## **CHƯƠNG 2 : ỨNG DỤNG WINFORM**

## **1. Thiết kế NDS (Normalized Data Structure)**

NDS tập trung vào việc chuẩn hóa dữ liệu để tránh sự dư thừa và đảm bảo tính toàn vẹn của dữ liệu.

Bảng NDS:

* Customer:
  + CustomerID (PRIMARY KEY)
  + CustomerName
  + Address
  + Region
  + CustomerType
  + Phone
  + Email
* Products:
  + ProductID (PRIMARY KEY)
  + ProductName
  + CategoryID (Foreign Key from ProductCategory)
  + BrandID
  + Price
  + Stock
  + Supplier
* Branches:
  + BranchID (PRIMARY KEY)
  + BranchName
  + BranchAddress
  + Region
* Sales:
  + SalesID (PRIMARY KEY)
  + SalesDate
  + CustomerID (Foreign Key from Customer)
  + BranchID (Foreign Key from Branches)
  + ProductID (Foreign Key from Products)
  + Quantity
  + Revenue
* ProductCategory:
  + CategoryID (PRIMARY KEY)
  + CategoryName

Phân tích:

* Chuẩn hóa: Tất cả các bảng được chuẩn hóa đến ít nhất là 3NF (Third Normal Form), nghĩa là không có sự dư thừa dữ liệu, mỗi bảng chứa thông tin riêng biệt và có quan hệ rõ ràng với nhau.
* Lợi ích: Dễ dàng bảo trì, cập nhật dữ liệu, và đảm bảo rằng dữ liệu không bị trùng lặp hoặc mâu thuẫn.
* Thách thức: Truy vấn dữ liệu có thể đòi hỏi nhiều phép nối (JOIN) giữa các bảng, có thể làm giảm hiệu suất trong các hệ thống lớn.

## **2. Thiết kế DDS (Denormalized Data Structure)**

DDS tập trung vào việc giảm chuẩn hóa để cải thiện hiệu suất truy vấn, đặc biệt trong các hệ thống yêu cầu truy vấn nhanh và hiệu quả.

Bảng DDS:

* Customer: (Giữ nguyên như NDS)
  + CustomerID (PRIMARY KEY)
  + CustomerName
  + Address
  + Region
  + CustomerType
  + Phone
  + Email
* Products: (Giữ nguyên như NDS)
  + ProductID (PRIMARY KEY)
  + ProductName
  + CategoryID
  + BrandID
  + Price
  + Stock
  + Supplier
* Branches: (Giữ nguyên như NDS)
  + BranchID (PRIMARY KEY)
  + BranchName
  + BranchAddress
  + Region
* Sales: (Bảng này sẽ có thêm thông tin từ các bảng khác để giảm thiểu phép nối JOIN khi truy vấn)
  + SalesID (PRIMARY KEY)
  + SalesDate
  + CustomerID (Foreign Key từ Customer)
  + CustomerName (denormalized từ bảng Customer)
  + BranchID (Foreign Key từ Branches)
  + BranchName (denormalized từ bảng Branches)
  + ProductID (Foreign Key từ Products)
  + ProductName (denormalized từ bảng Products)
  + CategoryID (Foreign Key từ ProductCategory)
  + CategoryName (denormalized từ bảng ProductCategory)
  + Quantity
  + Revenue
* ProductCategory: (Giữ nguyên như NDS)
  + CategoryID (PRIMARY KEY)
  + CategoryName

Phân tích:

* Giảm chuẩn hóa: Bảng Sales đã được giảm chuẩn hóa để chứa thêm thông tin từ các bảng Customer, Branches, Products và ProductCategory. Điều này giúp giảm số lượng phép nối (JOIN) khi truy vấn.
* Lợi ích: Tăng tốc độ truy vấn, đặc biệt hữu ích trong các hệ thống cần báo cáo nhanh hoặc trong kho dữ liệu (data warehouse).
* Thách thức: Cập nhật và bảo trì dữ liệu phức tạp hơn do thông tin dư thừa có thể tồn tại ở nhiều nơi. Khi dữ liệu gốc thay đổi (ví dụ, tên sản phẩm), cần đảm bảo rằng các bảng denormalized cũng được cập nhật tương ứng để tránh mâu thuẫn dữ liệu.

Tóm lại:

* NDS: sử dụng trong các ứng dụng OLTP (Online Transaction Processing) hoặc khi ưu tiên tính toàn vẹn và bảo trì dễ dàng của dữ liệu.
* DDS:sử dụng trong các ứng dụng OLAP (Online Analytical Processing) và khi hiệu suất truy vấn là ưu tiên hàng đầu.

## **Tạo cơ sở dữ liệu trong SQL gồm NDS và DDS**

|  |
| --- |
| -- Tạo bảng Customer (NDS)  CREATE TABLE Customer (  CustomerID INT PRIMARY KEY,  CustomerName NVARCHAR(255),  Address NVARCHAR(255),  Region NVARCHAR(50),  CustomerType NVARCHAR(50),  Phone NVARCHAR(15),  Email NVARCHAR(255)  );  -- Tạo bảng ProductCategory (NDS)  CREATE TABLE ProductCategory (  CategoryID INT PRIMARY KEY,  CategoryName NVARCHAR(255)  );  -- Tạo bảng Products (NDS)  CREATE TABLE Products (  ProductID INT PRIMARY KEY,  ProductName NVARCHAR(255),  CategoryID INT,  BrandID INT,  Price DECIMAL(18, 2),  Stock INT,  Supplier NVARCHAR(255),  FOREIGN KEY (CategoryID) REFERENCES ProductCategory(CategoryID)  );  -- Tạo bảng Branches (NDS)  CREATE TABLE Branches (  BranchID INT PRIMARY KEY,  BranchName NVARCHAR(255),  BranchAddress NVARCHAR(255),  Region NVARCHAR(50)  );  -- Tạo bảng Sales (NDS)  CREATE TABLE Sales (  SalesID INT PRIMARY KEY,  SalesDate DATE,  CustomerID INT,  BranchID INT,  ProductID INT,  Quantity INT,  Revenue DECIMAL(18, 2),  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),  FOREIGN KEY (BranchID) REFERENCES Branches(BranchID),  FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  );  -- Tạo bảng Customer\_DDS (DDS)  CREATE TABLE Customer\_DDS (  CustomerID INT PRIMARY KEY,  CustomerName NVARCHAR(255),  Address NVARCHAR(255),  Region NVARCHAR(50),  CustomerType NVARCHAR(50),  Phone NVARCHAR(15),  Email NVARCHAR(255)  );  -- Tạo bảng Product\_DDS (DDS)  CREATE TABLE Product\_DDS (  ProductID INT PRIMARY KEY,  ProductName NVARCHAR(255),  CategoryName NVARCHAR(255), -- Denormalized từ ProductCategory  BrandID INT,  Price DECIMAL(18, 2),  Stock INT,  Supplier NVARCHAR(255)  );  -- Tạo bảng Branch\_DDS (DDS)  CREATE TABLE Branch\_DDS (  BranchID INT PRIMARY KEY,  BranchName NVARCHAR(255),  BranchAddress NVARCHAR(255),  Region NVARCHAR(50)  );  -- Tạo bảng Sales\_DDS (DDS)  CREATE TABLE Sales\_DDS (  SalesID INT PRIMARY KEY,  SalesDate DATE,  CustomerID INT,  CustomerName NVARCHAR(255), -- Denormalized từ Customer  BranchID INT,  BranchName NVARCHAR(255), -- Denormalized từ Branches  ProductID INT,  ProductName NVARCHAR(255), -- Denormalized từ Products  CategoryName NVARCHAR(255), -- Denormalized từ ProductCategory  Quantity INT,  Revenue DECIMAL(18, 2)  ); |

## **Chuyển dữ liệu từ NDS sang DDS**

|  |
| --- |
| TRUNCATE TABLE Customer\_DDS;  TRUNCATE TABLE Product\_DDS;  TRUNCATE TABLE Branch\_DDS;  TRUNCATE TABLE Sales\_DDS;  -- Chuyển dữ liệu từ Customer (NDS) sang Customer\_DDS  INSERT INTO Customer\_DDS (CustomerID, CustomerName, Address, Region, CustomerType, Phone, Email)  SELECT CustomerID, CustomerName, Address, Region, CustomerType, Phone, Email  FROM Customer;  -- Chuyển dữ liệu từ Products và ProductCategory (NDS) sang Product\_DDS  INSERT INTO Product\_DDS (ProductID, ProductName, CategoryName, BrandID, Price, Stock, Supplier)  SELECT P.ProductID, P.ProductName, C.CategoryName, P.BrandID, P.Price, P.Stock, P.Supplier  FROM Products P  JOIN ProductCategory C ON P.CategoryID = C.CategoryID;  -- Chuyển dữ liệu từ Branches (NDS) sang Branch\_DDS  INSERT INTO Branch\_DDS (BranchID, BranchName, BranchAddress, Region)  SELECT BranchID, BranchName, BranchAddress, Region  FROM Branches;  -- Chuyển dữ liệu từ Sales, Customer, Products, Branches (NDS) sang Sales\_DDS  INSERT INTO Sales\_DDS (SalesID, SalesDate, CustomerID, CustomerName, BranchID, BranchName, ProductID, ProductName, CategoryName, Quantity, Revenue)  SELECT S.SalesID, S.SalesDate, S.CustomerID, C.CustomerName, S.BranchID, B.BranchName, S.ProductID, P.ProductName, PC.CategoryName, S.Quantity, S.Revenue  FROM Sales S  JOIN Customer C ON S.CustomerID = C.CustomerID  JOIN Branches B ON S.BranchID = B.BranchID  JOIN Products P ON S.ProductID = P.ProductID  JOIN ProductCategory PC ON P.CategoryID = PC.CategoryID; |

Tạo thêm bảng user để thực hiện đăng nhập

|  |
| --- |
| CREATE TABLE Users (  UserID INT PRIMARY KEY IDENTITY,  Username NVARCHAR(50) UNIQUE NOT NULL,  PasswordHash NVARCHAR(255) NOT NULL,  Role NVARCHAR(50) NOT NULL  );  -- Thêm tài khoản vào bảng Users  INSERT INTO Users (Username, PasswordHash, Role)  VALUES  ('admin', '123', 'Admin'); |

Thật sự mà nói thì nó là một trong những điều khó nhất mà bạn có thể làm , tôi thật sự không thích ngủ một tí nào, ban thật sự quá yếu đuối để có thể làm những điều đó, bạn hãy tưởng tượng được bạn làm gì để có thể làm những điều mà tôi không mong muốn, tạo một dds data với tên gọi là databusiness sau đó khởi tạo hệ thống bằng phần mềm sql server , ở đây bây giờ bạn có thể làm những điều bạn mong muốn, sau khi kết nối với tài khoản sa window authentication, tạo một database mới với tên là ElectronicData, sau đó ấn vào nó, chọn chuột trái vào database và ấn new query, ở đây , khi bạn đã ấn new query thành công rồi thì bạn hãy nhập toàn bộ câu lệnh tạo NDS và DDS bên trên, khi đó sẽ tạo thành công các bảng dữ liệu bao gồm 5 bảng là Customer, Sale, Product, CategogyProduct,