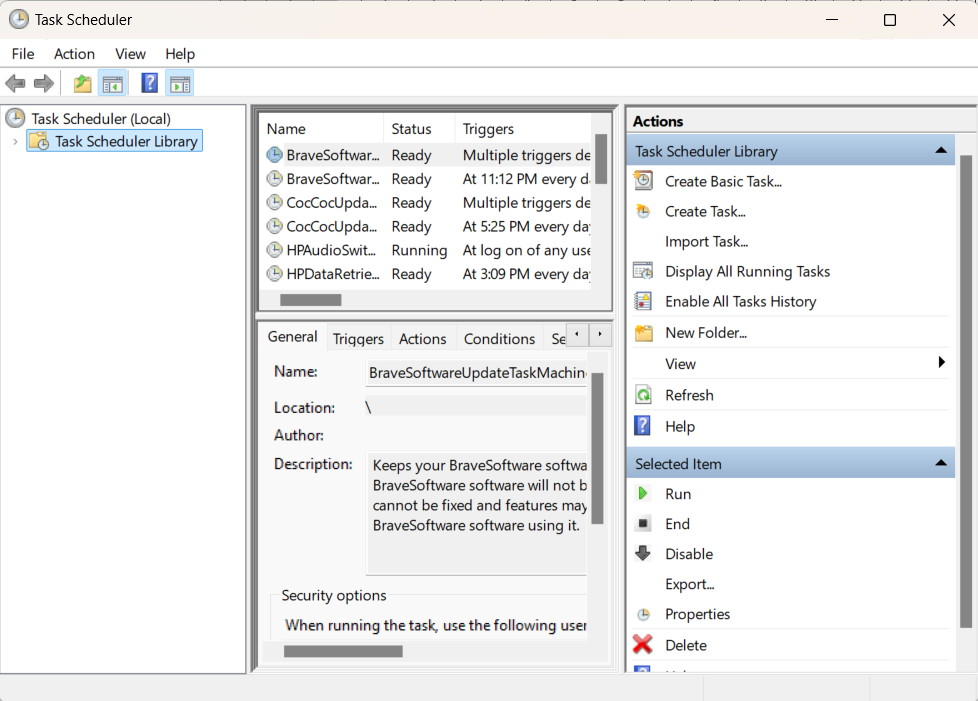
## 5.1 Chuẩn bị.

Sử dụng hệ điều hành window vào tác vụ task scheduler để thực hiện.

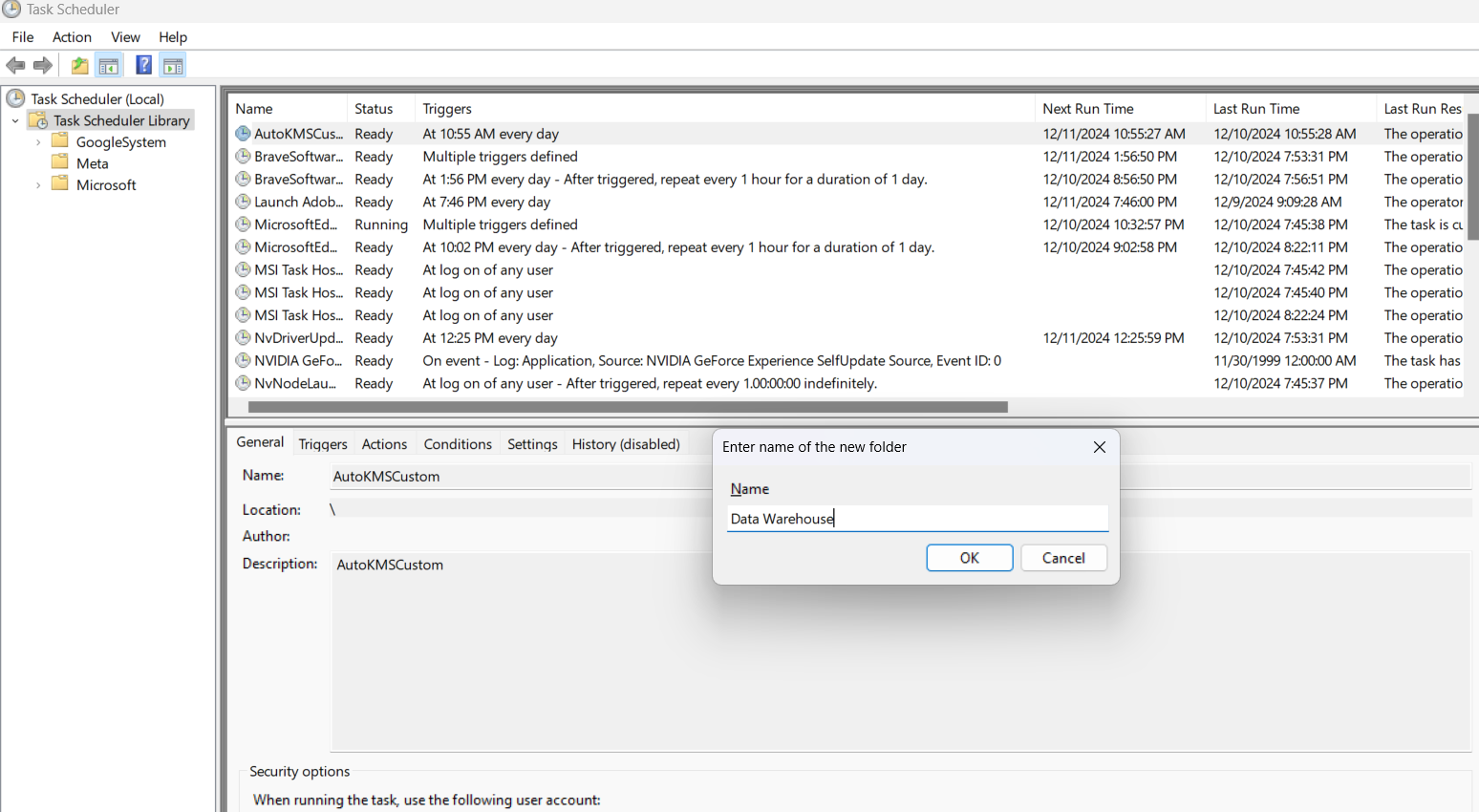
Khởi chạy Laragon

## Setup process

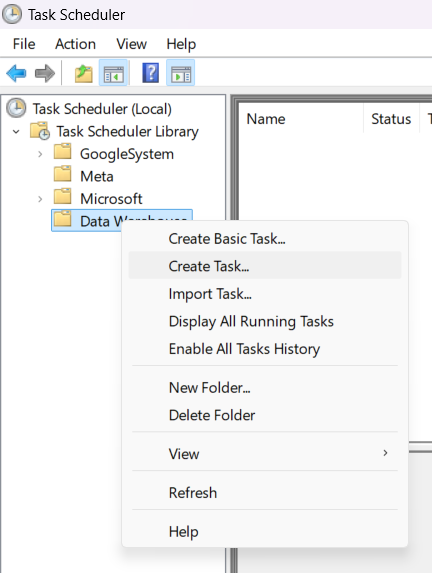
* Bước 1: Xác định đường dẫn file exe: D:\warehouseloader.exe
* Bước 2: Khởi chạy task scheduler



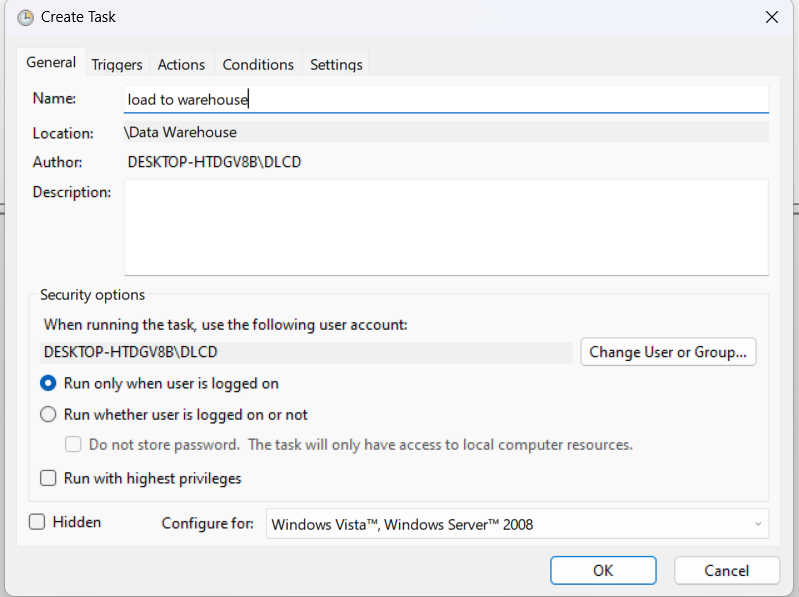
* Bước 3: Tạo thư mục Data Warehouse để làm việc. Nhấp chuột phải chọn new folder.



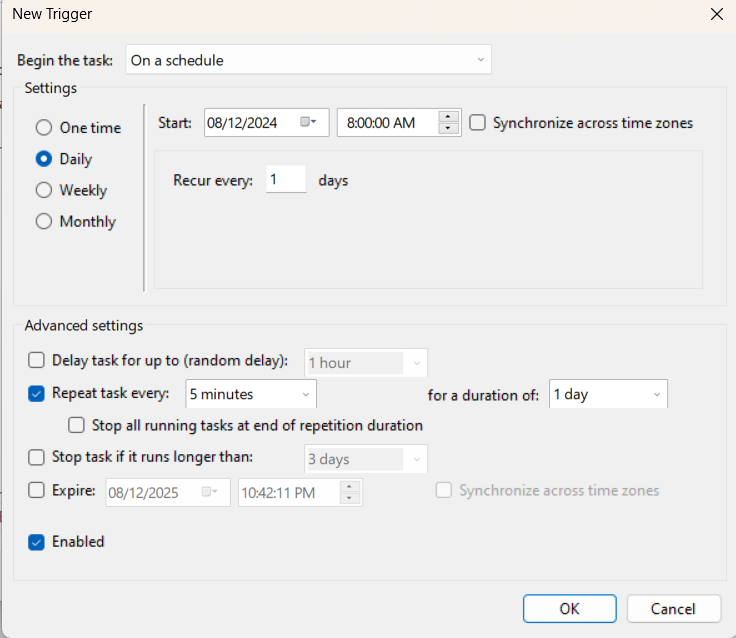
* Bước 3: Nhấp chuột phải vào khung trống và chọn create new task

****

* Bước 4: Ở tab general đặt tên task



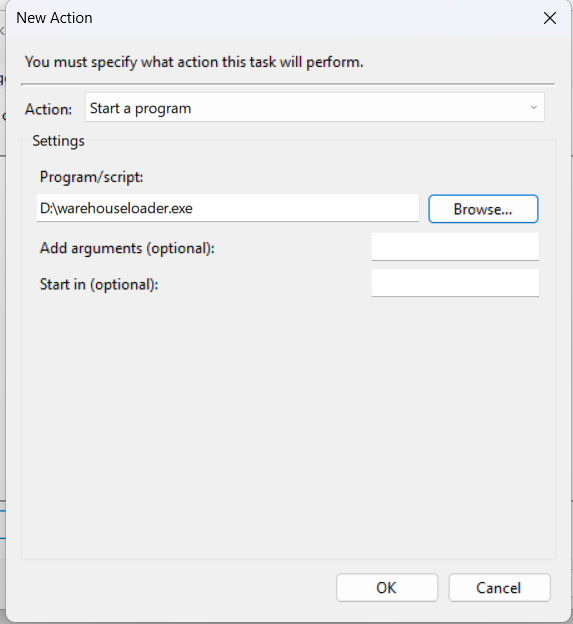
* Bước 5 Chuyển qua tab trigger nhấn new trigger cài đặt thời gian chạy cho file. Ở đây file load to datawarehouse chạy lúc 8h sáng mỗi ngày.



* Bước 6 Chuyển qua tab action

Action = Start a program

Program/script : Đường dẫn file đã ghi ở mục chuẩn bị



* Bước 7: Bấm OK để hoàn tất setup.

Hướng dẫn: <https://youtu.be/ON2uRHlt_GI?si=j6dTjuB3u24s6mNJ>