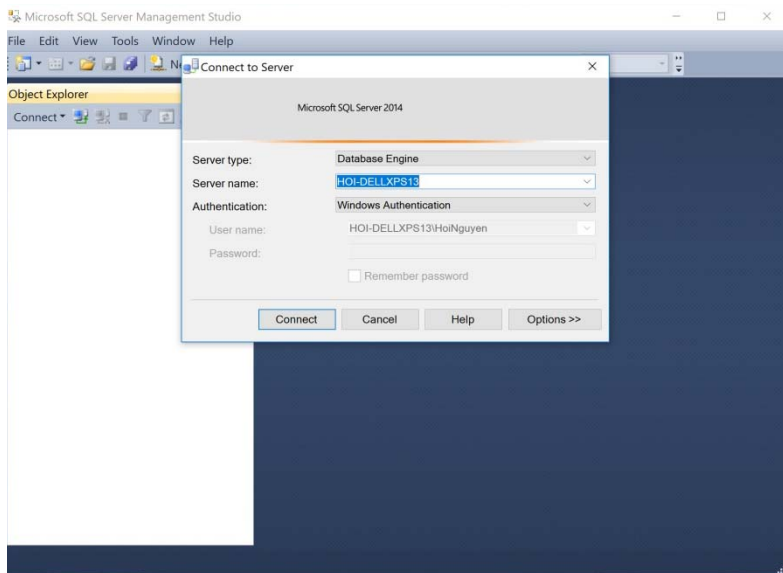
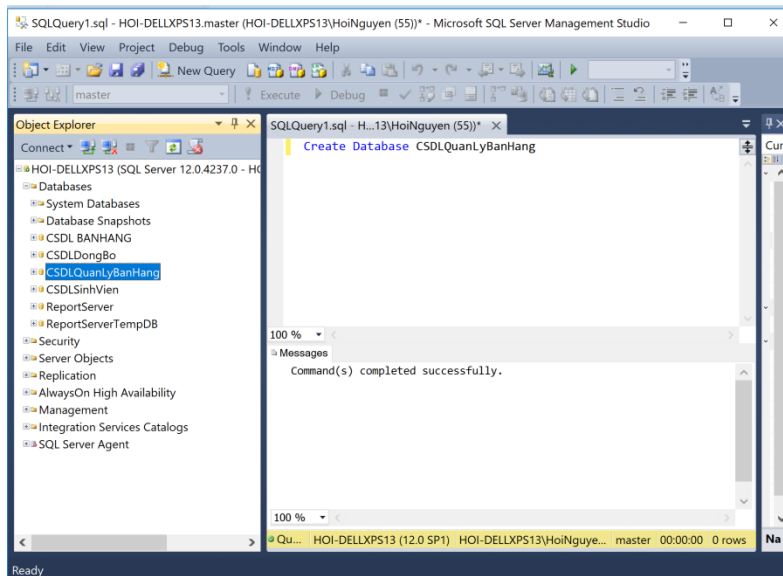


1. Khởi động SQL Server Management Studio



2. Tạo Cơ sở dữ liệu



3. Tạo các bảng

```
Create table NhanVien --Câu lệnh tạo bảng Nhân Viên
(MaNV Varchar(30) primary key,
TenNV Nvarchar(200),
SoCMTND Varchar(30),
GT Varchar(2)check (GT IN (1, 0)),
SĐT Varchar(20),
MaPhong Varchar(30),
```

```
Constraint FK Foreign key (MaPhong) References PhongBan(MaPhong)); -- Ràng buộc cho  
khóa ngoại
```

```
Create table KhachHang -- Tạo bảng Khách Hàng  
(MaKH Varchar(30) primary key,  
TenCTy Nvarchar(200),  
TenNguoiDD Nvarchar(200),  
DiaChi Nvarchar(200),  
TaiKhoan Varchar(50));
```

```
Create table Kho -- Tạo bảng Kho  
(MaKho Varchar(30) primary key,  
TenKho Nvarchar(200),  
Mota Nvarchar(200));
```

```
Create table LoaiHangHoa -- Tạo bảng LoaiHangHoa  
(MaLoai Varchar(30) primary key,  
TenLoai Nvarchar(200),  
Mota Nvarchar(200));
```

```
Create table HangHoa -- Tạo bảng Hàng Hóa  
(MaHH Varchar(30) primary key,  
TenHH Nvarchar(200),  
NgayPhatHanh SmallDateTime,  
DonGia Numeric(20,4),  
ThueVAT Numeric(20,4),  
Mota Nvarchar(200),  
TrangThai Nvarchar(200),  
ThoiGianBH Nvarchar(50),  
MaLoai Varchar(30),  
MaKho Varchar(30),  
Constraint FK1_HH Foreign key (MaLoai) References LoaiHangHoa (MaLoai),  
Constraint FK2_HH Foreign key (MaKho) References Kho(MaKho));
```

```
Create table HopDong -- Tạo bảng Hợp Đồng  
(MaHopDong Varchar(30) primary key,  
NoiDung Nvarchar(200),  
NgayGD SmallDateTime,  
TongTien Numeric(20,4),  
NgayTrienKhai SmallDateTime,  
MaHH Varchar(30),  
MaNV Varchar(30),  
MaKH Varchar(30),  
Constraint FK1_HopDong Foreign key (MaHH) References HangHoa (MaHH),  
Constraint FK2_HopDong Foreign key (MaNV) References NhanVien (MaNV),  
Constraint FK3_HopDong Foreign key (MaKH) References KhachHang (MaKH));
```

```
Create table HoaDon -- Tạo bảng Hóa Đơn  
(MaHoaDon Varchar(30) primary key,  
NoiDung Nvarchar(200),  
NgayGD SmallDateTime,  
TongTien Numeric(20,4),  
DaThanhToan Numeric(20,4),  
ConLai Numeric(20,4),  
TrangThai Nvarchar(30),  
MaHopDong Varchar(30),  
Constraint FK_HoaDon Foreign key (MaHopDong) References HopDong (MaHopDong));
```

```

Create table PhieuNo -- Tạo bảng Phiếu Nợ
(MaPhieuNo Varchar(30) primary key,
 NoiDung Nvarchar(200),
 NgayGD SmallDateTime,
 SoTienNo Numeric(20,4),
 HanThanhToan SmallDateTime,
 TrangThai Nvarchar(30),
 MaHopDong Varchar(30),
 Constraint FK_PhieuNo Foreign key (MaHopDong) References HopDong (MaHopDong));

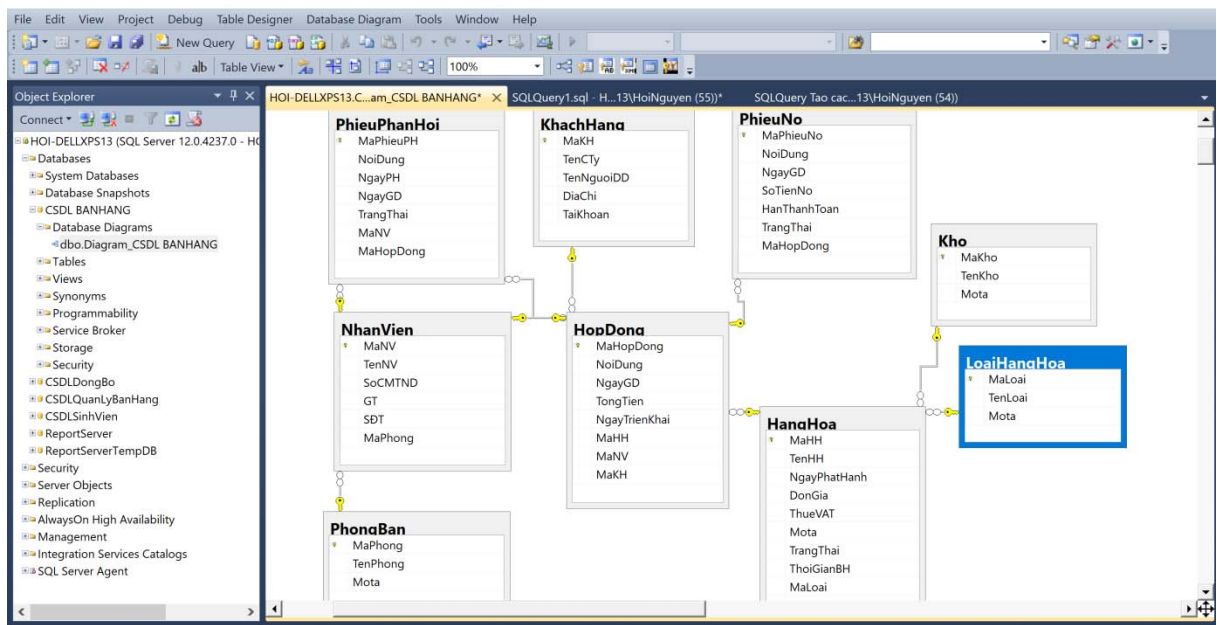
```

```

Create table PhieuPhanHoi -- Tạo bảng Phiếu Phản Hồi
(MaPhieuPH Varchar(30) primary key,
 NoiDung Nvarchar(200),
 NgayPH SmallDateTime,
 NgayGD SmallDateTime,
 TrangThai Nvarchar(30),
 MaNV Varchar(30),
 MaHopDong Varchar(30),
 Constraint FK1_PhieuPH Foreign key (MaHopDong) References HopDong (MaHopDong),
 Constraint FK2_PhieuPH Foreign key (MaNV) References NHanVien (MaNV));

```

4. Kết nối giữa các bảng



5. Chèn dữ liệu

```

--NHẬP DỮ LIỆU CHO BẢNG PHONGBAN
Insert into PhongBan Values ('CSD', N'Phòng kinh doanh và phát triển thị trường', N'Chuyên làm việc với khách hàng');
Insert into PhongBan Values ('CEB', N'Phòng nghiệp vụ', N'Chuyên làm việc về bên mảng xây dựng sản phẩm/dịch vụ');
Insert into PhongBan Values ('CAD', N'Phòng quản trị hành chính', N'Là bộ phận quản lý mọi việc của Công ty');
Insert into PhongBan Values ('CFD', N'Phòng tài chính kế toán', N'Chuyên làm việc về tài chính của Công ty');
Insert into PhongBan Values ('CSB', N'Phòng giải pháp doanh nghiệp', N'Chuyên làm việc với khách hàng về vấn đề tư vấn, hỗ trợ');

```

```
Insert into PhongBan Values ('CSC', N'Phòng tư vấn và hỗ trợ khách hàng', N'Chuyên làm việc với khách hàng');
Insert into PhongBan Values ('CRD', N'Phòng nghiên cứu và phát triển sản phẩm', N'Chuyên nghiên cứu nhằm tạo ra các sản phẩm nổi bật');
```

--NHẬP DỮ LIỆU CHO BẢNG NHANVIEN

```
Insert Into NhanVien Values ('NV01', N'Nguyễn Thị Thu Hằng', 025671093, 0, 01653269915, 'CSD');
Insert Into NhanVien Values ('NV02', N'Nguyễn Thị Hằng', 225671069, 0, 01653269918, 'CSD');
Insert Into NhanVien Values ('NV03', N'Nguyễn Thị Trang', 025671063, 0, 0953269915, 'CSD');
Insert Into NhanVien Values ('NV04', N'Nguyễn Văn Nam', 126871063, 1, 0989719365, 'CEB');
Insert Into NhanVien Values ('NV05', N'Lại Thế Quyết', 125601093, 1, 01653469987, 'CEB');
Insert Into NhanVien Values ('NV06', N'Nguyễn Văn Cường', 125671093, 1, 01629661467, 'CEB');
Insert Into NhanVien Values ('NV07', N'Trần Văn Thiện', 025671063, 1, 01657896654, 'CSC');
Insert Into NhanVien Values ('NV08', N'Ngô Duy Linh', 325671893, 1, 09893269915, 'CSB');
Insert Into NhanVien Values ('NV09', N'Nguyễn Cao Cường', 125671063, 1, 0989714365, 'CSD');
Insert Into NhanVien Values ('NV10', N'Phùng Thị Hằng', 119671063, 0, 01653269897, 'CSD');
```

--NHẬP DỮ LIỆU CHO BẢNG KHACHHANG

```
Insert Into KhachHang Values ('KH01', N'Công ty A', N'Trần Văn Mạnh', N'Hoài Đức - Hà Nội', 11246798001);
Insert Into KhachHang Values ('KH02', N'Công ty B', N'Trần Văn Đức', N'Mỹ Đức - Hà Nội', 23246798091);
Insert Into KhachHang Values ('KH03', N'Công ty C', N'Trần Hữu Hải', N'Ba Đình - Hà Nội', 01234567890);
Insert Into KhachHang Values ('KH04', N'Công ty D', N'Đình Mạnh Ninh', N'Thanh Oai - Hà Nội', 32249798098);
Insert Into KhachHang Values ('KH05', N'Công ty E', N'Nguyễn Hữu Hải', N'Quế Võ - Bắc Ninh', 32246798821);
Insert Into KhachHang Values ('KH06', N'Công ty F', N'Nguyễn Mai Lan', N'Yên Phong - Bắc Ninh', 22246788823);
Insert Into KhachHang Values ('KH07', N'Công ty G', N'Trần Văn Hải', N'Phủ Lý - Hà Nam', 02246768822);
Insert Into KhachHang Values ('KH08', N'Công ty H', N'Nguyễn Mạnh Nam', N'Quế Võ - Bắc Ninh', 32246798821);
Insert Into KhachHang Values ('KH09', N'Công ty I', N'Nguyễn Trung Anh', N'Quế Võ - Bắc Ninh', 92246798822);
Insert Into KhachHang Values ('KH10', N'Công ty K', N'Nguyễn Mạnh Cường', N'Hoàng Hóa - Thanh Hóa', 42247798821);
Insert Into KhachHang Values ('KH11', N'Công ty L', N'Nguyễn Cao Cường', N'Hoài Đức - Bắc Ninh', 62246798521);
Insert Into KhachHang Values ('KH12', N'Công ty M', N'Trần Hữu Hải', N'Quế Võ - Bắc Ninh', 92246798021);
Insert Into KhachHang Values ('KH13', N'Công ty N', N'Nguyễn Quốc Tường', N'Thanh oai - Hà Nội', 322467012329);
Insert Into KhachHang Values ('KH14', N'Công ty O', N'Nguyễn Văn Quân', N'Thanh oai - Hà Nội', 92246790131);
Insert Into KhachHang Values ('KH15', N'Công ty Q', N'Nguyễn Quốc Dân', N'Ba Đình - Bắc Ninh', 45646798821);
```

--NHẬP DỮ LIỆU CHO BẢNG KHO

```
Insert Into Kho Values ('K01', N'Kho phần mềm kế toán', N'Chứa phần mềm kế toán');
Insert Into Kho Values ('K02', N'Kho phần mềm hàng tồn kho', N'Chứa phần mềm hàng tồn kho');
Insert Into Kho Values ('K03', N'Kho phần mềm bán hàng', N'Chứa phần mềm bán hàng');
Insert Into Kho Values ('K04', N'Kho phần mềm quản lý nguồn nhân lực', N'Chứa phần mềm quản lý nguồn nhân lực');
```

```

Insert Into Kho Values ('K05', N'Kho phần mềm mua hàng', N'Chứa phần mềm mua hàng');
Insert Into Kho Values ('K06', N'Kho phần mềm quản trị doanh nghiệp ERP', N'Chứa phần mềm
quản trị doanh nghiệp ERP');
Insert Into Kho Values ('K07', N'Kho dịch vụ tư vấn quy trình', N'');
Insert Into Kho Values ('K08', N'Kho dịch vụ tư vấn mã hóa', N'');
Insert Into Kho Values ('K09', N'Kho dịch vụ triển khai', N'');
Insert Into Kho Values ('K10', N'Kho dịch vụ bảo hành', N'');
Insert Into Kho Values ('K11', N'Kho dịch vụ bảo trì', N'');

```

--NHẬP DỮ LIỆU CHO BẢNG LOAIHANGHOA

```

Insert Into LoaiHangHoa Values ('1', N'Sản phẩm', N'');
Insert Into LoaiHangHoa Values ('0', N'Dịch Vụ', N'');

```

--NHẬP DỮ LIỆU CHO BẢNG HANGHOA

```

Insert Into HangHoa Values ('HH01', N'Phần mềm kế toán 2004', '01-12-2003', 500.000,
50.000, N' ', N'Còn', N'1 năm', '1', 'K01');
Insert Into HangHoa Values ('HH02', N'Phần mềm quản lý hàng tồn kho', '01-07-2012',
600.000, 60.000, N' ', N'Còn', N'1 năm', '1', 'K02');
Insert Into HangHoa Values ('HH03', N'Phần mềm quản lý bán hàng', '01-07-2012', 500.000,
50.000, N' ', N'Còn', N'1 năm', '1', 'K03');
Insert Into HangHoa Values ('HH04', N'Phần mềm quản lý nguồn nhân lực', '01-12-2013',
500.000, 50.000, N' ', N'Còn', N'1 năm', '1', 'K04');
Insert Into HangHoa Values ('HH05', N'Phần mềm ERP 7.0', '01-12-2012', 2000.000, 200.000,
N' ', N'Còn', N'2 năm', '1', 'K06');
Insert Into HangHoa Values ('HH06', N'Phần mềm ERP 8.0', '01-12-2003', 3000.000, 300.000,
N' ', N'Còn', N'2 năm', '1', 'K06');
Insert Into HangHoa Values ('HH07', N'Dịch vụ tư vấn quy trình', '01-01-2004', 400.000,
40.000, N' ', N'Còn', N'1 năm', '0', 'K07');
Insert Into HangHoa Values ('HH08', N'Dịch vụ tư vấn mã hóa', '01-12-2003', 500.000,
50.000, N' ', N'Còn', N'1 năm', '0', 'K08');
Insert Into HangHoa Values ('HH09', N'Dịch vụ triển khai', '01-12-2003', 500.000, 50.000,
N' ', N'Còn', N'1 năm', '0', 'K09');
Insert Into HangHoa Values ('HH10', N'Dịch vụ bảo hành', '01-12-2003', 500.000, 50.000, N'
', N'Còn', N'1 năm', '0', 'K10');
Insert Into HangHoa Values ('HH11', N'Dịch vụ bảo trì', '01-12-2003', 500.000, 50.000, N'
', N'Còn', N'1 năm', '0', 'K10');

```

--NHẬP DỮ LIỆU CHO BẢNG HOPDONG

```

Insert Into HopDong Values ('HopDong01', N'', '01-20-2016', 600.000, '02-01-2016', 'HH01',
'NV01', 'KH01');
Insert Into HopDong Values ('HopDong02', N'', '01-26-2016', 670.000, '02-01-2016', 'HH02',
'NV01', 'KH02');
Insert Into HopDong Values ('HopDong03', N'', '01-22-2017', 600.000, '02-01-2017',
'HH01', 'NV01', 'KH09');
Insert Into HopDong Values ('HopDong04', N'', '01-24-2017', 2600.000, '02-05-2017',
'HH05', 'NV01', 'KH04');
Insert Into HopDong Values ('HopDong05', N'', '02-20-2017', 3900.000, '03-01-2017',
'HH06', 'NV01', 'KH11');
Insert Into HopDong Values ('HopDong06', N'', '03-10-2017', 600.000, '03-20-2017',
'HH04', 'NV01', 'KH03');
Insert Into HopDong Values ('HopDong07', N'', '04-15-2017', 3600.000, '04-30-2017',
'HH06', 'NV01', 'KH08');
Insert Into HopDong Values ('HopDong08', N'', '01-19-2018', 600.000, '02-01-2018',
'HH09', 'NV01', 'KH07');
Insert Into HopDong Values ('HopDong08', N'', '02-20-2018', 600.000, '03-01-2018',
'HH10', 'NV01', 'KH05');
Insert Into HopDong Values ('HopDong10', N'', '04-20-2018', 600.000, '04-30-2018',
'HH01', 'NV01', 'KH09');

```

--NHẬP DỮ LIỆU CHO BẢNG HOADON

```

Insert Into HoaDon Values ('HD01', N'', '01-20-2016', 600.000, 600.000, 0, N'Đã thanh toán
đủ', 'HopDong01');
Insert Into HoaDon Values ('HD02', N'', '01-26-2016', 670.000, 500.000, 160.000, N'Chưa
thanh toán đủ', 'HopDong02');
Insert Into HoaDon Values ('HD03', N'', '01-22-2017', 600.000, 500.000, 100.000, N'Chưa
thanh toán đủ', 'HopDong03');
Insert Into HoaDon Values ('HD04', N'', '01-24-2017', 2600.000, 2600.000, 0, N'Đã thanh
toán đủ', 'HopDong04');
Insert Into HoaDon Values ('HD05', N'', '02-20-2017', 3900.000, 3500.000, 400.000, N'Chưa
thanh toán đủ', 'HopDong05');
Insert Into HoaDon Values ('HD06', N'', '03-10-2017', 600.000, 600.000, 0, N'Đã thanh toán
đủ', 'HopDong06');
Insert Into HoaDon Values ('HD07', N'', '04-15-2017', 3600.000, 3000.000, 600.000, N'Chưa
thanh toán đủ', 'HopDong07');
Insert Into HoaDon Values ('HD08', N'', '01-19-2018', 600.000, 600.000, 0, N'Đã thanh toán
đủ', 'HopDong08');
Insert Into HoaDon Values ('HD09', N'', '02-20-2018', 600.000, 500.000, 100.000, N'Chưa
thanh toán đủ', 'HopDong09');
Insert Into HoaDon Values ('HD10', N'', '04-20-2018', 600.000, 500.000, 100.000, N'Chưa
thanh toán đủ', 'HopDong10');

```

--NHẬP DỮ LIỆU CHO BẢNG PHIEUNO

```

Insert Into PhieuNo Values ('PN01', N'', '01-26-2016', 160.000, '02-26-2016', N'Chưa thanh
toán đủ', 'HopDong02');
Insert Into PhieuNo Values ('PN02', N'', '01-22-2017', 100.000, '02-22-2017', N'Chưa thanh
toán đủ', 'HopDong03');
Insert Into PhieuNo Values ('PN03', N'', '02-20-2017', 400.000, '03-20-2017', N'Chưa thanh
toán đủ', 'HopDong05');
Insert Into PhieuNo Values ('PN04', N'', '04-15-2017', 600.000, '05-15-2017', N'Chưa thanh
toán đủ', 'HopDong07');
Insert Into PhieuNo Values ('PN05', N'', '02-20-2018', 100.000, '03-20-2018', N'Chưa thanh
toán đủ', 'HopDong09');
Insert Into PhieuNo Values ('PN06', N'', '04-20-2018', 100.000, '05-20-2018', N'Chưa thanh
toán đủ', 'HopDong10');

```

--NHẬP DỮ LIỆU CHO BẢNG PHIEUPHANHOI

```

Insert Into PhieuPhanHoi Values ('PH01', N'phần mềm bị lỗi kết nối với dữ liệu', '02-01-
2016', '03-01-2016', N'Đã giải quyết xong', 'NV01', 'HopDong01');
Insert Into PhieuPhanHoi Values ('PH02', N'phần mềm bị lỗi kết nối với máy chủ', '02-01-
2016', '03-01-2016', N'Đã giải quyết xong', 'NV05', 'HopDong02');
Insert Into PhieuPhanHoi Values ('PH03', N'phần mềm bị lỗi kết nối với dữ liệu', '04-15-
2017', '05-15-2017', N'Đã giải quyết xong', 'NV03', 'HopDong03');
Insert Into PhieuPhanHoi Values ('PH04', N'phần mềm bị lỗi kết nối với dữ liệu', '07-22-
2017', '08-22-2017', N'Đã giải quyết xong', 'NV01', 'HopDong04');
Insert Into PhieuPhanHoi Values ('PH05', N'phần mềm bị lỗi kết nối với dữ liệu', '09-20-
2017', '10-20-2017', N'Đã giải quyết xong', 'NV07', 'HopDong05');

```

6. Xem dữ liệu

```

Select * from HangHoa
Select * from HoaDon
Select * from HopDong
Select * from KhachHang
Select * from Kho
Select * from LoiHangHoa
Select * from NhanVien
Select * from PhieuNo
Select * from PhieuPhanHoi
Select * from PhongBan

```

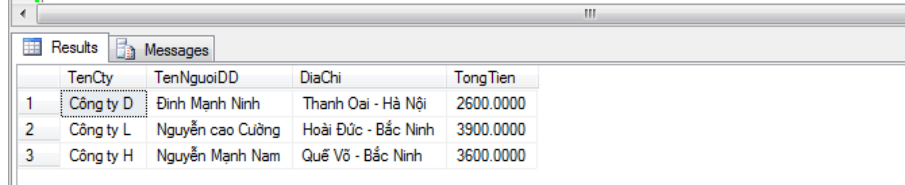
7. Truy vấn

Ví dụ 1: Hiển thị những khách hàng có giá trị hợp đồng lớn hơn 1 tỷ VNĐ?

Câu lệnh truy xuất được viết như sau:

Thực hiện trên Hệ quản trị CSDL SQL Server như sau:

```
-- Ví dụ 1: Hiển thị những khách hàng có giá trị hợp đồng lớn hơn 1 tỷ VNĐ?  
Select TenCty, TenNguoiDD, DiaChi, TongTien  
From HopDong a Inner join KháchHang b On a.MaKH = b.MaKH  
Where a.TongTien > 1000.000;
```



TenCty	TenNguoiDD	DiaChi	TongTien
1 Công ty D	Đình Mạnh Ninh	Thanh Oai - Hà Nội	2600.0000
2 Công ty L	Nguyễn cao Cường	Hoài Đức - Bắc Ninh	3900.0000
3 Công ty H	Nguyễn Mạnh Nam	Quế Võ - Bắc Ninh	3600.0000

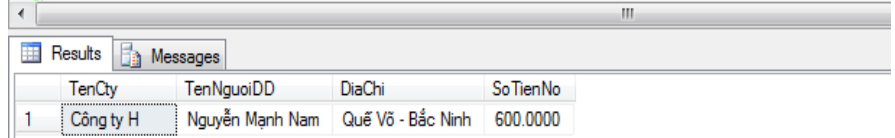
Kết quả truy vấn ví dụ 1 (Nguồn: SQL Server 2014).

Ví dụ 2: Hiển thị những khách hàng còn nợ với số tiền ≥ 500.000 của những hợp đồng được ký kết năm 2017?

Câu lệnh truy xuất được viết như sau:

Thực hiện trên Hệ quản trị CSDL SQL Server như sau:

```
-- Ví dụ 2: Hiển thị những khách hàng còn nợ với số tiền  $\geq 500.000$  của  
Select TenCty, TenNguoiDD, DiaChi, SoTienNo  
From KháchHang a Inner join HopDong b On a.MaKH = b.MaKH  
Inner join PhieuNo c On b.MaHopDong = c.MaHopDong  
Where (YEAR(b.NgayGD) = 2017) AND (c.SoTienNo  $\geq$  500.000);
```



TenCty	TenNguoiDD	DiaChi	SoTienNo
1 Công ty H	Nguyễn Mạnh Nam	Quế Võ - Bắc Ninh	600.0000

CÁC THAO TÁC QUẢN TRỊ TRONG SQL SERVER

Bài toán 1: Bổ sung các ràng buộc cho bảng

Bổ sung thêm các ràng buộc về khóa cho các bảng dữ liệu.

```

ALTER TABLE [dbo].[ChiTietHoaDonMua] ADD PRIMARY KEY ([MaHDM], [MaSP])
GO

-----
-- Tạo khóa chính cho bảng Hình Thức Thanh Toán
-----
ALTER TABLE [dbo].[HinhThucThanhToan] ADD PRIMARY KEY ([MaHTTT])
GO

-----
-- Tạo khóa ngoài cho bảng Chi tiết hóa đơn bán
-----
ALTER TABLE [dbo].[ChiTietHoaDonBan] ADD FOREIGN KEY ([MaHDB]) REFERENCES [dbo].[HoaDonBan] ([MaHDB]) ON DELETE NO
GO
ALTER TABLE [dbo].[ChiTietHoaDonBan] ADD FOREIGN KEY ([MaSP]) REFERENCES [dbo].[SanPham] ([MaSP]) ON DELETE NO
GO
ALTER TABLE [dbo].[ChiTietHoaDonBan] ADD FOREIGN KEY ([MaHTTT]) REFERENCES [dbo].[HinhThucThanhToan] ([MaHTTT])
GO

-----
-- Tạo khóa ngoài cho bảng Chi tiết hóa đơn mua
-----
ALTER TABLE [dbo].[ChiTietHoaDonMua] ADD FOREIGN KEY ([MaHDM]) REFERENCES [dbo].[HoaDonMua] ([MaHDM]) ON DELETE NO
GO
ALTER TABLE [dbo].[ChiTietHoaDonMua] ADD FOREIGN KEY ([MaSP]) REFERENCES [dbo].[SanPham] ([MaSP]) ON DELETE NO
GO

```

Các ràng buộc về khóa cho các bảng dữ liệu

Bài toán 2: Tạo View

View là một Virtual Table (Table ảo), nó không giống với table thông thường, mà nó chứa các columns và dữ liệu của các Table khác nhau. Views không chỉ chứa data mà còn là một tập Queries như table thông thường.

Ví dụ: Tạo khung nhìn danh sách sản phẩm đã bán trên 10.000.000 đ

```

-- TẠO VIEW--
--1. Hiển thị bảng có hóa đơn trên 10.000.000 đ
CREATE VIEW vw_Hanghoa
as
SELECT SanPham.TenSP, ChiTietHoaDonBan.Soluong, ChiTietHoaDonBan.Dongia, ChiTietHoaDonBan.Thanhtien
FROM SanPham inner join ChiTietHoaDonBan on ChiTietHoaDonBan.MaSP=SanPham.MaSP
WHERE ChiTietHoaDonBan.Thanhtien > 10000000
GO
SELECT * FROM vw_Hanghoa
GO

```

	TenSP	Soluong	Dongia	Thanhtien
1	Phần mềm quản lý bán sách	5	3250000	16250000
2	Phần mềm quản lý đơn thư khiêu nại, tố cáo	6	4570000	27420000

View “Danh sách sản phẩm đã bán trên 10.000.000đ”

- Tạo khung nhìn “Danh sách nhân viên được tác động vào hệ thống và quyền hạn của họ”.


```
--2. Hiển thị danh sách nhân viên được tác động vào hệ thống và quyền hạn của họ
CREATE VIEW vw_Quyensudung1
AS
SELECT * FROM NhanVien
WHERE NhanVien.Trangthai='1'
GO
SELECT * FROM vw_Quyensudung1
GO
```

ID	MaNV	TenNV	Diachi	SDT	Taikhoan	Matkhau	Trangthai	Phanquyen
1	1	NV01	Admin	Hà Nội	1679610685	admin	admin	1
2	2	NV02	Hoàng Thị Phương Châm	Thanh Hóa	1262156462	phuongcham	123456	1
3	3	NV03	Phạm Mai Phương	Bắc Ninh	1662117126	maiphuongc	123456	1

View “Danh sách nhân viên được tác động vào hệ thống và quyền hạn của họ”.

Bài toán 3: Tạo các thủ tục Procedure

Stored Procedure trong SQL Server bao gồm nhiều câu lệnh SQL mà bạn lưu lại cho những lần sử dụng sau (sử dụng nhiều lần). Vì vậy, nếu bạn nghĩ về một truy vấn mà bạn viết lặp đi lặp lại, thay vì phải viết truy vấn đó bạn nên lưu lại như một stored procedure (thủ tục) và sau đó chỉ cần gọi stored procedure để thực thi câu lệnh SQL.

Cú pháp

CREATE PROCEDURE tên_sp([@parameter_name parameter_data_type [= default] [,...n]])

AS

câu_lệnh_sql;

Ví dụ: Tạo stored procedure để thêm dữ liệu vào bảng NHANVIEN với các tham số là các dữ liệu cần thêm vào các cột.

- Tạo thủ tục

```
CREATE PROCEDURE sp_ThemNhanVien (@MaNv char(15), @TenNv nvarchar(50), @GioiTinh nvarchar(4), @TenPhong nvarchar(30))
```

AS

Begin

```
INSERT INTO NHANVIEN
```

```
VALUES(@MaNv,@TenNv,@GioiTinh,@TenPhong)
```

```
End;
```

- Chạy thủ tục sp_ThemNhanVien

```
exec sp_ThemNhanVien 'nv05', N'Lê Tài', N'Nam', N'Phòng IT'
```

```
-- Hủy procedure Get_Employee_Infos nếu nó đã tồn tại.
```

```
-- (Để cho phép tạo lại)
```

```
IF OBJECT_ID(N'dbo.Get_Employee_Infos', N'P') IS NOT NULL
```

```

DROP PROCEDURE Get_Employee_Infos;
GO

-- Thủ tục truyền vào p_Emp_Id
-- Và trả về v_First_Name, v_Last_Name, v_Dept_Id.
CREATE PROCEDURE Get_Employee_Infos (@p_Emp_Id integer
    , @v_First_Name varchar(50) OUTPUT
    , @v_Last_Name varchar(50) OUTPUT
    , @v_Dept_Id integer OUTPUT)
AS
BEGIN
-- Sử dụng lệnh Print để in ra một chuỗi (Dành cho lập trình viên).
-- Sử dụng Cast để ép kiểu số Integer về dạng chuỗi (Varchar).
-- Sử dụng toán tử + để nối hai chuỗi.
PRINT 'Parameter @p_Emp_Id = ' + CAST(@p_Emp_Id AS varchar(15));
--
-- Query dữ liệu từ bảng và gán giá trị vào các biến.
--
SELECT
    @v_First_Name = Emp.First_Name,
    @v_Last_Name = Emp.Last_Name,
    @v_Dept_Id = Emp.Dept_Id
FROM Employee Emp
WHERE Emp.Emp_Id = @p_Emp_Id;

--
-- Log dành cho người lập trình
--
PRINT 'Found Record!';
PRINT ' @v_First_Name= ' + @v_First_Name;
PRINT ' @v_Last_Name= ' + @v_Last_Name;
PRINT ' @v_Dept_Id= ' + CAST(@v_Dept_Id AS varchar(15));

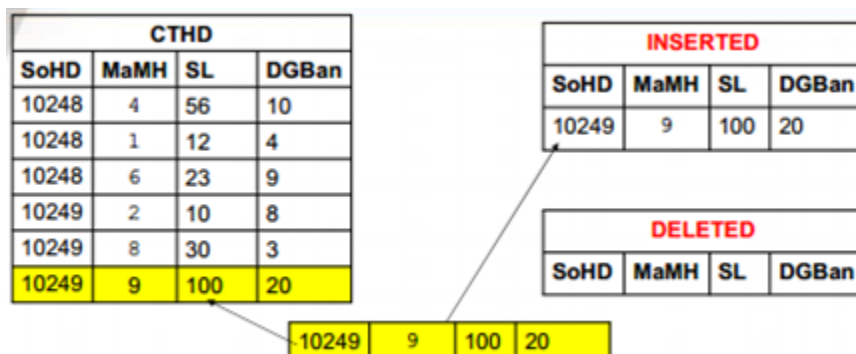
END;

```

Bài toán 4: Tạo Trigger

Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ (nhiều bảng/table) hoặc trên nhiều dòng (nhiều record) của bảng.

Ví dụ một chi tiết hoá đơn không được nhiều hơn 10 hoá đơn (tức là trong bảng CTHD không được vượt quá 10 SoHD)



Trigger trong sql server 2014 – Tạo trigger

Cú pháp tạo trigger trong sql server

CREATE TRIGGER tên_trigger ON tên_bảng

FOR {DELETE, INSERT, UPDATE}

AS

câu_lệnh_sql

Bên dưới là một vài lưu ý

Khi trigger được thực hiện, SQL tự động tạo ra 2 bảng tạm với cùng cấu trúc với bảng chứa trigger.

INSERTED chứa dữ liệu mới khi chúng ta thực thi câu lệnh Insert hoặc câu lệnh Update.

DELETED chứa những bản ghi bị xoá khi chúng ta thực thi câu lệnh Delete hoặc chứa dữ liệu cũ khi chúng ta thực thi câu lệnh Update.

Trigger trong sql server 2014 – Khi thêm dữ liệu

Tình huống: Kiểm tra ràng buộc một CTHD không chứa nhiều hơn 10 SoHD.

Trong hình, chúng tôi đang chọn SoHD là 10249. Hiện tại trong bảng CTHD trước khi thêm đang có 2. Dòng màu vàng là dữ liệu chúng ta muốn thêm vào.

CTHD			
SoHD	MaMH	SL	DGBan
10248	4	56	10
10248	1	12	4
10248	6	23	9
10249	2	10	8
10249	8	30	3
10249	9	100	20

INSERTED			
SoHD	MaMH	SL	DGBan
10249	9	100	20

DELETED			
SoHD	MaMH	SL	DGBan

10249	9	100	20
-------	---	-----	----

Câu lệnh tạo trigger trong sql server

Create Trigger tr_SoCTHD On CTHD For Insert

As

If (Select Count(a.SoHD)

From CTHD a Inner Join INSERTED b On a.SoHD = b.SoHD) > 10

Begin

Print 'So CTHD Khong the > 10'

RollBack Tran

End

Inner Join là phép kết giữa 2 bảng. Lệnh Print dùng để hiển thị thông báo đến người dùng. Lệnh RollBack Tran dùng để ngăn chặn xử lý thêm mới dữ liệu.

Trigger trong sql server 2014 – Khi xoá dữ liệu

Tình huống: Khi xoá một dòng trong bảng CTHD, chúng ta phải tăng số lượng của bảng MATHANG

CTHD			
SoHD	MaMH	SL	DGBan
10248	4	56	10
10248	1	12	4
10248	6	23	9
10249	2	10	8
10249	8	30	3

INSERTED			
SoHD	MaMH	SL	DGBan

DELETED			
SoHD	MaMH	SL	DGBan
10249	9	100	20

10249	9	100	20
-------	---	-----	----

Câu lệnh tạo trigger trong sql server

Create Trigger tr_XoaCTHD On CTHD

For Delete

As

Update MATHANG Set SoTon = SoTon + SL From DELETED

Where MATHANG.MaMH = DELETED.MaMH

Trigger trong sql server 2014 – Khi cập nhật dữ liệu

Tình huống: Khi thay đổi cột SL trên bảng CTHD, chúng ta phải cập nhật lại dữ liệu cho cột SoTon trong bảng MATHANG

CTHD			
SoHD	MaMH	SL	DGBan
10248	4	56	10
10248	1	12	4
10248	6	23	9
10249	2	10	8
10249	8	30	3
10249	9	100	20

INSERTED			
SoHD	MaMH	SL	DGBan
10249	9	10	20

DELETED			
SoHD	MaMH	SL	DGBan
10249	9	100	20

10

Câu lệnh tạo trigger

Create Trigger tr_SuaCTHD On CTHD

For Update As

Declare @D int

Select @D = Count(*)

From MatHang a, DELETED b, INSERTED c

Where a.MaMH = b.MaMH

And a.MaMH = c.MaMH

And SoTon + b.SL - c.SL < 0

If (@D > 0)

Begin

Print 'Không đủ hàng để bán'

RollBack Tran

Return

End

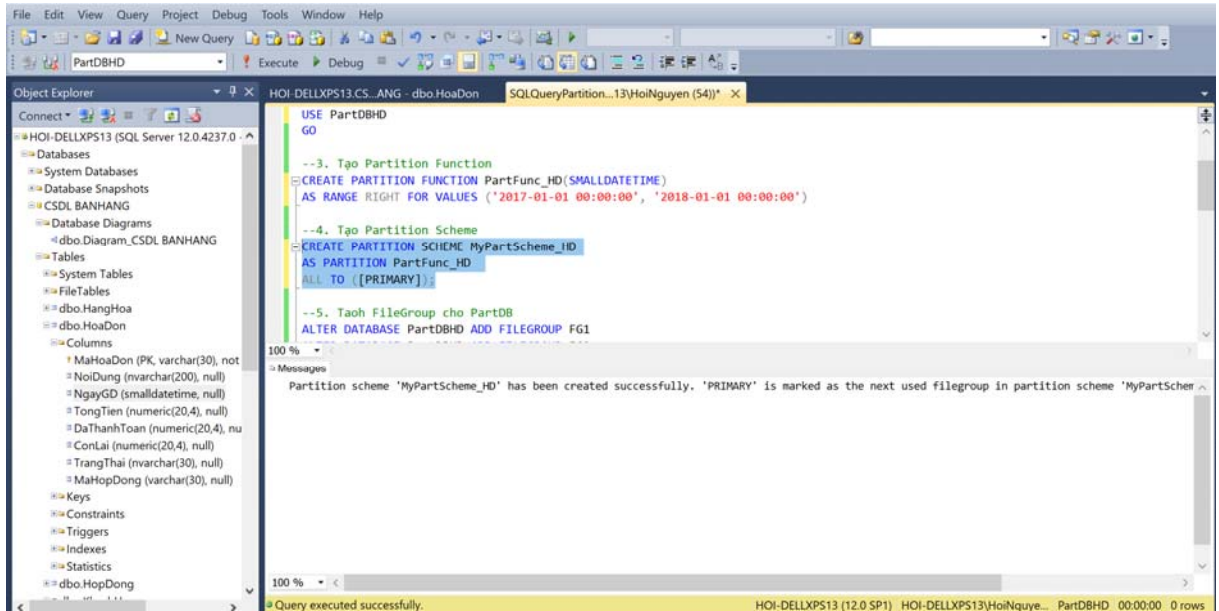
--Cập nhật số lượng tồn trong bảng MATHANG

Update MatHang Set SoTon = SoTon + b.SL - c.SL

From MatHang a, DELETED b, INSERTED c

Where a.MaMH = b.MaMH And a.MaMH = c.MaMH

Bài toán 5: Phân hoạch bảng



```
USE PartDBHD
GO

--3. Tạo Partition Function
CREATE PARTITION FUNCTION PartFunc_HD(SMALLDATETIME)
AS RANGE RIGHT FOR VALUES ('2017-01-01 00:00:00', '2018-01-01 00:00:00')

--4. Tạo Partition Scheme
CREATE PARTITION SCHEME MyPartScheme_ID
AS PARTITION PartFunc_HD
ALL TO ([PRIMARY])

--5. Tạo FileGroup cho PartDB
ALTER DATABASE PartDBHD ADD FILEGROUP FG1
```

Partition scheme 'MyPartScheme_HD' has been created successfully. 'PRIMARY' is marked as the next used filegroup in partition scheme 'MyPartScheme_HD'

Query executed successfully.

Bài toán 6: Backup và Restore CSDL

Backup và Restore CSDL (Sao lưu và khôi phục dữ liệu) là vấn đề rất quan trọng khi quản trị CSDL, đảm bảo cho dữ liệu không bị mất mát và nếu có sự cố xảy ra thì có thể khôi phục lại trạng thái của CSDL trước khi xảy ra sự cố. Công việc này phải dựa vào cơ chế Backup và Restore của các hệ QTCSDL và để thực hiện tốt công việc này thì người quản trị dữ liệu phải có kịch bản sao lưu và khôi phục CSDL cho hệ thống một cách cẩn thận và chi tiết.

SQL Server cung cấp ba loại backup như sau:

Full Backup: Backup toàn bộ dữ liệu tại thời điểm thực hiện (thường dùng nhất).

Differential Backup: Backup các dữ liệu mới được cập nhật kể từ lần full backup trước đó.

Transaction Log Backup: Backup các log record hiện có trong log file. Nghĩa là sao lưu các hành động (các thao tác xảy ra đối với database) chứ không sao lưu dữ liệu. Đồng thời nó cũng cắt bỏ (truncate) log file, loại bỏ các log record vừa được backup ra khỏi log file. Vì thế khi thấy log file tăng quá lớn, có nhiều khả năng là bạn chưa từng backup transaction log bao giờ.

- Tạo Backup:

```

-- TẠO BACKUP--

USE DB_Sale_Management
GO
-- Tạo sao lưu full cho CSDL DB_Sale_Management.bak
-- Vào ổ C ta chọn đường dẫn: 'C:\Program File\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\DB_Sale_Management.bak'
BACKUP DATABASE DB_Sale_Management
TO DISK = 'C:\Program File\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\DB_Sale_Management.bak'
WITH NOINIT,
-- Không ghi đè file
NOFORMAT,
-- Không định dạng lại thiết bị sao lưu--> dữ liệu tồn tại không bị xóa
NAME =N'backup_DB_Sale_Management_1an1',
-- Tên bản sao lưu
SKIP, NOREWIND, NOUNLOAD
-- skip: bỏ qua sự phục hồi các bản backup hết hạn, tránh bị ghi đè dữ liệu, rewind: không lưu lại file backup
-- Số phần trăm thông báo thành công thể hiện trong quá trình sao lưu

```

Backup dữ liệu DB_Sale_Management

- Tạo Restore CSDL:

```

-- TẠO RESTORE--

USE DB_Sale_Management
GO
-- tạo phục hồi CSDL, từ file đã sao lưu trong ổ C
RESTORE DATABASE DB_Sale_Management
FROM DISK = 'C:\Program File\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\DB_Sale_Management.bak'
WITH FILE = 1,
--xác định số bản sao được phục hồi trong tập tin chứa các bản sao lưu
NOUNLOAD,
-- bản backup sẽ không được tự động load(tải) lại sau khi đã thực hiện thành công 1 backup
REPLACE,
-- xóa database DB_Sale_Management trong CSDL và thay thế bằng file backup được phục hồi stats=10
-- cứ sau 10% được phục hồi bạn sẽ được SQL Server thông báo

```

Restore dữ liệu DB_Sale_Management

- Kiểm tra lại sau khi Restore

```

*/
USE master
GO
IF EXISTS (SELECT * FROM sys.databases WHERE name = 'DB_Sale_Management')
DROP DATABASE DB_Sale_Management
-----
--TẠO DATABASE--
GO
CREATE DATABASE DB_Sale_Management
GO
USE DB_Sale_Management
-----
--TẠO BẢNG--
-----
-- Tạo bảng Khách Hàng
-----
GO
CREATE TABLE [dbo].[KhachHang] (
[ID] int NOT NULL IDENTITY(1,1) ,
[MaKH] nvarchar(10) NOT NULL ,
[TenKH] nvarchar(50) NOT NULL ,
[Địa chỉ] nvarchar(250) NOT NULL ,
[SĐT] int NOT NULL ,
[Email] nvarchar(250) NOT NULL
)

```

Cơ sở dữ liệu đã được phục hồi sau khi Restore

Bài toán 7: Đồng bộ hóa CSDL

Hầu hết các doanh nghiệp hiện nay đều sử dụng phần mềm chạy trên nền tảng Server – Client. Vì hiện tại phần mềm hoạt động trên cơ sở Server - Client nếu 1 ngày Server chính có trục trặc hoặc Database có vấn đề như bị dữ liệu bị hỏng thì hệ thống chúng ta sẽ ngừng hoạt động cho đến khi Server và Database được phục hồi như vậy sẽ gây không ít tổn thất cho công ty.

Vì vậy cần phải đưa ra giải pháp làm sao để khi Server hoặc Database có trục trặc thì hệ thống vẫn phải hoạt động bình thường không bị ảnh hưởng để công ty có thời gian phục hồi sự cố.

Bài toán 8: Tạo và phân quyền người dùng

- *Tạo tài khoản*

+ Tạo người đăng nhập

+ Tạo tài khoản cho người dùng

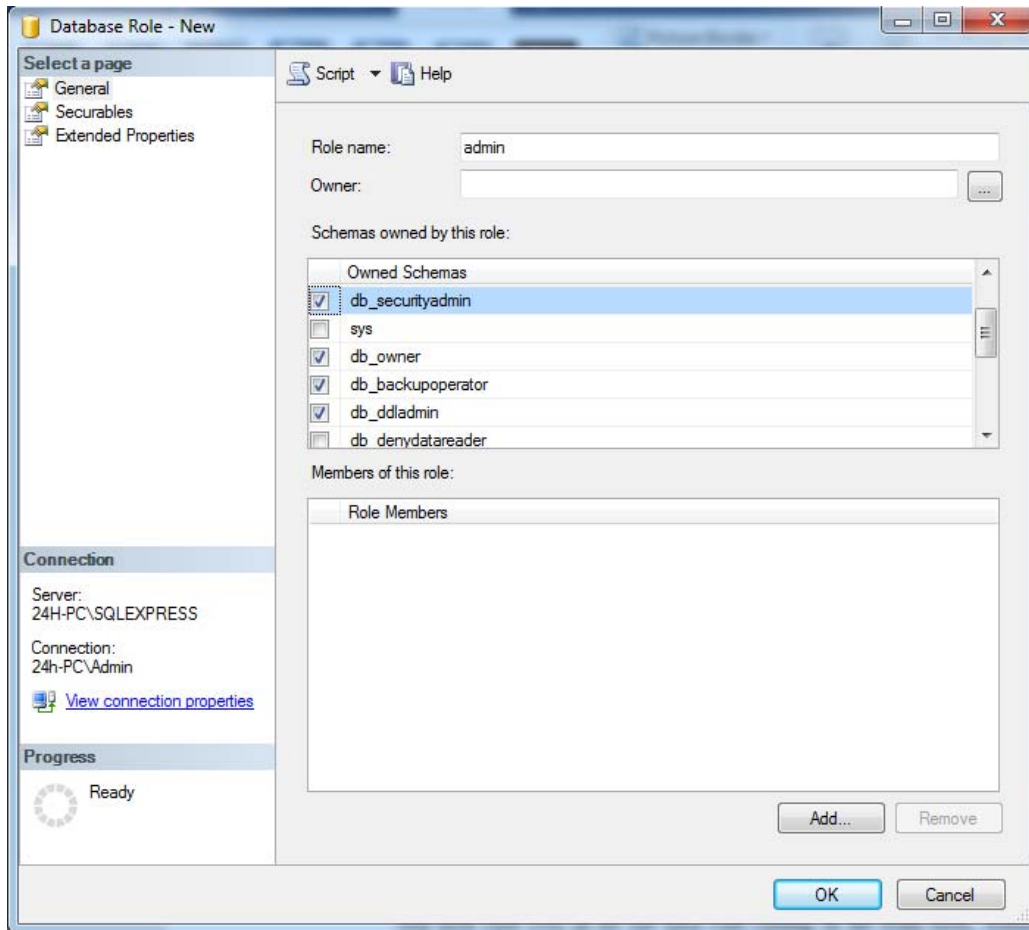

```

GO
--TẠO LOGIN--
CREATE LOGIN admin
WITH PASSWORD = 'admin',
DEFAULT_DATABASE= DB_Sale_Management
-- thực hiện tạo login trên database DB_Sale_Management
-----
--TẠO USER--
GO
CREATE USER phuongcham
FOR LOGIN ADMIN
-----
--PHÂN QUYỀN CHO LOGIN--
GO
GRANT all
ON Sanpham -- tên bảng muốn phân quyền
TO admin
-- all: dùng để cấp quyền insert, update, delete, select, references cho login có tên là: phuongcham
/* - nếu muốn cấp quyền tác động đến CSDL như create table, view, procedure, database
ta bỏ chữ all và thay vào các lệnh như: create table, create view, create procedure, create database.
hoặc nếu muốn cho user đó có quyền select trên một cột nào đó của bảng
chẳng hạn: cột ma_san_pham ta bỏ chữ all đi và viết: grant select on SanPham(MaSP) to admin,
nếu ko viết tên cột bên cạnh bảng đó thì được hiểu là có quyền tác động toàn bộ đến bảng đó */
GO

```

- Phân quyền người dùng

- + Quyền cho Admin: Người quản trị hệ thống có toàn quyền với hệ thống, được thực hiện mọi chức năng của phần mềm.
- + Quyền cho nhân viên kế toán: được xem tất cả các bảng và chỉ mình kế toán được thao tác việc nhập hóa đơn mua vào, lập hóa đơn chi tiết.
- + Quyền cho nhân viên kinh doanh: được thao tác với bảng nhà cung cấp, sản phẩm, loại sản phẩm và lập hóa đơn bán hàng cho khách hàng.
- + Quyền cho nhân viên kho: được thao tác với bảng nhà cung cấp, sản phẩm, loại sản phẩm.



Hình 3.32. Phân quyền cho nhóm quyền

Bài toán 9: Mã hóa dữ liệu

Mã hóa chủ yếu là để dữ liệu của chúng ta an toàn hơn, tránh sự soi mói tò mò của những kẻ không phận sự. Vì vậy, để đảm bảo tính bảo mật về tài khoản của từng nhân viên, thì cần phải mã hóa mật khẩu của nhân viên.

Phần mềm đã thực hiện mã hóa mật khẩu nhân viên qua <http://www.md5.cz/>. MD5 viết tắt của cụm từ Message-Digest algorithm 5 là một thuật toán mã hóa theo chuẩn RFC 1321 để tạo ra 1 chuỗi 128 bit từ 1 chuỗi dữ liệu bất kỳ. Mã MD5 có tác dụng kiểm tra tính toàn vẹn của tập tin khi tải về trên mạng hoặc copy giữa các thiết bị lưu trữ với nhau.

function md5()

Online generator [md5 hash of a string](#)

md5 ()

hash darling, hash!

md5 checksum:

21232f297a57a5a743894a0e4a801fc3

Implementations MD5:

[php manual function md5\(\)](#) | [md5 in JavaScript](#) | [md5 in MySql](#) | [md5 in MariaDB](#)

[MD5 on Wikipedia.org](#)

Mã hóa dữ liệu bằng MD5

	ID	MaNV	TenNV	Diachi	SDT	Taikhon	Matkhou	Trangthai	Phanquye
▶	1	NV01	Admin	Hà Nội	1679610685	admin	5a743894a0e4a801fc3	True	1
	2	NV02	Hoàng Thị Phương Chăm	Thanh Hóa	1262156462	phuongcham	e10adc3949ba59ab...	True	2
	3	NV03	Phạm Mai Phương	Bắc Ninh	1662117126	maiphuong	e10adc3949ba59ab...	True	3
	4	NV04	Cao Thị Thanh Thảo	Nam Định	169999666	thanhthao	e10adc3949ba59ab...	False	4
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Mật khẩu sau khi được mã hóa