**TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG**

**KHOA: THỐNG KÊ – TIN HỌC**



**BÁO CÁO BÀI TẬP NHÓM**

**MÔN HỌC: QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ KHÁCH HÀNG NỢ TẠP HÓA CAFE**

**GVHD:** ThS. Cao Thị Nhâm

**Lớp:** 46K21.3

**Nhóm 05:**

1. Đặng Nữ Hoàng Ngọc - Leader
2. Nguyễn Thị Kim Hồng
3. Trần Thị Kiều Nhung
4. Phan Hoàng Oanh
5. Trần Thị Tằm

*Đà Nẵng, tháng 11 năm 2022*

MỤC LỤC

**[1. Hóa đơn, chứng từ 3](#_Toc26252)**

**[1.1. Hóa đơn nhập hàng 3](#_Toc9905)**

**[1.2. Hóa đơn khách hàng nợ 3](#_Toc14210)**

**[2. Thiết kế cơ sở dữ liệu 3](#_Toc19780)**

**[2.1. Thiết kế cơ sở dữ liệu ở mức khái niệm 3](#_Toc16270)**

**[2.1.1. Xây dựng ER cho hóa đơn nhập hàng 3](#_Toc26992)**

**[2.1.2. Xây dựng ER cho hóa đơn khách hàng nợ 3](#_Toc22331)**

**[2.2. Thiết kế cơ sở dữ liệu ở mức logic 3](#_Toc17014)**

**[2.3. Thiết kế cơ sở dữ liệu ở mức vật lý 3](#_Toc15903)**

**[3. Xây dựng cơ sở dữ liệu 3](#_Toc16776)**

**[4. Xác định và tạo các Index cho các thuộc tính cần thiết 3](#_Toc2254)**

**[5. Xây dựng cơ chế bảo mật tương ứng và giải thích 3](#_Toc24684)**

**[5.1. Authentication 3](#_Toc11119)**

**[5.2. Authority 3](#_Toc1598)**

**[6. Tạo Backup cho cơ sở dữ liệu 3](#_Toc28156)**

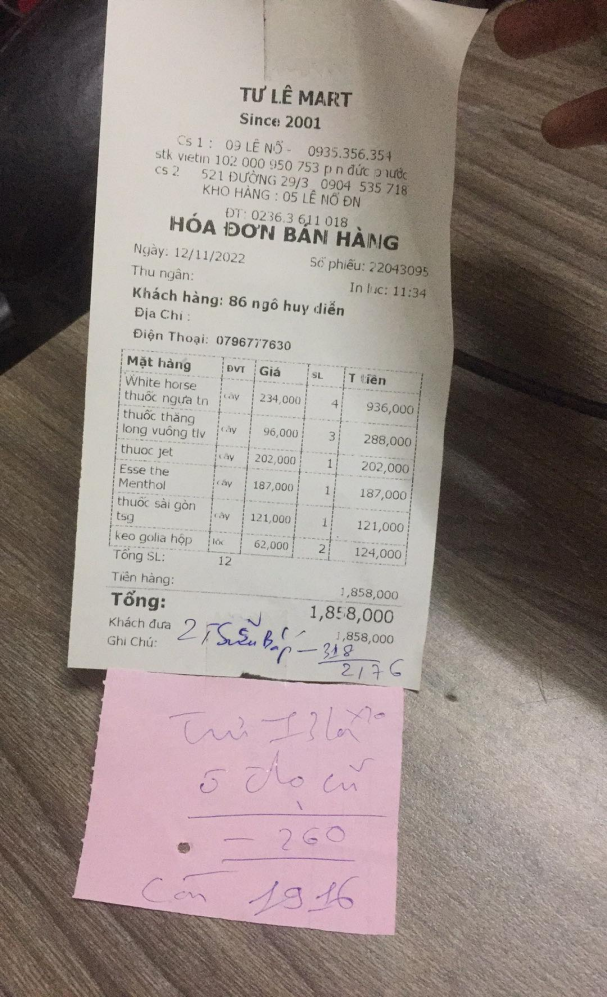
**[6.1. Backup thủ công 3](#_Toc2866)**

**[6.2. Backup tự động 3](#_Toc16240)**

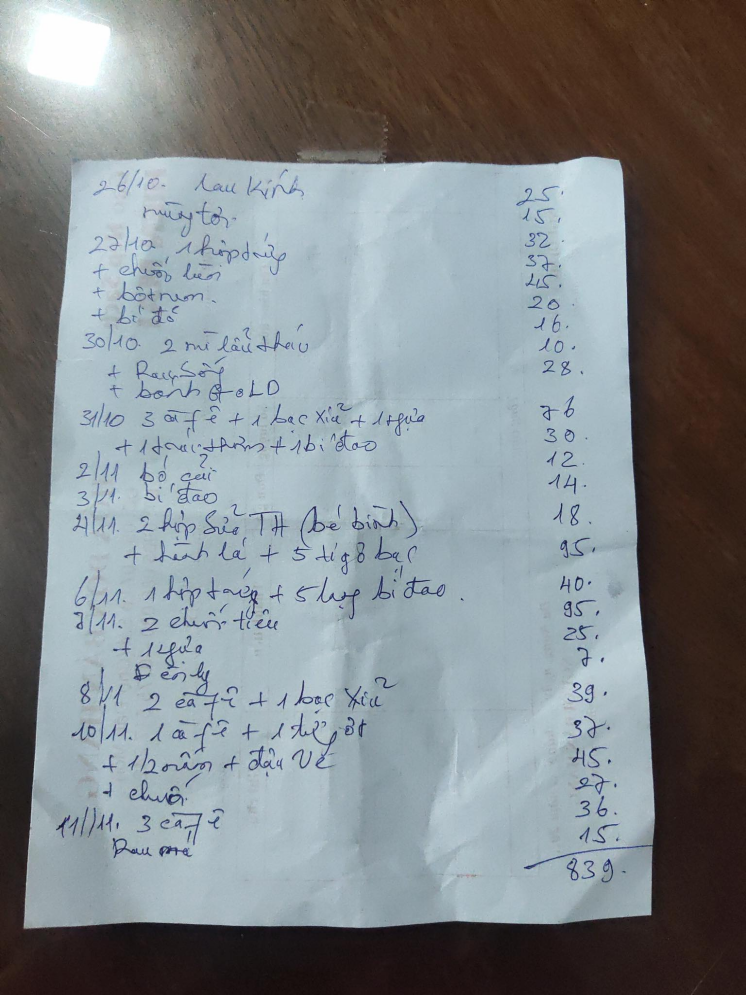
**[8. Phương pháp giải quyết vấn đề khi dữ liệu quá lớn 4](#_Toc9098)**

**[9. Phương pháp đề phòng khi phần mềm bị tấn công bằng SQL INJECTION 4](#_Toc18286)**

1. **Hóa đơn, chứng từ**
   1. **Hóa đơn nhập hàng**

****

* 1. **Hóa đơn khách hàng nợ**

****

1. **Thiết kế cơ sở dữ liệu**
   1. **Thiết kế cơ sở dữ liệu ở mức khái niệm**
      1. **Xây dựng ER cho hóa đơn nhập hàng**

|  |  |  |
| --- | --- | --- |
| **Từ trong HSDL** | **Từ rõ nghĩa** | **Viết tắt** |
| **Số** | Mã đơn nhập hàng | MaDNH |
| **Người bán hàng** | Tên nhà cung cấp | TenNCC |
| **Địa chỉ** | Địa chỉ | DiaChi |
| **Số điện thoại** | Số điện thoại | SDT |
| **Số lượng** | Số lượng nhập hàng | SoLuongNH |
| **Ngày** | Ngày nhập hàng | NgayNH |
| **Tên hàng và quy cách sản phẩm** | Tên hàng hóa | TenHang |
| **Giá đơn vị** | Đơn giá nhập | Don GiaN |
| **Thành tiền** | Thành tiền | ThanhTien |
| **Tổng cộng** | Tổng cộng tiền thanh toán | TongTien |

Ghi chú thêm: Số lượng hàng có sẵn

* **Xác định thực thể thuộc tính**

NHA CUNG CAP (MaNCC, TenNCC, DC, SDT)

HANG(TenHang, MaH, DonGiaN, SoluongHCS, SoLuongNH)

* + 1. **Xây dựng ER cho hóa đơn nợ**

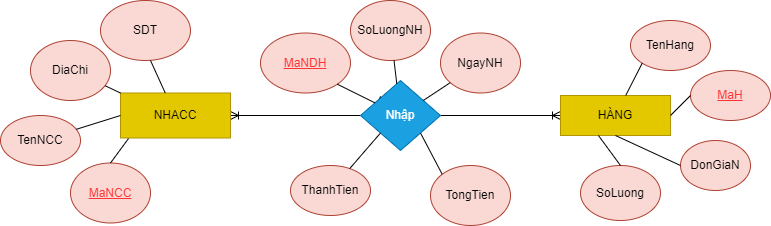
|  |  |  |
| --- | --- | --- |
| **Từ trong hóa đơn** | **Từ rõ nghĩa** | **Viết tắt** |
| **Số** | Số hóa đơn nợ | SoHDN |
| **Tên người mua nợ** | Tên người mua nợ | TenKH |
| **Địa chỉ** | Địa chỉ | Diachi |
| **Tên hàng** | Tên hàng | TenHang |
| **Ngày** | Ngày mua hàng | NgayNH |
| **Số lượng** | Số lượng nợ | SoLuongNo |
| **Đơn giá** | Đơn giá bán | DonGiaB |
| **Thành tiền** | Thành tiền | ThanhTien |
| **Tổng tiền hàng** | Tổng tiền hàng | TongTien |

**·** Ghi chú thêm: số hóa đơn, địa chỉ

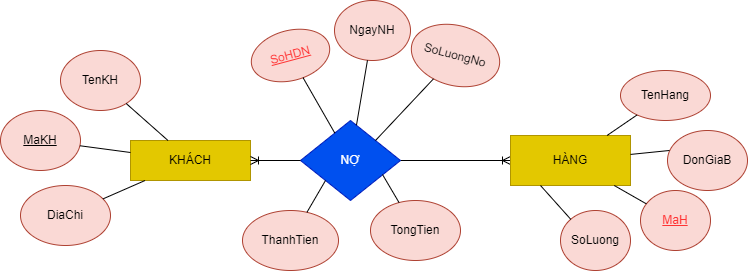
KHACH: TenKH, MaKH, Diachi

HÀNG: TenHang, MaH, DonGiaB, SoLuongNo

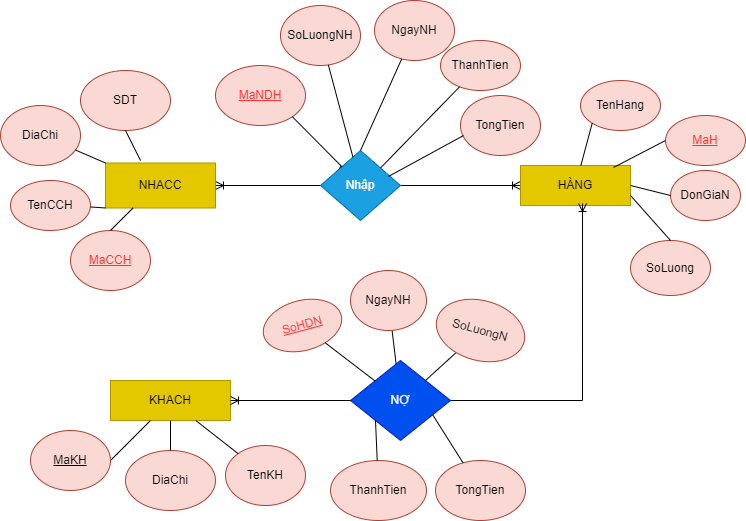
* **Sơ đồ ER-1**

****

**- Sơ đồ ER-2**



* **Tích hợp ER-1 và ER-2**



* 1. **Thiết kế cơ sở dữ liệu mức logic**
* **Chuyển thực thể ta thu được**

NHA CUNG CAP (TenNCC, MaNCC, DC, SDT)

KHACH ( TenKH, MaKH, Diachi)

HÀNG (TenHang, MaH, DonGiaN, DonGiaB, SoluongHCS)

* **Chuyển đổi quan hệ ta thu được**

NHAP (MaNDH, NgayNH, SoLuongNH, DonGiaN, ThanhTien, TongTien, MaH, MaNCC)

NO ( SoHDN , NgayNH, SoLuongN, DongiaB, ThanhTien, TongTien, MaH, MaKH)

* **Chuẩn hóa chi tiết**

Chuẩn hóa quan hệ NHẬP thu được

NHAP ( MaNDH,NgayNH, TongTien, MaNCC)

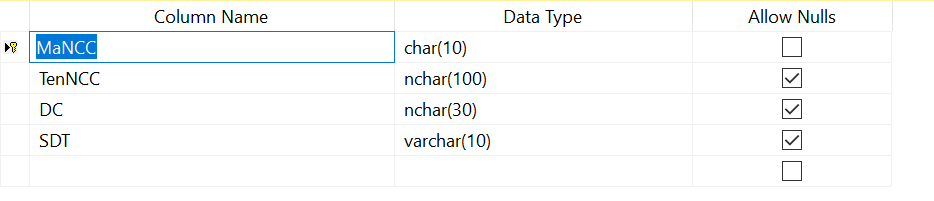
NHAP CHI TIET (MaDNHChiTiet, MaDNH, MaH, SoLuongNH, ThanhTien)

* **Chuẩn hóa quan hệ NỢ thu được**

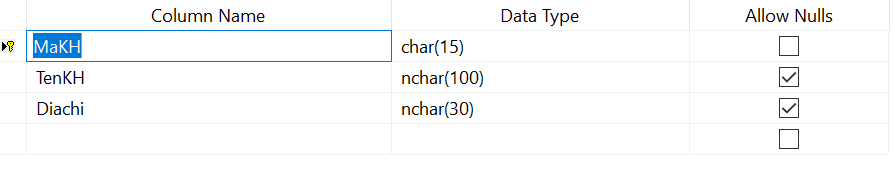
NO (SoHDN, MaKH, NgayNH, TongTien)

NO CHI TIET (SoHD ChiTiet, SoHDN, MaH, SoLuongNH, ThanhTien)

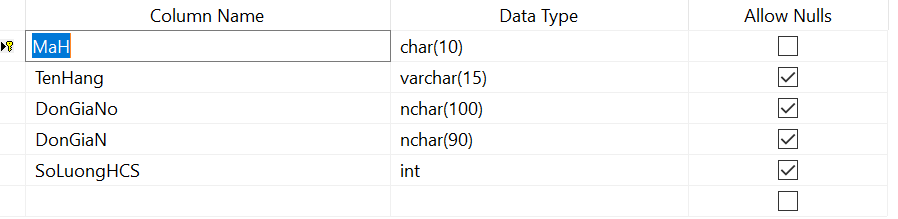
* 1. **Thiết kế cơ sở dữ liệu ở mức vật lý**
* **Bảng NHACUNGCAP**

****

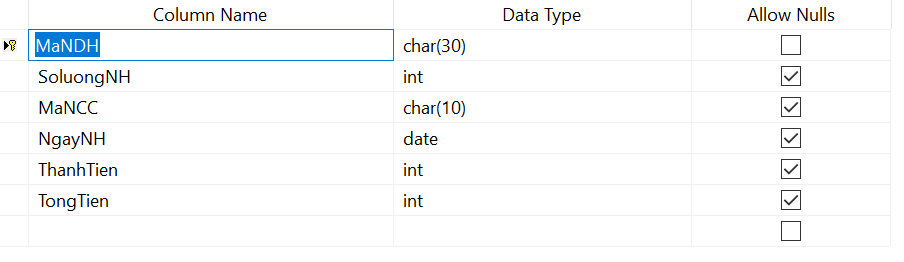
* **Bảng KHACH**

****

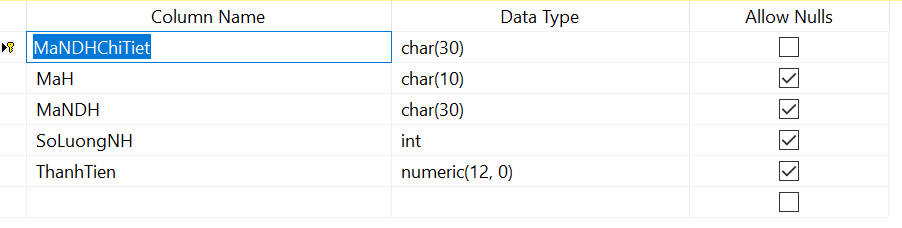
* **Bảng HANGG**

****

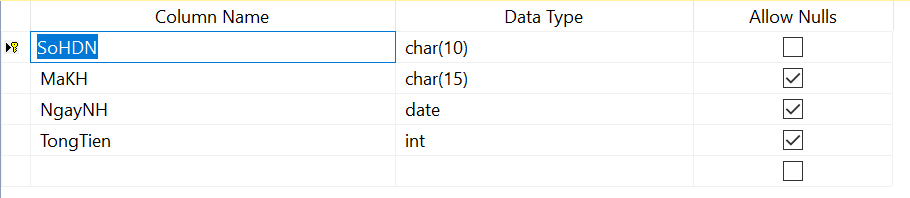
* **Bảng NHAP**

****

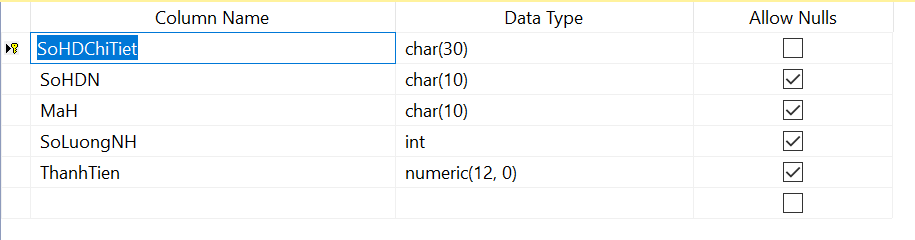
* **Bảng NHAPCHITIET**

****

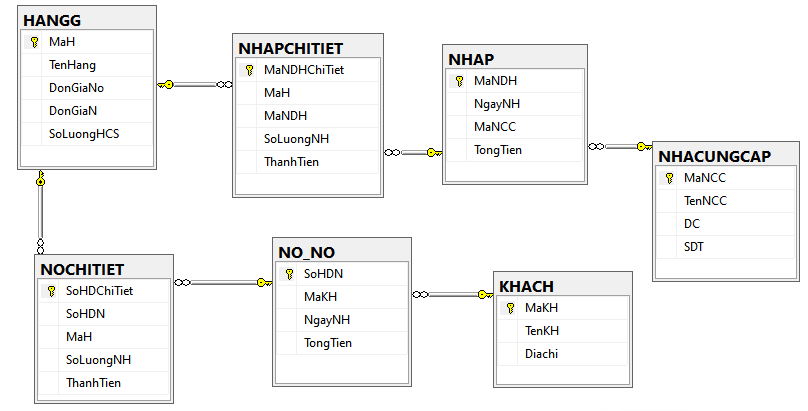
* **Bảng NO\_NO (Nợ)**

****

* **Bảng NOCHITIET**

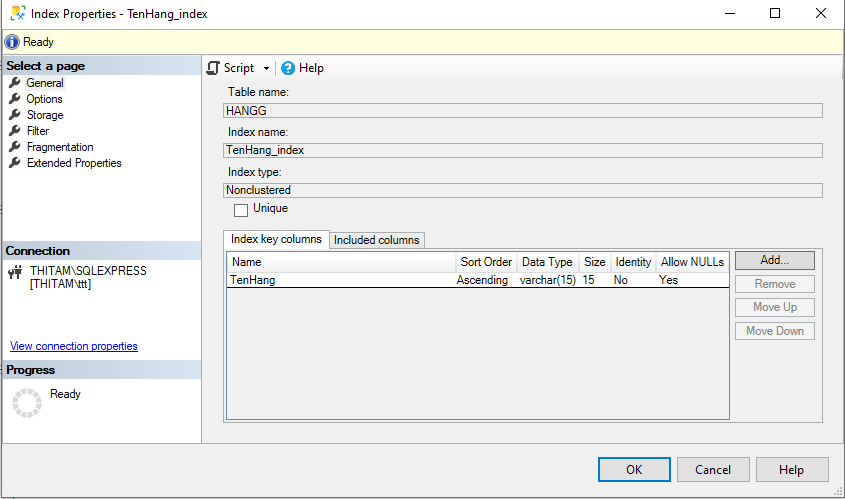
****

1. **Xây dựng cơ sở dữ liệu**

****

1. **Xác định và tạo các Index cho các thuộc tính cần thiết**

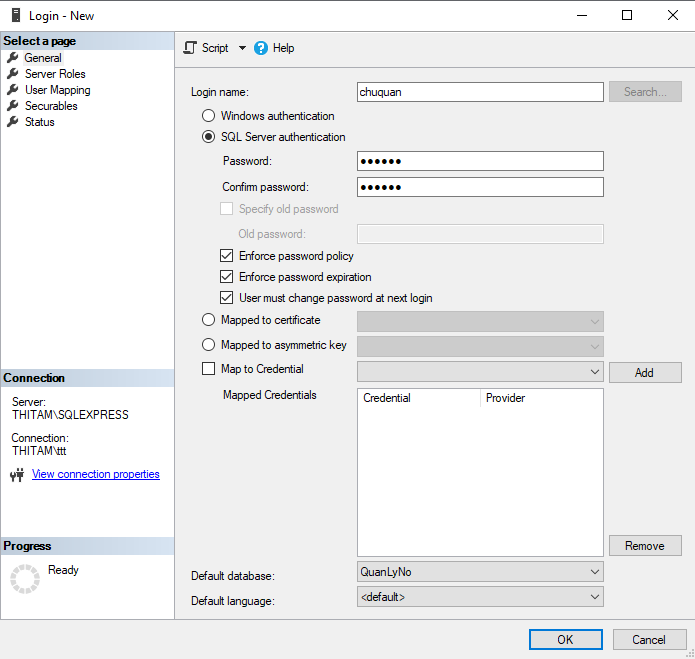
**Trong bảng HANGG chúng ta sử dụng index cho cột TenHang**

****

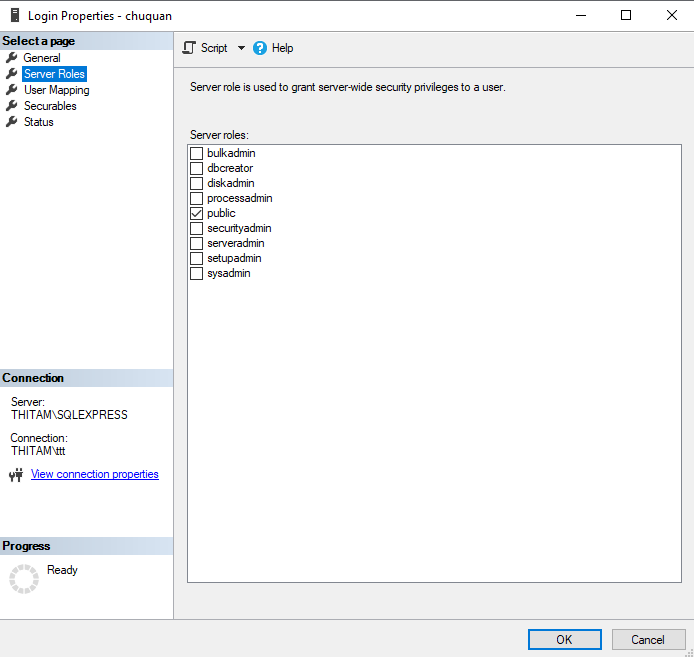
1. **Xây dựng cơ chế bảo mật tương ứng và giải thích**
   1. **Authentication**

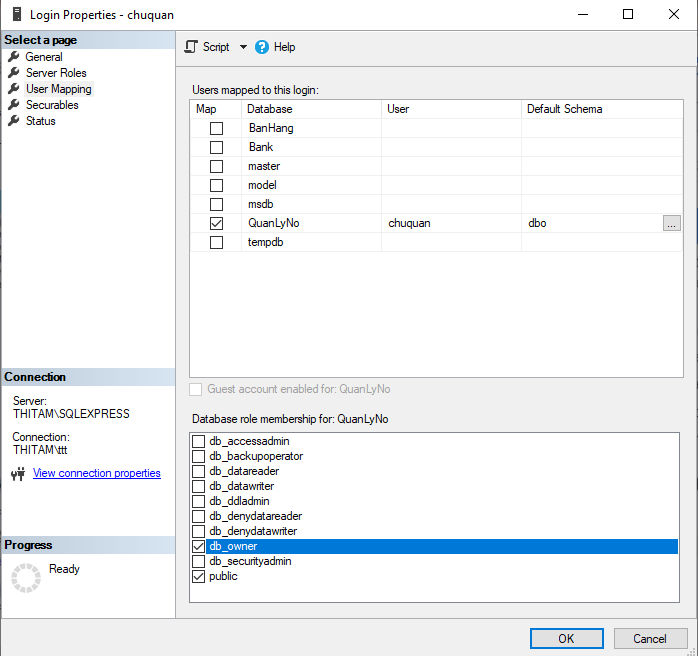
Vì với SQL Server authentication thì cơ chế giống như bình thường, cần phải cung cấp loggin name và password. SQL Server sẽ kiểm tra và quyết định có cho đăng nhập vào hay không.

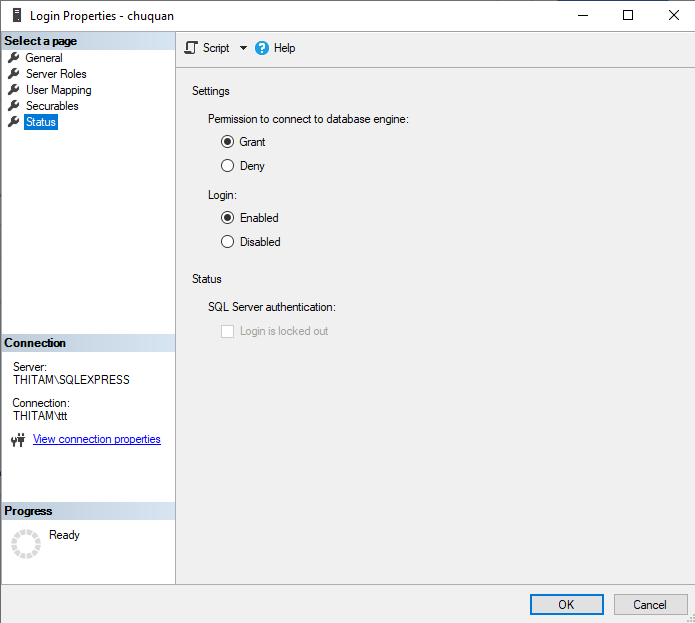
**Bước 1**: Tạo tài khoản login

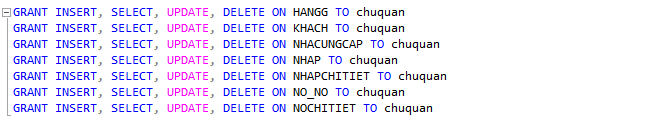
Tạo tài khoản *chuquan* với Database mặc định là QuanLyNo 

Tiếp theo, chúng ta chọn tab **Server Roles**, chọn Server roles là public.



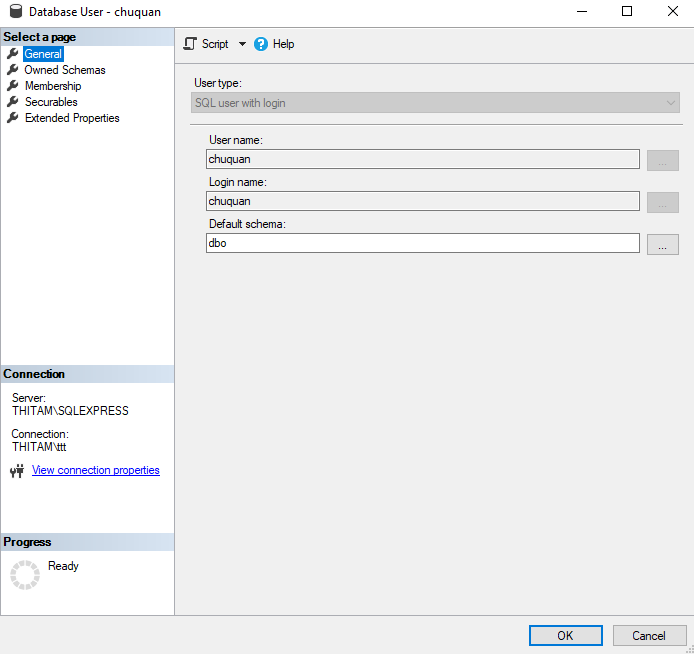
**Bước 2:** Ở page User Mapping, chọn những Database cho phép user kết nối

Để có thể login vào SQL server, ngoài quyền user phải ở trạng thái (Status) như sau:****

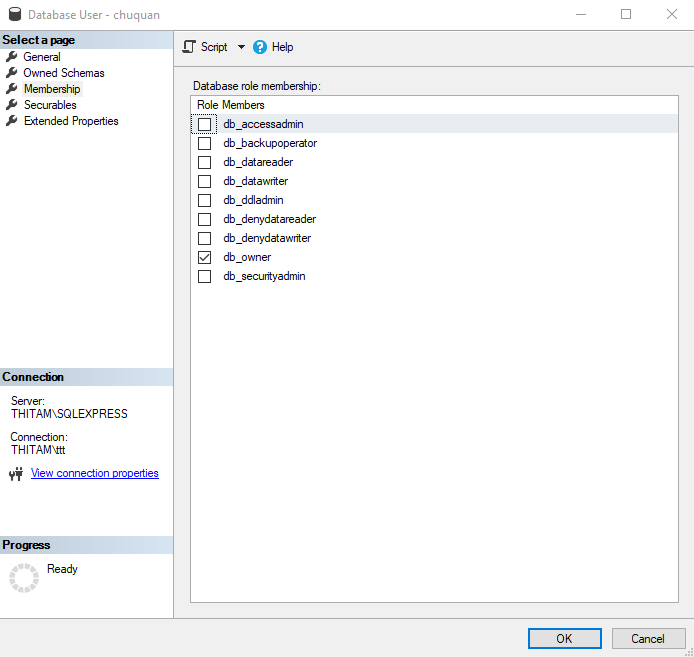
**Bước 3:**  Sử dụng lệnh T-SQL thêm quyền cho người dùng ****

**5.2. Authority**

* Tạo người sử dụng (User)

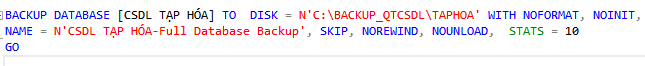
Tạo người sử dụng với Username phải trùng với login name đã tạo ở Bước 1.****

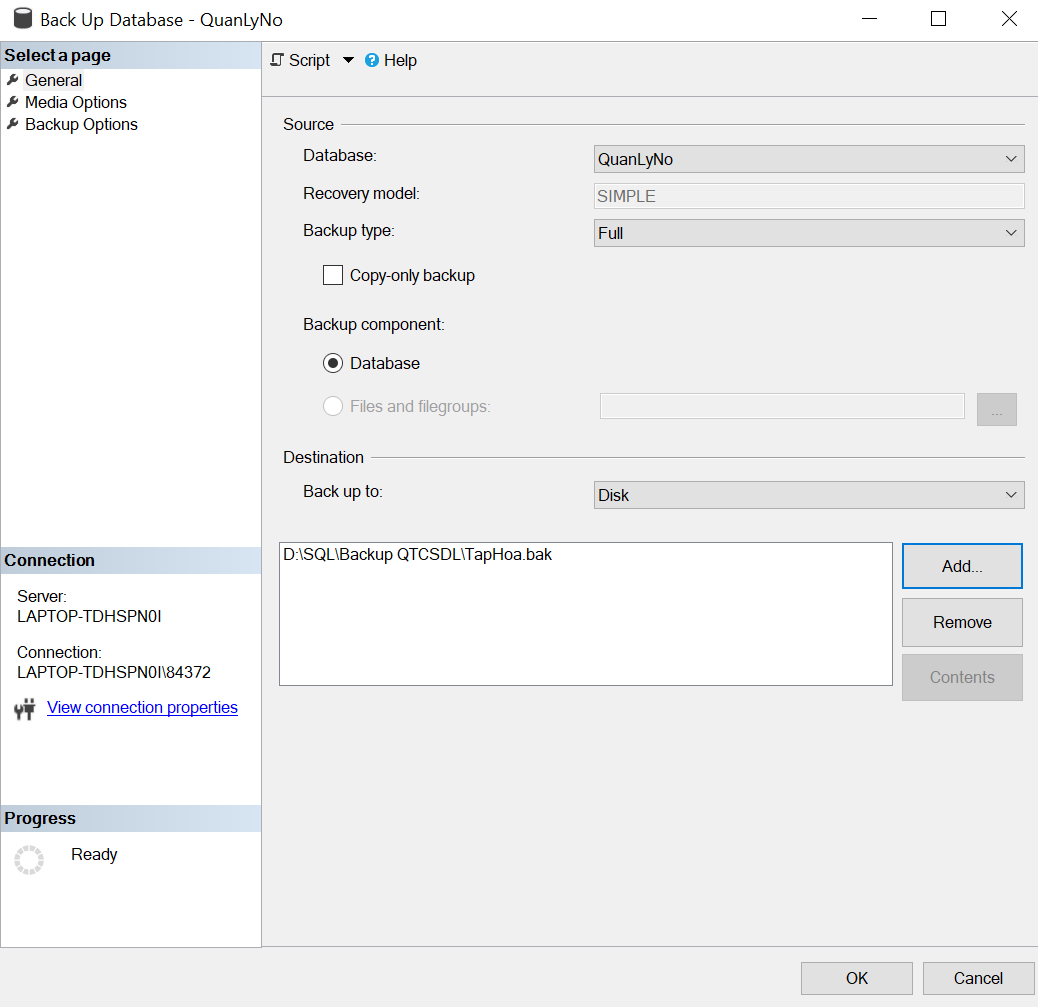
Chủ quán là người sở hữu và có mọi quyền sử dụng với Database QuanLyNo

db\_owner cho phép chủ quán có tất cả các quyền trong Database QuanLyNo****

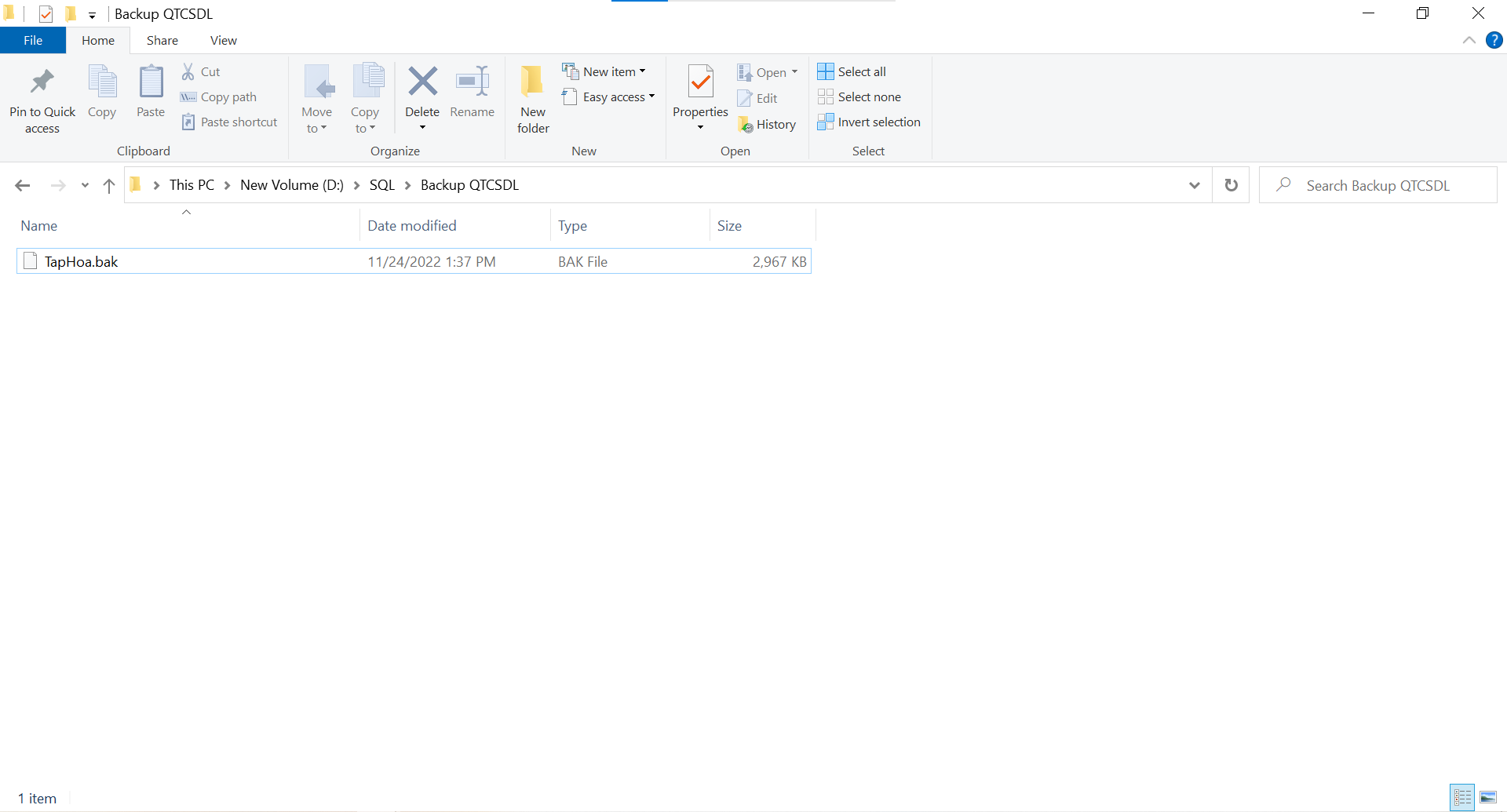
1. **Tạo Backup cho Cơ sở dữ liệu**
   1. **Backup thủ công**

**- Tạo backup cho database QuanLyNo**

****

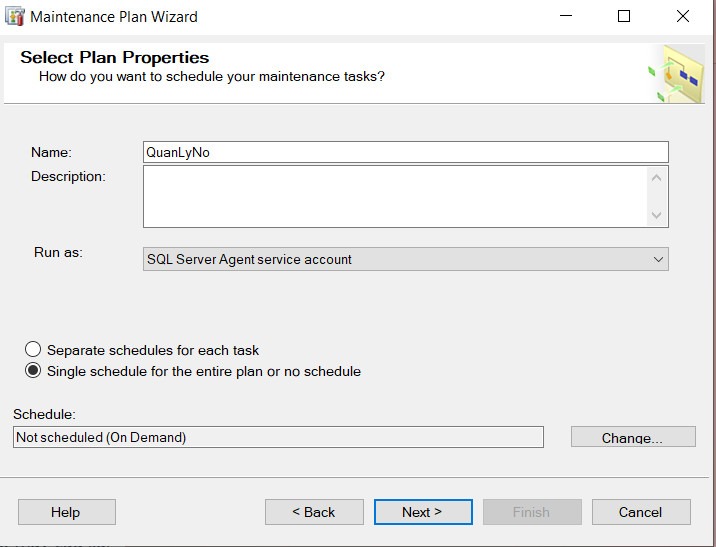
****

**File sao lưu**

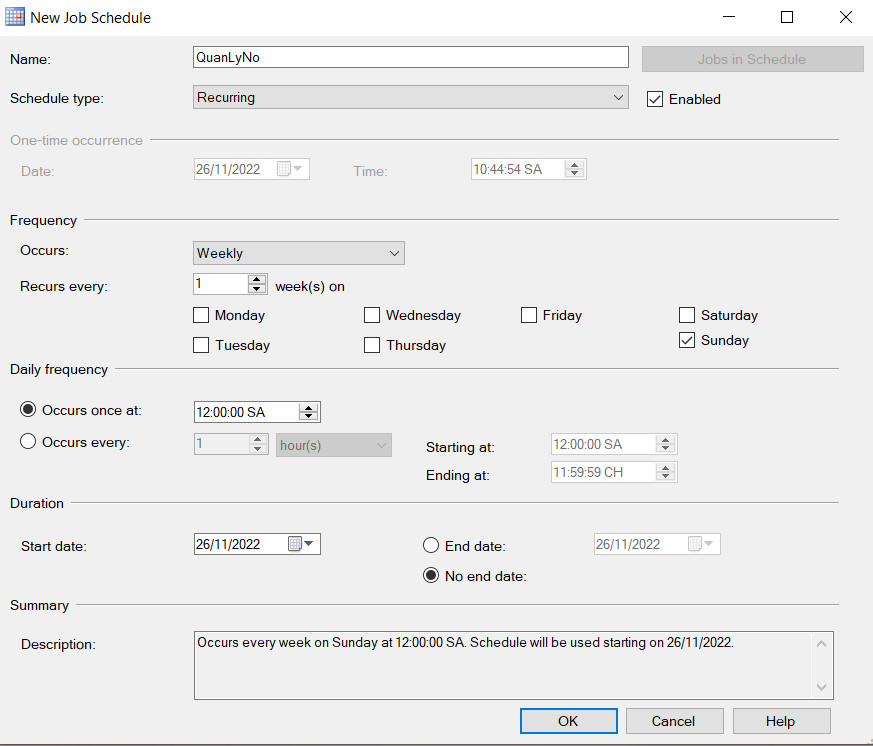
****

* 1. **Backup tự động**

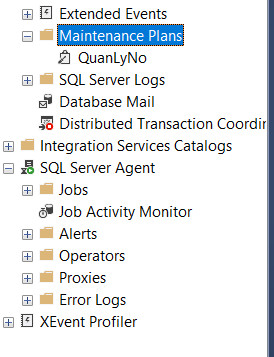
B1: Đặt tên cho backup và chọn ổ đĩa sao lưu

****

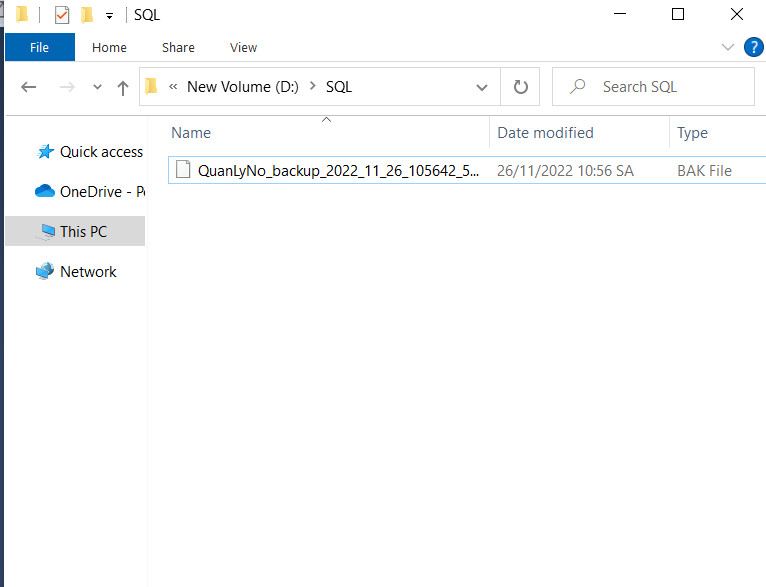
B2: Chọn thời gian backup dữ liệu

****

B3: Kiểm tra backup thành công

****

**File sao lưu**

****

1. **Phát triển ứng dụng**

**8. Phương pháp giải quyết vấn đề khi dữ liệu quá lớn**

**Giải pháp: Giảm dung lượng Log file của SQL Server**

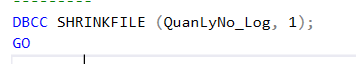
SQL Server sẽ ghi file LOG với dung lượng rất lớn, dữ liệu SQL Server có thể chỉ vài chục hoặc vài trăm Mb nhưng file LOG có thể lên đến hàng chục GB, nó sẽ lưu bất kỳ transaction nào vào LOG kể cả khi đã commit.  
Vì vậy, cách tối ưu nhất mà nhóm chúng em đã chọn lọc ra đó chính là **Giảm dung lượng Log file của SQL Server.**

**Phương pháp triển khai:**

* Bước 1: Chạy câu lệnh như hình bên dưới. SET RECOVERY của DB về SIMPLE, ở chế độ này sau khi Transaction được COMMIT, sẽ tự động xóa. Do vậy File LOG của database ở chế độ này thường rất nhỏ.



* Bước 2: Dùng DBCC SHRINKFILE để SHRINK file log xuống còn 1 Mb. Nếu không set Recovery về SIMPLE, thì sẽ không thể xóa bỏ hết các Transaction đã được COMMIT. SHRINKFILE chỉ thu dọn và sắp xếp và phân bố lại dữ liệu, bỏ các vùng trống để giải phóng bộ nhớ, chứ không phải xóa dữ liệu. Vì thế ở chế độ FULL, SHRINKFILE hầu như không tác dụng, hoặc nếu có thì file LOG dung lượng giảm đi không đáng kể.



* Bước 3: SET RECOVERY về lại FULL. Trên MSDN cũng khuyên nếu muốn Backup LOG, các bạn nên chuyển về chế độ SIMPLE, hơn là backup LOG với Truncate\_Only và No\_LOG.



*Chú thích:*

* DBCC: DataBase Consistency Check
* MSDN: Microsoft Developer Network

**9. Phương án đề phòng khi phần mềm bị tấn công bằng SQL INJECTION**

Trước khi đưa ra các phương án để đề phòng khi phần mềm bị tấn công bằng SQL INJECTION, ta cần hiểu tấn công bởi SQL Injection là gì?

SQL Injection là một kỹ thuật được thực hiện nhằm lợi dụng vào các lỗ hổng liên quan đến việc truy vấn của các ứng dụng. Nó được triển khai bằng cách chèn thêm một đoạn SQL để làm sai lệnh đi các câu truy vấn ban đầu, từ đó sẽ khai thác các dữ liệu từ database.   
Vậy tấn công bằng SQL Injection là gì? Ví dụ sau đây sẽ làm rõ hơn về nó:  
Khi click vào form Đăng nhập thì người dùng cần phải nhập dữ liệu tại button “Tìm kiếm” và sau đó sẽ nhập các văn bản tìm kiếm. Lúc này, các biểu mẫu lưu dữ liệu và toàn bộ các dữ liệu này sẽ được đi vào cơ sở dữ liệu. Đối với kẻ tấn công, thay cho việc nhập đúng dữ liệu thì chúng sẽ lợi dụng lỗ hổng để insert rồi nhập những câu lệnh SQL bất hợp pháp để lấy dữ liệu của người dùng. Khi đó, SQL Injection sẽ thực hiện với ngôn ngữ lập trình SQL và chúng được dùng với mục đích quản lý dữ liệu đã được lưu trữ trong toàn bộ các cơ sở dữ liệu.   
SQL Injection là một cách tấn công khá nguy hiểm, nó có thể hack mọi thông tin người dùng, thay đổi các dữ liệu quan trọng trong một hệ thống, nói dễ hiểu hơn là nó ăn cắp các dữ liệu của cá nhân, hệ thống hay các trang web,…  
  
Chính vì những điều trên mà ta phải tìm cách để tự vệ cũng như đưa ra giải pháp để tránh sự tấn công SQL Injection. Và phương án tốt nhất mà nhóm chúng em lựa chọn là **Backup dữ liệu thường xuyên.**Backup nghĩa là sao lưu dữ liệu trên máy tính, hiểu đơn giản là sao chép và lưu trữ tất cả các dữ liệu trên máy tính, được xem như một bản sao để dành cho việc đảm bảo rằng sẽ không mất dữ liệu hoàn toàn và vĩnh viễn. Việc lưu trữ dữ liệu dự phòng để nhằm sử dụng khi cần thiết trong trường hợp bị tấn công bằng SQL Injection.   
Giải pháp trên đã được nhóm triển khai ở mục **R6**.