

Chuẩn bị:

- Sử dụng SQL Server, chạy script đã cho để tạo csdl SALESMANAGEMENT gồm 2 bảng:
Category(**CatID**, CatName)
Product(**ProductID**, ProductName, UnitPrice, Quantity, CatID)
- **Sử dụng .NET 5** để viết ứng dụng WPF
- Tạo Solution tên là: Họ tên sinh viên_3 số cuối mã sinh viên, ví dụ: **NguyenXuanBinh_123**

Nộp bài: Kiểm tra đảm bảo file nén chứa solution và file word câu 2a. Nộp theo hướng dẫn của giám thị.

Câu 1: Sử dụng WPF thực hiện các yêu cầu sau:

a) (L3. 1đ) Thiết kế giao diện như sau:

The screenshot shows a WPF application window titled "Course Exam". Inside the window, there is a section titled "SALES MANAGEMENT". Below this title, there are five input fields: "Product ID", "Product Name", "Unit Price", "Quantity", and "Category" (which is a dropdown menu). Below these fields is a table with the following columns: "Product ID", "Product Name", "Category ID", "Unit Price", "Quantity", and "Amount". The table is currently empty. At the bottom of the window, there are four buttons: "Insert", "Update", "Delete", and "Search".

b) (L3. 1đ) Tạo Style và Template (đặt trong Window.Resources hoặc App.xaml) thực hiện các yêu cầu sau:

- Khi người sử dụng di chuyển con chuột lên các text box, màu chữ của textbox là **red**, màu nền là **yellow**
- Thiết lập các điều khiển Button có hình Ellip, màu của các button là: **aqua** (tương tự như hình dưới).

Insert

Update

Delete

Search

Câu 2: Viết chương trình có áp dụng LinQ và Entity Framework Core (EF) để thực hiện các yêu cầu sau:

a) (L3. 1đ) Tạo model cho cơ sở dữ liệu SALESMANAGEMENT (sử dụng thư viện Microsoft.EntityFrameworkCore.SqlServer và Microsoft.EntityFrameworkCore.Tools **phiên bản 5.0.7**).

Sau đó thêm bảng Supplier(**ID**, Name, Telephone) vào csdl. Cập nhật các lớp trong model để ánh xạ csdl SALESMANAGEMENT sau khi thêm bảng. Lưu nội dung đã cập nhật trong model ra file word có tên là câu 2a.docx và lưu vào trong thư mục chứa Project.

b) (L3. 2đ) Hiển thị các mặt hàng có quantity ≤ 150 trong DataGridView ngay khi hiển thị window. Thông tin gồm: Product ID, Product Name, Category ID, Unit Price, Quantity, Amount (= Unit Price * Quantity). Dữ liệu được sắp xếp theo Product Name tăng dần.

DataGridView được định dạng như sau:

- Tiêu đề cột: chữ đậm, căn giữa
- Dữ liệu cột thành tiền: căn phải, chữ nghiêng đậm, màu *Dark Blue*, có phân cách thành số dạng hàng nghìn, hàng triệu.

c) (L3. 1đ) Thêm 1 bản ghi vào bảng **Product** khi nhấn nút **Insert**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu.

- Xử lý ngoại lệ và kiểm tra dữ liệu người dùng nhập: Quantity và Unit Price phải là số nguyên dương.

- Hiển thị lại dữ liệu sau khi cập nhật vào csdl lên DataGridView.

d) (L3. 1đ) Sửa thông tin product được chọn trong danh sách khi nhấn nút **Update**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu, không sửa Product ID.

- Xử lý ngoại lệ và kiểm tra dữ liệu người dùng nhập: Quantity và Unit Price phải là số nguyên dương.

- Hiển thị lại dữ liệu sau khi cập nhật vào csdl lên DataGridView.

e) (L3. 1đ) Xóa Product được chọn trong danh sách khi nhấn nút **Delete**.

- Hiển thị thông báo xác nhận trước khi xóa.

- Hiển thị lại dữ liệu sau khi cập nhật vào csdl lên DataGridView.

f) (L3. 1đ) Sử dụng **ComboBox** hiển thị Category Name từ bảng **Category** nhưng khi chọn thì lấy Category ID để cập nhật dữ liệu.

g) (L3. 1đ) Khi nhấn nút **Tìm**: Hiển thị thông tin các Category lên DataGridView sang cửa sổ khác, gồm các cột: Category ID, Category Name, tổng số tiền của các mặt hàng thuộc danh mục.

Ghi chú: Thí sinh không được sử dụng tài liệu khi thực hiện bài thi kết thúc học phần.

- HẾT -