**BỘ KHOA HỌC VÀ CÔNG NGHỆ**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----------------------------



**BÁO CÁO   
ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI: QUẢN LÝ VẬT TƯ SỬ DỤNG CƠ SỞ DỮ LIỆU PHÂN TÁN**

**Môn học:** Cơ sở Dữ liệu Phân tán (INT 1414)

**Giảng viên hướng dẫn:** Lê Hà Thanh

**Thực hiện bởi nhóm sinh viên, bao gồm:**

1. Trần Nhật Quân N19DCCN153 <Trưởng nhóm>
2. <Họ tên thành viên 02> <MSSV> <Thành viên>
3. <Họ tên thành viên 03> <MSSV> <Thành viên>

**TP.HCM, tháng 6/2025**

# MỤC LỤC

[MỤC LỤC 2](#_xaddb4acds68)

[DANH SÁCH HÌNH, BẢNG 3](#_3luc1botht5j)

[TÓM TẮT 4](#_pqkl5nhxk6uu)

[Chương I. TỔNG QUAN. 5](#_norsytibmoyo)

[1. Giới thiệu đề tài. 5](#_dtahqrythqvz)

[2. Cơ sở lý thuyết. 5](#_8xowix846ixw)

[Chương II. THIẾT KẾ HỆ THỐNG 6](#_pdvkvfty5sdb)

[Chương III. TRIỂN KHAI HỆ THỐNG 7](#_20oqo41t6mxc)

[Chương IV. KẾT LUẬN 8](#_4srrr9vb6x1q)

[TÀI LIỆU THAM KHẢO 9](#_kqjnjvlraycf)

# DANH SÁCH HÌNH, BẢNG

# TÓM TẮT

# CHƯƠNG I. TỔNG QUAN

## Giới thiệu đề tài.

Trong bối cảnh các doanh nghiệp ngày càng mở rộng quy mô hoạt động với nhiều chi nhánh phân bố tại các khu vực địa lý khác nhau, nhu cầu quản lý tập trung và hiệu quả dữ liệu phát sinh từ các chi nhánh ngày càng trở nên quan trọng. Một trong những lĩnh vực đòi hỏi khả năng quản lý phân tán rõ rệt là **quản lý vật tư** — nơi dữ liệu nhập, xuất và tồn kho vật tư liên tục được cập nhật từ nhiều chi nhánh, nhưng vẫn cần được tổng hợp chính xác phục vụ điều hành và ra quyết định.

Đề tài **"Quản lý vật tư sử dụng cơ sở dữ liệu phân tán"** được xây dựng nhằm mô phỏng một hệ thống quản lý vật tư có nhiều chi nhánh hoạt động song song, ứng dụng mô hình **cơ sở dữ liệu phân tán** để đảm bảo hiệu quả về hiệu suất, độ tin cậy và tính sẵn sàng của dữ liệu.

Hệ thống bao gồm:

- Một **cơ sở dữ liệu trung tâm (master)** lưu trữ đầy đủ toàn bộ dữ liệu toàn hệ thống, phục vụ mục đích tổng hợp và sao lưu.

- Một cơ sở dữ liệu phân tán theo chiều dọc dùng để tra cứu thông tin nhân viên.

- N cơ sở dữ liệu **phân tán theo chiều ngang**, mỗi cơ sở dữ liệu tương ứng với một chi nhánh, lưu trữ thông tin về nhập, xuất vật tư và nhân viên riêng của chi nhánh đó. Trong web demo của đề tài này chúng em sẽ tạo hai cơ sở dữ liệu phân tán ngang cho hai chi nhánh.

Hệ thống web được phát triển cho phép người dùng thao tác các chức năng như quản lý vật tư, lập phiếu nhập/xuất, tìm kiếm nhân viên, và tra cứu dữ liệu theo từng chi nhánh.

## 2. Cơ sở lý thuyết.

#### a. **Cơ sở dữ liệu phân tán**

Cơ sở dữ liệu phân tán (Distributed Database – DDB) là hệ thống gồm nhiều cơ sở dữ liệu được phân bố tại các vị trí địa lý khác nhau nhưng hoạt động như một hệ thống thống nhất. Người dùng có thể truy cập dữ liệu mà không cần biết dữ liệu được lưu trữ ở đâu. Mục tiêu của DDB là nâng cao hiệu năng, tính sẵn sàng và độ tin cậy của hệ thống.

#### b. **Các loại phân mảnh dữ liệu**

**- Phân mảnh ngang**: chia dữ liệu theo hàng (record), mỗi phân mảnh chứa một tập con các bản ghi.

**- Phân mảnh dọc**: chia dữ liệu theo cột (column), mỗi phân mảnh chứa một số cột của bảng ban đầu.

**- Phân mảnh kết hợp (hybrid)**: kết hợp cả phân mảnh ngang và dọc.

Trong đề tài này:

- Dữ liệu kho, thông tin đơn hàng, nhập, xuất vật tư, nhân viên được phân mảng ngang chia theo từng chi nhánh

- Một phân mảnh dọc lưu dữ liệu nhân viên để tra cứu thông tin nhanh.

#### c. **Quản lý vật tư**

Quản lý vật tư bao gồm các nghiệp vụ: thêm vật tư, nhập kho, xuất kho, kiểm tra tồn kho, và theo dõi lịch sử giao dịch. Trong hệ phân tán, mỗi chi nhánh quản lý vật tư riêng, nhưng vẫn cần tổng hợp về hệ thống trung tâm để thống kê toàn bộ.

#### d. **Ứng dụng trong thực tế**

Việc áp dụng cơ sở dữ liệu phân tán giúp mô phỏng tốt các hệ thống lớn trong thực tế như hệ thống siêu thị, chuỗi cửa hàng, hoặc doanh nghiệp đa chi nhánh.

# CHƯƠNG II. THIẾT KẾ HỆ THỐNG

## Tổng thể hệ thống.

Hệ thống quản lý vật tư được xây dựng theo mô hình **phân tán dữ liệu**, phù hợp với tổ chức có nhiều chi nhánh hoạt động độc lập nhưng vẫn cần đồng bộ và tổng hợp dữ liệu về trung tâm. Cấu trúc tổng thể của hệ thống bao gồm ba thành phần chính:

- Web: Giao diện người dùng cho phép thao tác các chức năng như quản lý vật tư, nhập – xuất kho, tra cứu nhân viên. Ứng dụng này giao tiếp với các cơ sở dữ liệu thông qua API.

- API: Là lớp trung gian giữa frontend và các cơ sở dữ liệu. API sẽ định tuyến truy vấn đến đúng cơ sở dữ liệu tùy theo nghiệp vụ.

- Các CSDL: Nơi lưu trữ dữ liệu gồm các csdl master, csdl phân tán dọc lưu dữ liệu nhân viên, csdl phân tán ngang lưu dữ liệu đơn đặt hàng, nhập, xuất của các chi nhánh  
 - Redis: Lưu trữ tạm thời accessToken để quản lý phiên đăng nhập cho nhân viên

## Thiết kế cơ sở dữ liệu.

# Bảng: Branch

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| branchId | varchar(50) |
| name | varchar(50) |
| address | varchar(255) |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Customer

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| customerId | int |
| name | varchar(50) |
| address | varchar(255) |
| phone | varchar(255) |
| email | varchar(255) |
| note | text |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: ExportReceipt

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| exportId | varchar(50) |
| orderId | varchar(max) |
| warehouseId | int |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: ExportReceiptDetail

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| exportId | varchar(50) |
| productId | int |
| quantity | decimal(15 |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Identity

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| branchId | varchar(50) |
| num | int |
| name | varchar(255) |

# Bảng: ImportReceipt

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| importId | varchar(50) |
| orderId | varchar(max) |
| warehouseId | int |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: ImportReceiptDetail

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| importId | varchar(50) |
| productId | int |
| quantity | decimal(15 |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Order

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| orderId | varchar(50) |
| type | varchar(255) |
| status | varchar(255) |
| sourceWarehouseId | int |
| destinationWarehouseId | int |
| customerId | int |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: OrderDetail

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| orderId | varchar(50) |
| productId | int |
| quantity | decimal(15 |
| price | decimal(15 |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Product

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| productId | int |
| name | varchar(50) |
| unit | varchar(50) |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Setting

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| settingId | int |
| section | varchar(255) |
| key | text |
| value | varchar(255) |
| isHidden | tinyint |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: User

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| userId | varchar(255) |
| password | varchar(255) |
| name | varchar(50) |
| address | varchar(255) |
| phone | varchar(255) |
| email | varchar(255) |
| dob | date |
| role | int |
| branchId | varchar(50) |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

# Bảng: Warehouse

|  |  |
| --- | --- |
| Tên cột | Kiểu dữ liệu |
| warehouseId | int |
| name | varchar(50) |
| address | varchar(255) |
| branchId | varchar(50) |
| createdTime | datetime2(7) |
| updatedTime | datetime2(7) |
| createdBy | varchar(255) |
| updatedBy | varchar(255) |
| deletedTime | datetime2(7) |

## Bảo mật và phân quyền.

Hệ thống sử dụng cơ chế xác thực bằng **access token** để bảo vệ API và quản lý phiên đăng nhập của người dùng. Các token này được lưu trữ tạm thời trong **Redis** nhằm đảm bảo hiệu suất truy xuất cao và khả năng kiểm soát phiên đăng nhập. **Access token** được tạo sau khi người dùng đăng nhập thành công và được lưu trong Redis kèm theo thông tin định danh. Hệ thống có thể dễ dàng thu hồi quyền truy cập bằng việc thu hồi token đã lưu trong redis( xoá nhân viên, nhân viên đăng xuất).

# CHƯƠNG III. TRIỂN KHAI HỆ THỐNG

## Môi trường triển khai.

- CSDL: Sử dụng Microsoft SQL Server

- Redis

- Backend: Express, Node.js, Typescript

- Frontend

- Github: Lưu trữ code và quản lý phiên bản

## Các bước triển khai.

- Cài đặt Microsoft SQL Server và cấu hình các database

- Cài đặt redis

- Tạo database QLVT và cấu hình phân tán xuống các database nhánh.

- Backend:

+ Tạo file .env trong source code và điền các field cần có dựa theo file .env.example

+ Cài đặt dependencies bằng lệnh yarn install

+ Khởi chạy server backend bằng lệnh yarn start:dev

- Frontend:

# CHƯƠNG IV. KẾT LUẬN

Đề tài “Quản lý vật tư sử dụng cơ sở dữ liệu phân tán” đã giúp chúng em hiểu rõ hơn về cách tổ chức, triển khai và quản lý dữ liệu trong môi trường phân tán. Thông qua việc xây dựng hệ thống gồm một cơ sở dữ liệu trung tâm, một cơ sở dữ liệu phân tán dọc, và nhiều cơ sở dữ liệu phân tán ngang, hệ thống đã cho thấy khả năng **tăng hiệu suất**, **phân chia tải**, và **dễ dàng mở rộng theo số lượng chi nhánh**.

Tuy vẫn còn một số hạn chế về mặt xử lý đồng bộ dữ liệu và giao diện người dùng, nhưng đồ án đã đạt được mục tiêu đề ra, đồng thời là nền tảng vững chắc để phát triển các hệ thống lớn hơn trong tương lai.

# TÀI LIỆU THAM KHẢO