

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

**PHẦN MỀM QUẢN LÝ TÀI KHOẢN
NGÂN HÀNG**

SINH VIÊN THỰC HIỆN: Nhóm 14

NGUYỄN MINH TIẾN - 12110193

LÂM KHÁNH TÀI - 12110168

GIÁO VIÊN HƯỚNG DẪN: NGUYỄN THÀNH SƠN

Tp. Hồ Chí Minh, ngày 22 tháng 11 năm 2014

QUẢN LÝ TÀI KHOẢN NGÂN HÀNG

I. **Phát biểu bài toán**

Ngày xưa các thương gia thường giao thương với nhau bằng những vật chất hay của cải, dần dần về sau thì đồng tiền bắt đầu ra đời từ đó nó bắt đầu làm vật phẩm trung gian cho giao thương. Đồng tiền được hình thành, các thương gia đó về sau có người là triệu phú còn có người là tỷ phú.... với số tiền lớn như vậy họ đã đem nó và gửi ngân hàng vì ngân hàng sau khi gửi họ vẫn có lời nhờ lãi suất. Vì vậy, ngày nay nhiều ngân hàng được ra đời với mục đích cho vay hay gửi tiền tiết kiệm.

II. **Mô tả chi tiết phần mềm**

Phần mềm quản lý tài khoản ngân hàng phục vụ cho quá trình quản lý ngân hàng. Giúp cho khách hàng dễ dàng quản lý tài khoản của riêng mình và tối ưu bảo mật thông tin của khách hàng. Quản lý bao gồm: Gửi tiền, rút tiền và chuyển khoản.

Quản lý gửi tiền: Khách hàng nếu đã có tài khoản trong ngân hàng thì sẽ trực tiếp vào mã tài khoản của khách hàng. Nếu khách hàng chưa có mã thì phải mang theo thẻ chứng minh thư và ngân hàng sẽ tạo tài khoản trực tiếp tại chi nhánh mà khách hàng đang sử dụng sau đó khách hàng có thể gửi tiền vào tài khoản của mình. Lưu ý: nếu khách hàng gửi tiền sẽ có kỳ hạn để ngân hàng tính và tăng lãi suất theo quý hoặc năm cho quý khách hàng.


Quản lý chuyển khoản: Đối với khách hàng muốn chuyển khoản từ tài khoản của mình qua tài khoản của người khác thì khi khách hàng vào chi nhánh vui lòng cầm theo thẻ chứng minh để ngân hàng xác nhận mã tài khoản sau đó khách hàng có quyền chuyển khoản vào tài khoản khác thuộc ngân hàng. Còn đối với khách hàng không có nhu cầu gửi trực tiếp từ tài khoản của mình hay chưa có tài khoản tại ngân hàng thì nhân viên chi nhánh sẽ phát phiếu khách hàng (phiếu chuyển khoản) và khách hàng có thể chuyển vào tài khoản mà mình muốn chuyển.

Quản lý rút tiền: Khách hàng muốn rút tiền phải mang theo chứng minh thư đến chi nhánh để ngân hàng xác nhận và sau đó khách hàng có quyền rút tiền từ trong tài khoản của mình.


III. Thiết kế hệ thống

1. Các bảng cơ sở dữ liệu

Bảng TAI_KHOAN (MaTK, TenTK, Tuoi, DchiKH, SDT, CMND, GioiTinh, NgayCap, SoDu, PIN, KiHan, LaiSuat)

	Column Name	Data Type
	MaTK	nchar(10)
	TenTK	ntext
	Tuoi	int
	DChiKH	ntext
	SDT	int
	CMND	int
	GioiTinh	nchar(10)
	NgayCap	datetime
	SoDu	money
	PIN	nchar(10)
	KiHan	int
	LaiSuat	real

Bảng PHIEU_KH (MaPhieuKH, MaGD, MaTKNhan, TenKH, Dchi, SDT, SoTien)

	Column Name	Data Type
	MaPhieuKH	nchar(10)
	MaGD	nchar(10)
	MaTKNhan	nchar(10)
	TenKH	ntext
	DChi	ntext
	SDT	int
	SoTien	money

Bảng PHIEU_GD (MaPhieuGD, MaTK, MaNV, MaGD, TenGD, SoTien, ThoiGian)

	Column Name	Data Type
▶ 🔑	MaPhieuGD	nchar(10)
	MaTK	nchar(10)
	MaNV	nchar(10)
	MaGD	nchar(10)
	TenGD	ntext
	SoTien	money
	ThoiGian	datetime

Bảng NHANH_VIEN (MaNV, MaCN, TenNV, Tuoi, GioiTinh, DiaChi, Luong, MaQli, MatKhau)

	Column Name	Data Type
▶ 🔑	MaNV	nchar(10)
	MaCN	nchar(10)
	TenNV	ntext
	Tuoi	int
	GioiTinh	nchar(10)
	DiaChi	ntext
	Luong	money
	MaQli	nchar(10)
	MatKhau	nchar(10)

Bảng LOAI_GD (LoaiGD, TenGD)

	Column Name	Data Type
▶ 🔑	LoaiGD	nchar(10)
	TenGD	ntext

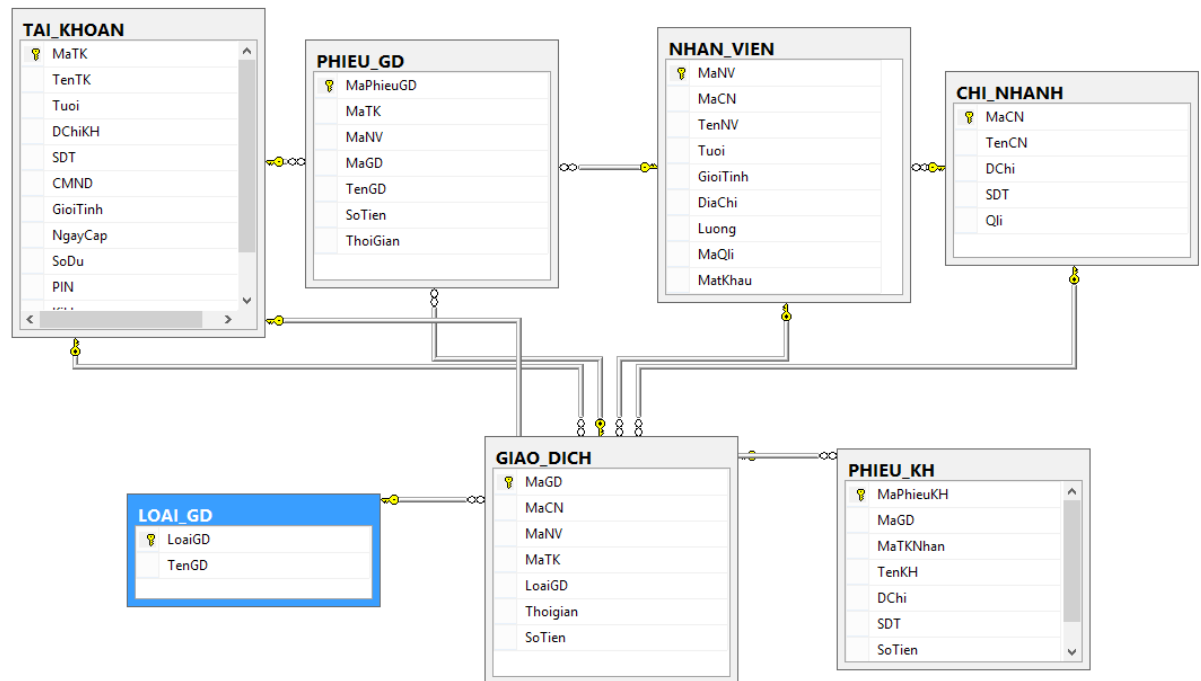
Bảng GIAO_DICH (MaGD, MaCN, MaNV, MaTK, LoaiGD, Thoigian, SoTien)

	Column Name	Data Type
▶	MaGD	nchar(10)
	MaCN	nchar(10)
	MaNV	nchar(10)
	MaTK	nchar(10)
	LoaiGD	nchar(10)
	Thoigian	datetime
	SoTien	money

Bảng CHI_NHANH (MaCN, TenCN, Dchi, SDT, Qli)

	Column Name	Data Type
▶	MaCN	nchar(10)
	TenCN	ntext
	DChi	ntext
	SDT	int
	Qli	nchar(10)

2. Quan Hệ giữa các bảng



IV. Trình bày code trên SQL

1. Tạo các bảng dữ liệu:

```

CREATE TABLE CHI_NHANH (
    MaCN nchar(10) primary key,
    TenCN ntext not null,
    DChi ntext not null,
    SDT int not null,
    Qli nchar(10) not null
)

CREATE TABLE NHAN_VIEN (
    MaNV nchar(10) not null primary key,
    MaCN nchar(10) not null foreign key references CHI_NHANH(MaCN),
    TenNV ntext not null,
    Tuoi int not null,
    GioiTinh nchar(10) not null,
    DiaChi ntext not null,
    Luong money not null,
    MaQli nchar(10) not null,
  
```

```
)  
    MatKhau nchar(10) not null
```

```
CREATE TABLE TAI_KHOAN (  
    MaTK nchar(10) not null primary key,  
    TenTK ntext not null,  
    Tuoi int not null,  
    DChiKH ntext not null,  
    SDT int not null,  
    CMND int not null,  
    GioiTinh nchar(10) not null,  
    NgayCap datetime not null,  
    SoDu money not null,  
    PIN nchar(10) not null,  
    KiHan int not null,  
    LaiSuat real not null  
)
```

```
CREATE TABLE PHIEU_KH (  
    MaPhieuKH nchar(10) not null primary key,  
    MaGD nchar(10) not null foreign key references GIAO_DICH(MaGD),  
    TenKH ntext not null,  
    DChi ntext not null,  
    SDT int not null,  
    SoTien money not null  
)
```

```
CREATE TABLE LOAI_GD (  
    LoaiGD nchar(10) not null primary key,  
    TenGD ntext not null  
)
```

```
CREATE TABLE GIAO_DICH (  
    MaGD nchar(10) not null primary key,  
    MaCN nchar(10) not null foreign key references CHI_NHANH(MaCN),  
    MaNV nchar(10) not null foreign key references NHAN_VIEN(MaNV),  
    MaTK nchar(10) not null foreign key references TAI_KHOAN(MaTK),  
    LoaiGD nchar(10) not null foreign key references LOAI_GD(LoaiGD),  
    Thoigian datetime not null,  
    SoTien money not null  
)
```

```

CREATE TABLE PHIEU_GD (
    MaPhieuGD nchar(10) not null primary key,
    MaTK nchar(10) not null foreign key references TAI_KHOAN(MaTK),
    MaNV nchar(10) not null foreign key references NHAN_VIEN(MaNV),
    MaGD nchar(10) not null foreign key references GIAO_DICH(MaGD),
    TenGD ntext not null,
    SoTien money not null,
    ThoiGian datetime not null,
)

```

2. Tạo các ràng buộc trigger trên các bảng:

a. Bảng CHI_NHANH

Số điện thoại cầu chi nhánh phải từ 9-12 số

```

CREATE TRIGGER SĐT_ChìNhanh
ON CHI_NHANH
AFTER INSERT, UPDATE
AS
IF UPDATE(SDT)
BEGIN
    DECLARE @mcn nchar (10)
    SET @mcn = (SELECT MaCN FROM inserted)
    DECLARE @SDT int
    SET @SDT = (SELECT CHI_NHANH.SDT FROM inserted,
    CHI_NHANH WHERE CHI_NHANH.MaCN =@mcn)
    DECLARE @Len int
    SET @Len = (SELECT LEN (@SDT))
    IF (@Len <9 OR @Len >12)
    BEGIN
        PRINT N'Số điện thoại phải từ 9 đến 12 số!'
        ROLLBACK TRAN
        RETURN
    END
END

```

b. Bảng NHAN_VIEN

Nhân viên phải lớn hơn 18 tuổi

```

CREATE TRIGGER TuổiNhanVien
ON NHAN_VIEN
AFTER INSERT, UPDATE

```

```

AS
IF UPDATE(Tuoi)
BEGIN
    DECLARE @Tuoi int
    SET @Tuoi = (SELECT Tuoi FROM inserted)
    IF (@Tuoi < 18)
    BEGIN
        PRINT N'Tuổi nhân viên phải đủ 18!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

Giới tính của nhân viên chỉ là Nam hoặc Nữ.

```

CREATE TRIGGER DungGioiTinh
ON NHAN_VIEN
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @GioiTinh nchar(10)
    SET @GioiTinh = (SELECT GioiTinh FROM inserted)
    IF (@GioiTinh <> 'Nam' and @GioiTinh <> 'Nữ')
    BEGIN
        PRINT N'Giới tính không được khác Nam hoặc Nữ!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

Lương nhân viên phải từ 3 triệu trở lên.

```

CREATE TRIGGER LuongNhanVien
ON NHAN_VIEN
AFTER INSERT, UPDATE
AS
IF UPDATE(Luong)
BEGIN
    DECLARE @Luong money
    SET @Luong = (SELECT Luong FROM inserted)
    IF (@Luong < 3000000)

```

```

        BEGIN
            PRINT N'Lương nhân viên phải từ 3 triệu trở lên!'
            ROLLBACK TRAN
            RETURN
        END
    END
END

```

Người quản lí cũng là một nhân viên.

```

CREATE TRIGGER QuanLyCungLaNhanVien
ON NHAN_VIEN
AFTER INSERT, UPDATE
AS
IF UPDATE(MaQli)
BEGIN
    DECLARE @idmn nchar(10)
    SET @idmn = (SELECT MaQli FROM inserted)
    DECLARE @count int
    SET @count = (SELECT COUNT(MaQli) FROM NHAN_VIEN
WHERE @idmn = NHAN_VIEN.MaNV)
    if(@count=0)
    BEGIN
        PRINT N'Người quản lí phải là một nhân viên!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

c. Bảng TAI_KHOAN

Tuổi của người tạo tài khoản phải đủ 18 tuổi.

```

CREATE TRIGGER TuoiKhachHang
ON TAI_KHOAN
AFTER INSERT, UPDATE
AS
IF UPDATE (Tuoi)
BEGIN
    DECLARE @age int
    SET @age = (select Tuoi from inserted)

```

```

        IF (@age < 18)
        BEGIN
            PRINT N'Tuổi phải đủ 18!'
            ROLLBACK TRAN
            RETURN
        END
    END
END

```

Số dư tài khoản phải trên 50 nghìn đồng.

```

CREATE TRIGGER SoDuTK
ON TAI_KHOAN
AFTER INSERT, UPDATE
AS
IF UPDATE (SoDu)
BEGIN
    DECLARE @eid nchar(10)
    SET @eid = (SELECT MaTK FROM inserted)
    DECLARE @tien money
    SET @tien = (SELECT SoDu from TAI_KHOAN where
    TAI_KHOAN.MaTK = @eid)
    IF (@tien < 50000)
    BEGIN
        PRINT N'Số dư tài khoản phải trên 50 nghìn đồng!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

Mã PIN phải có ít nhất 8 kí tự.

```

ALTER TRIGGER MaPIN
ON TAI_KHOAN
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @id nchar(10)
    SET @id = (SELECT MaTK from inserted)
    DECLARE @pin nchar(10)
    SET @pin = (SELECT TAI_KHOAN.PIN FROM inserted, TAI_KHOAN
    WHERE @id = TAI_KHOAN.MaTK)

```

```

DECLARE @len int
SET @len = (SELECT LEN(@pin))
IF (@len < 8)
BEGIN
    PRINT N'Mã PIN phải có ít nhất 8 kí tự!'
    ROLLBACK TRAN
    RETURN
END
END

```

Số CMND phải có 9 chữ số.

```

CREATE TRIGGER SoCMND
ON TAI_KHOAN
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @id nchar(10)
    SET @id = (SELECT MaTK FROM inserted)
    DECLARE @cmnd int
    SET @cmnd = (SELECT TAI_KHOAN.CMND FROM
inserted,TAI_KHOAN WHERE @id = TAI_KHOAN.MaTK)
    DECLARE @len int
    SET @len = (SELECT LEN(@cmnd))
    IF (@len <> 9)
    BEGIN
        PRINT N'Số CMND phải có 9 chữ số!'
        ROLLBACK TRAN
        RETURN
    END
END

```

Số điện thoại của tài khoản có từ 9-12 số.

```

CREATE TRIGGER SDT_TaiKhoan
ON TAI_KHOAN
AFTER INSERT, UPDATE
AS
IF UPDATE(SDT)
BEGIN
    DECLARE @mtk nchar (10)
    SET @mtk = (SELECT MaTK FROM inserted)

```

```

DECLARE @SDT int
SET @SDT = (SELECT TAI_KHOAN.SDT FROM inserted, TAI_KHOAN
WHERE TAI_KHOAN.MaTK = @mtk)
DECLARE @Len int
SET @Len = (SELECT LEN (@SDT))
IF (@Len <9 OR @Len >12)
BEGIN
    PRINT N'Số điện thoại phải từ 9 đến 12 số!'
    ROLLBACK TRAN
    RETURN
END
END

```

d. Bảng PHIEU_KH

Số tiền gửi vào tài khoản luôn luôn trên 100.000 đồng

```

CREATE TRIGGER SoTienGuiTren100
ON PHIEU_KH
AFTER INSERT, UPDATE
AS
IF UPDATE(SoTien)
BEGIN
    DECLARE @SoTienGui money
    SET @SoTienGui = (SELECT SoTien from inserted)
    IF (@SoTienGui <100000)
    BEGIN
        PRINT N'Số tiền gửi phải trên 100000!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

Mỗi phiếu khách hàng tương ứng với một giao dịch.

```

CREATE TRIGGER MoiPhieuTuongUngMotMaGD
ON PHIEU_KH
AFTER INSERT, UPDATE
AS
BEGIN

```



```

DECLARE @mgd nchar(10)
SET @mgd = (SELECT MaGD FROM inserted)
DECLARE @dem int
SET @dem = (SELECT COUNT(MaGD) FROM PHIEU_KH WHERE
@mgd = PHIEU_KH.MaGD)
IF (@dem > 1)
BEGIN
    PRINT N'Mỗi Phiếu chỉ tương ứng 1 mã GD!'
    ROLLBACK TRAN
    RETURN
END
END

```

Số điện thoại trong phiếu khách hàng phải từ 9-12 số.

```

CREATE TRIGGER SDT_PhieuKH
ON PHIEU_KH
AFTER INSERT, UPDATE
AS
IF UPDATE (SDT)
BEGIN
    DECLARE @MaPhieuKH nchar (10)
    SET @MaPhieuKH = (SELECT MaPhieuKH FROM inserted)
    DECLARE @SDT int
    SET @SDT = (SELECT PHIEU_KH.SDT FROM inserted, PHIEU_KH
WHERE PHIEU_KH.MaPhieuKH=@MaPhieuKH)
    DECLARE @Len int
    SET @Len = (SELECT LEN (@SDT))
    IF (@Len <9 OR @Len >12)
    BEGIN
        PRINT N'Số điện thoại phải từ 9 đến 12 số!'
        ROLLBACK TRAN
        RETURN
    END
END
END

```

e. Bảng LOAI_GD

Loại giao dịch chỉ gồm 3 giá trị: R - Rút, CK - Chuyển Khoản, GTK - Gửi Tiết Kiệm.

```

CREATE TRIGGER LoaiGiaoDich
ON LOAI_GD
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @lgd nchar(10)
    SET @lgd = (SELECT LoaiGD FROM inserted)
    IF (@lgd <> 'R' AND @lgd <> 'CK' AND @lgd <> 'GTK')
    BEGIN
        PRINT N'Loại giao dịch không đúng! Loại giao dịch chỉ gồm: R, CK,
        GTK.'
        ROLLBACK TRAN
        RETURN
    END
END

```

f. Bảng GIAO_DICH

Số tiền dùng trong giao dịch Rút hoặc Chuyển Khoản không được vượt quá số dư trong tài khoản, đồng thời tài khoản phải còn ít nhất 50000 đồng.

```

CREATE TRIGGER TienGiaoDich
ON GIAO_DICH
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @magd nchar(10)
    SET @magd = (SELECT MaGD FROM inserted)
    DECLARE @matk nchar(10)
    SET @matk = (SELECT MaTK FROM inserted)
    DECLARE @money money
    SET @money = (SELECT GIAO_DICH.SoTien FROM inserted,
    GIAO_DICH WHERE @magd = GIAO_DICH.MaGD)
    DECLARE @loaigd nchar(10)
    SET @loaigd = (SELECT GIAO_DICH.LoaiGD FROM inserted,
    GIAO_DICH WHERE @magd = GIAO_DICH.MaGD)
    DECLARE @sodu money
    SET @sodu = (SELECT SoDu FROM TAI_KHOAN WHERE @matk =
    TAI_KHOAN.MaTK)
    IF (@loaigd <> 'GTK' AND @money > @sodu - 50000)

```

```

BEGIN
    PRINT N'Số tiền trong tài khoản không đủ để giao dịch!'
    ROLLBACK TRAN
    RETURN
END
END

```

g. Bảng PHIEU_GD

Mỗi phiếu giao dịch chỉ tương ứng một mã giao dịch.

```

CREATE TRIGGER PhieuGD_GiaoDich
ON PHIEU_GD
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @mgd nchar(10)
    SET @mgd = (SELECT MaGD FROM inserted)
    DECLARE @dem int
    SET @dem = (SELECT COUNT(MaGD) FROM PHIEU_GD WHERE
    @mgd = PHIEU_GD.MaGD)
    IF (@dem > 1)
    BEGIN
        PRINT N'Mỗi Phiếu chỉ tương ứng 1 mã GD!'
        ROLLBACK TRAN
        RETURN
    END
END

```

3. Tạo các thủ tục thêm, sửa, xóa dữ liệu

STORE Thêm nhân viên

```

CREATE PROC Them_Nhan_Vien
    @MaNV nchar(10),
    @MaCN nchar(10),
    @TenNV ntext,
    @Tuoi int,
    @GioiTinh nchar(10)
    @DiaChi ntext,
    @Luong money,

```

```

        @MaQli nchar(10),
        @MatKhau nchar(10)
as
BEGIN TRAN
BEGIN TRY
        INSERT INTO NHAN_VIEN
        (
                MaNV, MaCN, TenNV, Tuoi, GioiTinh, DiaChi,
                Luong, MaQli, MatKhau
        )
        VALUES
        (
                @MaNV, @MaCN, @TenNV, @Tuoi,
                @GioiTinh, @DiaChi, @Luong, @MaQli, @MatKhau
        )
        COMMIT
END TRY
BEGIN CATCH
        ROLLBACK
        DECLARE @ErrorMessage VARCHAR(2000)
        SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
        PRINT(@ErrorMessage)
END CATCH
STORE Thêm Chi nhánh

```

```

CREATE PROC Them_Chi_Nhanh
@MaCN nchar (10),
@TenCN ntext,
@DChi ntext,
@SDT int,
@Qli nchar (10)
AS
BEGIN TRAN
BEGIN TRY
        INSERT INTO CHI_NHANH
        (
                MaCN, TenCN, DChi, SDT, Qli
        )
        VALUES
        (

```

```

        @MaCN ,@TenCN ,@DChi ,@SDT ,@Qli
    )
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

```

STORE Thêm Phiếu Khách hàng
CREATE PROC Them_PhiuKH
    @MaPhieuKH nchar (10),
    @MaGD nchar (10),
    @MaTKNhan nchar (10),
    @TenKH ntext,
    @DChi ntext,
    @SDT int,
    @SoTien money
as
BEGIN TRAN
BEGIN TRY
    INSERT INTO PHIEU_KH
        (
            MaPhieuKH, MaGD, MaTKNhan, TenKH, DChi, SDT,
SoTien
        )
        VALUES
        (
            @MaPhieuKH,@MaGD ,@MaTKNhan,@TenKH
,@DChi ,@SDT,@SoTien
        )
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)

```

END CATCH

STORE Sửa Nhân Viên

CREATE PROC Sua_Nhan_Vien

@MaNV nchar(10),

@MaCN nchar (10),

@TenNV ntext,

@Tuoi int,

@GioiTinh nchar (10),

@DiaChi ntext,

@Luong money,

@MaQli nchar (10),

@MatKhau nchar (10)

as

BEGIN TRAN

BEGIN TRY

UPDATE NHAN_VIEN

SET MaNV=@MaNV, MaCN=@MaCN, TenNV=@TenNV, Tuoi = @Tuoi,
GioiTinh=@GioiTinh, DiaChi=@DiaChi, Luong=@Luong, MaQli= @MaQli,
MatKhau=@MatKhau

COMMIT

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage VARCHAR(2000)

SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()

PRINT(@ErrorMessage)

END CATCH

STORE Sửa Chi Nhánh

CREATE PROC Sua_Chi_Nhanh

@MaCN nchar (10),

@TenCN ntext,

@DChi ntext,

@SDT int,

@QLi nchar (10)

AS

```

BEGIN TRAN
BEGIN TRY
    UPDATE CHI_NHANH
    SET MaCN= @MaCN, TenCN = @TenCN, DChi=@DChi, SDT = @SDT,
    QLi = @QLi

COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

```

STORE Sửa phiếu khách hàng
CREATE PROC Sua_PhieuKH
    @MaPhieuKH nchar (10),
    @MaGD nchar (10),
    @MaTKNhan nchar (10),
    @TenKH ntext,
    @DChi ntext,
    @SDT int,
    @SoTien money
as
BEGIN TRAN
BEGIN TRY
    UPDATE PHIEU_KH
    SET MaPhieuKH=@MaPhieuKH, MaGD = @MaGD,
    MaTKNhan=@MaTKNhan, TenKH=@TenKH, DChi=@DChi, SDT=@SDT,
    SoTien=@SoTien
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH
STORE Xóa Nhân viên

```

```

CREATE PROC Xoa_Nhan_Vien
    @MaNV nchar(10)
as
BEGIN TRAN
BEGIN TRY
    DELETE NHAN_VIEN Where NHAN_VIEN.MaNV = @MaNV
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

STORE xóa chi nhánh

```

CREATE PROC XoaChiNhanh
    @MaCN nchar (10)
AS
BEGIN TRAN
BEGIN TRY
    DELETE CHI_NHANH where CHI_NHANH.MaCN = @MaCN
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

STORE Thêm tài khoản

```

CREATE PROC ThemTaiKhoan
    @matk nchar(10),
    @tenkh ntext,
    @tuoi int,
    @dchi ntext,
    @sdt int,

```



```

    @cmnd int,
    @gioitinh nchar(10),
    @ngaycap datetime,
    @sodu money,
    @pin nchar(10),
    @laisuat real,
    @kihan int
AS
BEGIN TRAN
BEGIN TRY
    INSERT INTO TAI_KHOAN
    (
        MaTK, TenTK, Tuoi, DChiKH, SDT, CMND, GioiTinh, NgayCap, SoDu,
        PIN, LaiSuat, KiHan
    )
    VALUES
    (
        @matk, @tenkh, @tuoi, @dchi, @sdt, @cmnd, @gioitinh, @ngaycap,
        @sodu, @pin, @laisuat, @kihan
    )
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH
STORE Sửa tài khoản
CREAT PROC SuaTaiKhoan
    @MaTK nchar(10),
    @TenKH nchar (10),
    @Tuoi int,
    @DiaChi ntext,
    @SDT int,
    @CMND int,
    @GioiTinh nchar(10),
    @NgayCap datetime,
    @SoDu money,
    @PIN nchar(10),

```

```

    @KiHan int,
    @LaiSuat real
as
BEGIN TRAN
BEGIN TRY
    UPDATE TAI_KHOAN
    SET MaTK=@MaTK, TenTK=@TenKH, Tuoi=@Tuoi,
    @DiaChi=@DiaChi, SDT=@SDT, CMND=@CMND, GioiTinh=@GioiTinh,
    NgayCap=@NgayCap, SoDu=@SoDu, PIN=@PIN, KiHan=@KiHan,
    LaiSuat=@LaiSuat
    COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

```

STORE Xóa tài khoản
CREATE PROC XoaTaiKhoan
    @MaTK nchar(10)
as
BEGIN TRAN
BEGIN TRY
    DELETE TAI_KHOAN Where TAI_KHOAN.MaTK=@MaTK
COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    DECLARE @ErrorMessage VARCHAR(2000)
    SELECT @ErrorMessage = 'Lỗi: ' + ERROR_MESSAGE()
    PRINT(@ErrorMessage)
END CATCH

```

V. **Mô tả chức năng form**
1. Form đăng nhập

NV_Dang_Nhap

NGÂN HÀNG TATI

BANK

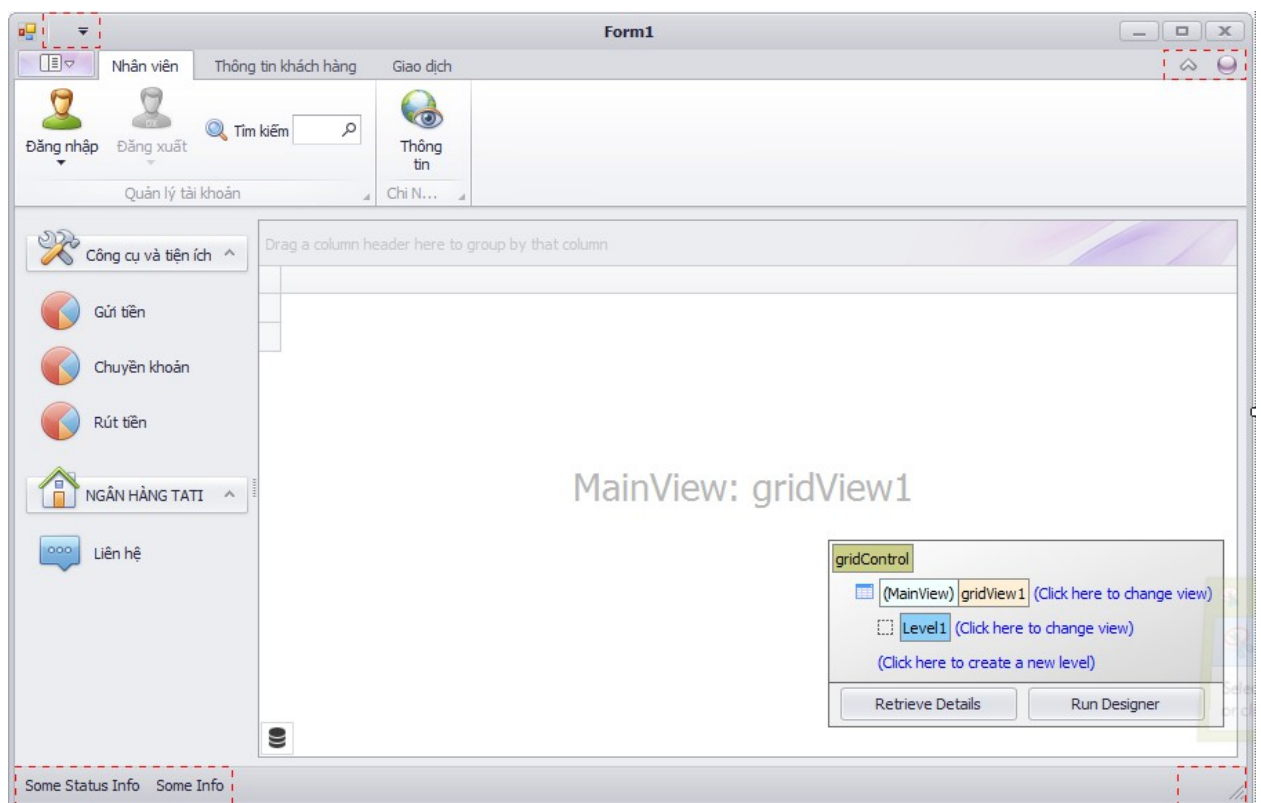
Username

Password

 Đăng nhập

Hình 1. Giao diện đăng nhập của phần mềm quản lý.

2. Giao diện chính của phần mềm quản lý.



Hình 2. Giao diện chính của phần mềm quản lý tài khoản ngân hàng

Đây là form đăng nhập dành cho nhân viên. Khi nhân viên đăng nhập hệ thống sẽ lưu vào thời gian làm việc của nhân viên đó. Khi khách hàng đến và thực hiện giao dịch sẽ có 3 hình thức cho khách hàng giao dịch là rút tiền, chuyển khoản và gửi tiền.

3. Form đăng ký tài khoản.

The screenshot shows a software window titled "Dang_Ky_Tai_Khoan". Inside the window, at the top center, is a logo for "NGÂN HÀNG TATI" in a stylized blue and green font. Below the logo, the text "Đăng ký tài khoản" is written in bold. The form consists of several input fields arranged in two columns. The left column includes fields for "Họ và tên", "Ngày tháng năm sinh", "Địa chỉ", "Số điện thoại", and "Chứng minh nhân dân". The right column includes a dropdown menu for "Giới tính", a field for "Ngày cấp thẻ", a field for "Mã Tài Khoản", and a field for "Số dư tài khoản". At the bottom center of the form is a button labeled "Đăng ký tài khoản".

Hình 3. Form đăng ký tài khoản

Nếu khách chưa có tài khoản và muốn đăng ký tài khoản phải cầm theo chứng minh thư và bản photo chứng minh thư cùng với 2 tấm hình 4x6. Đây là

form đăng ký tài khoản cho khách hàng mong muốn tạo tài khoản tại ngân hàng.

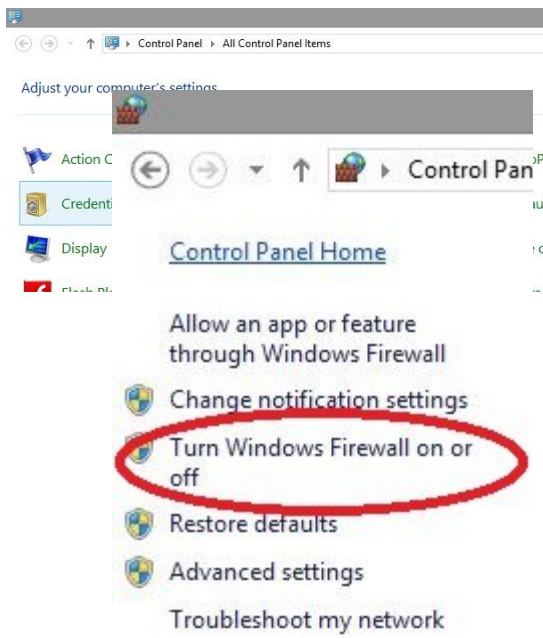
4. Form Giao dịch



The screenshot shows a Windows-style application window titled "Fr_GiaoDich". At the top center is the logo for "NGÂN HÀNG TATI" in a stylized blue and green font. Below the logo, the text "Khách hàng" is displayed. The form contains several input fields arranged in two columns: "Mã Tài Khoản" (Account Code), "Mã Chi Nhánh" (Branch Code), "Mã Giao Dịch" (Transaction Code), "Mã Nhân Viên" (Employee Code), "Loại Giao Dịch" (Transaction Type) with a dropdown arrow, "Thời Gian" (Time), "Mã Tài Khoản Nhận" (Receiving Account Code), and "Số Tiền" (Amount). There is a "Ký tên" (Signature) label and a large "Giao Dịch" (Transaction) button at the bottom center.

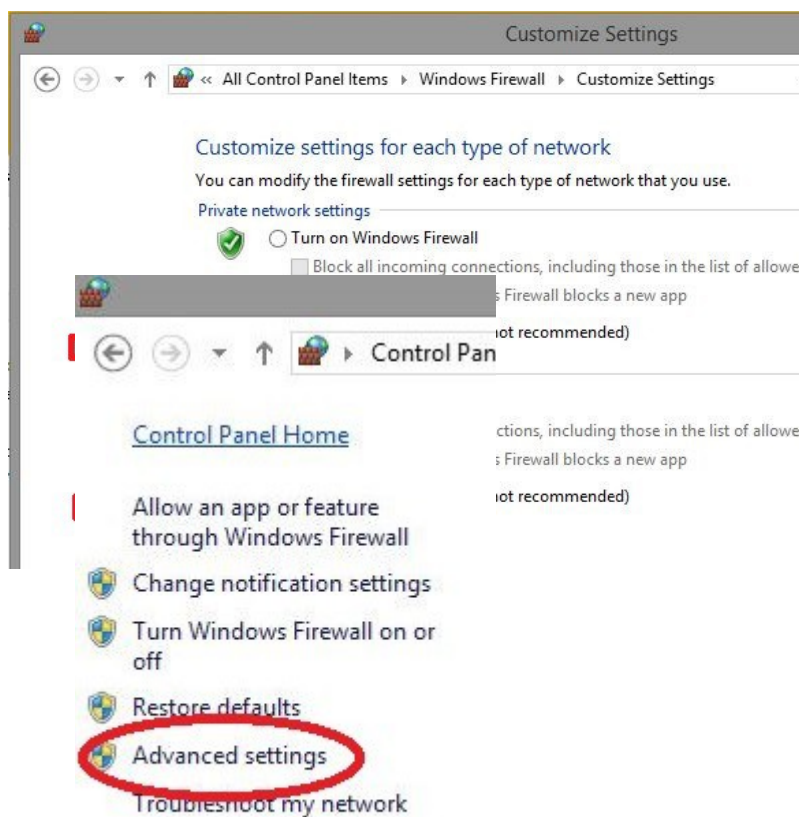
VI. Kết nối IP SQL Sever

Đầu tiên Click chuột trái vào Control Panel



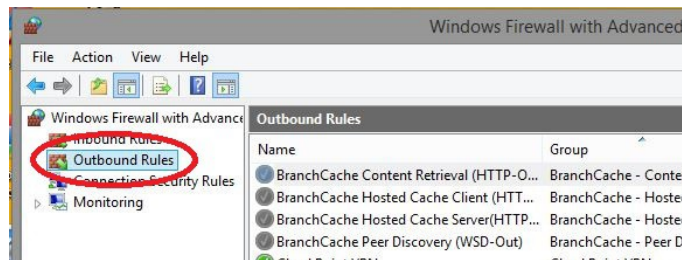
Sau đó, chọn Windows Firewall như hình.

Chọn như hình trên.

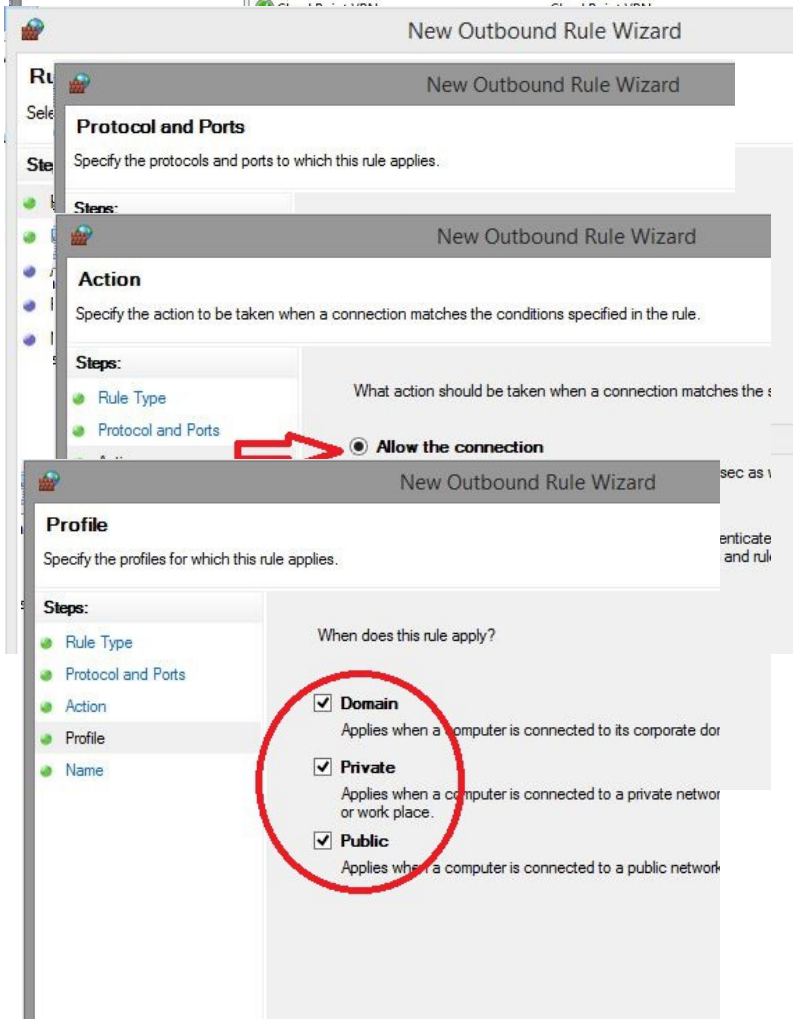


Chọn như hình vẽ để tắt hết tường lửa đi và nhấn OK. Việc này nhằm giúp việc kết nối được thành công 100%.

Sau khi đã tắt tường lửa xong bạn chọn tiếp Advanced settings như hình để tạo Port.



Chọn nút được đánh dấu hình tròn.
Sau đó nhấn vào New Rule bên tay
phải màn hình để mở hộp tạo Port.



Thiết lập Port bạn chọn như
hình trên. Rồi nhấn Next.

Chọn như hình. Và tại
Specific remote ports đánh
vào 1433. Rồi nhấn next.

Chọn Allow the connection để
cho phép kết nối. Tiếp tục
nhấn Next

Tick vào 3 ô trên. Rồi nhấn Next

New Outbound Rule Wizard

Name
Specify the name and description of this rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name**

Name:
KetNoiSQL

Description (optional):
Ket noi qua hub

Đặt tên cho Port tại Name và Mô tả Tại Description. Rồi nhấn Finish hoàn thành việc tạo Port.

Search

Everywhere ▾

SQL Server Configuration Manager



SQL Server Manag



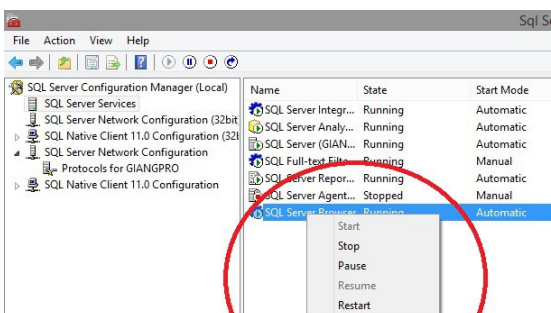
SQL Server Configu
Manager

sql

sql server surface area con

sql server management stu

sql server express



Tiếp đến là thiết lập SQL. Tìm kiếm SQL Server Configuration Manager và nhấn vào nó.

Nếu SQL Server Brower chưa bật thì hãy click chuột phải và chọn như hình.

SQL Server Browser Properties

Log On Service Advanced

Log on as:

☒ Built-in account:

Network Service

☐ This account:

Account Name: Browse

Password:

Confirm password:

Service status: Running

Start Stop Pause Restart

Network

SQL Server Browser Properties

Log On Service Advanced

General

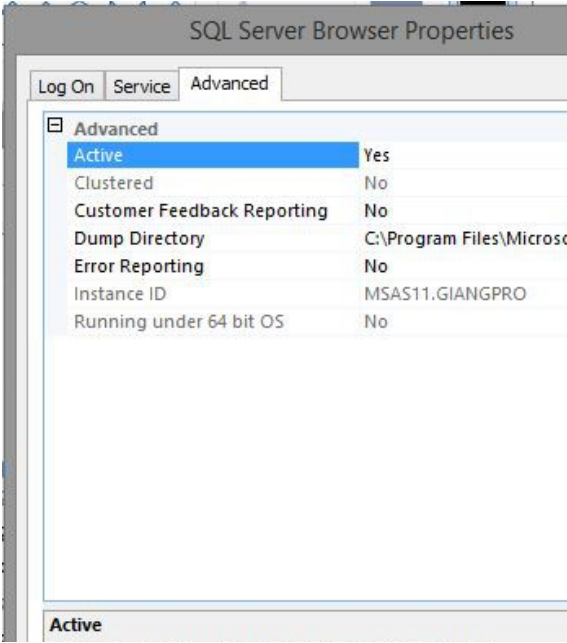
Binary Path	"C:\Program Files (x86)\Micro
Error Control	1
Exit Code	0
Host Name	GIANGPRO-PC
Name	SQL Server Browser
Process ID	4040
SQL Service Type	
Start Mode	Automatic
State	Running

Binary Path

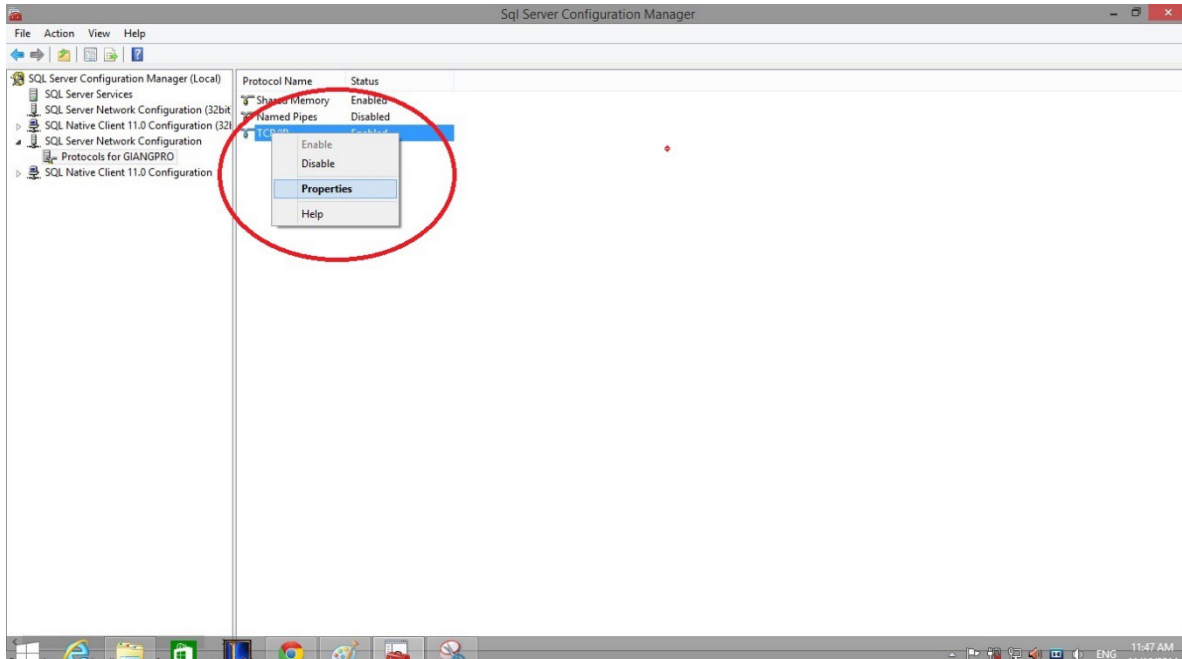
Path to the executable file for this service.

Chọn Built – in Account là Service. Làm như hình.

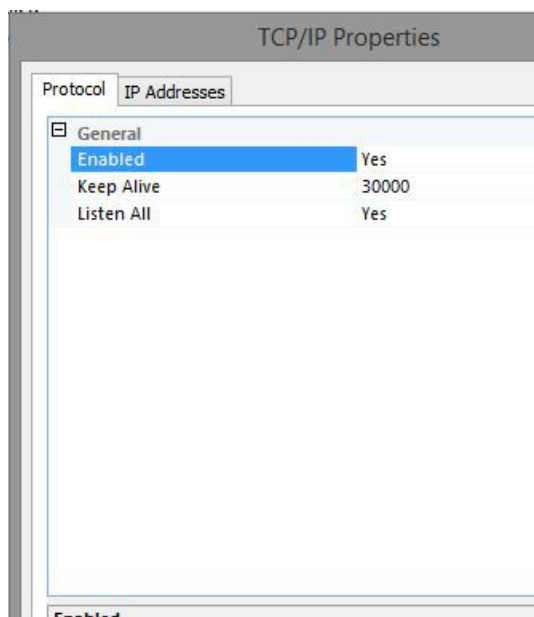
Bật Start Mode thành Automatic.



Active là Yes. Sau đó nhấn Apply OK.



Tiếp đến chọn SQL Server Network Configuration chọn Protocol for {Tên cấp phát server} khi chọn xong màn hình sẽ như hình ảnh và click chuột phải vào TCP/IP chọn như hình.



Hiện lên bảng trên và chọn Enabled là Yes.

```
C:\Windows\system32\cmd.exe

Default Gateway . . . . . :

Ethernet adapter vEthernet {Qualcomm Atheros AR5BWB222 Wireless Network Adapter Virtual Switch}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : lan

Ethernet adapter vEthernet {Realtek PCIe GBE Family Controller Virtual Switch}:

    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::c9ec:285a:da2c:a396%12
    IPv4 Address. . . . . : 192.168.1.84
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

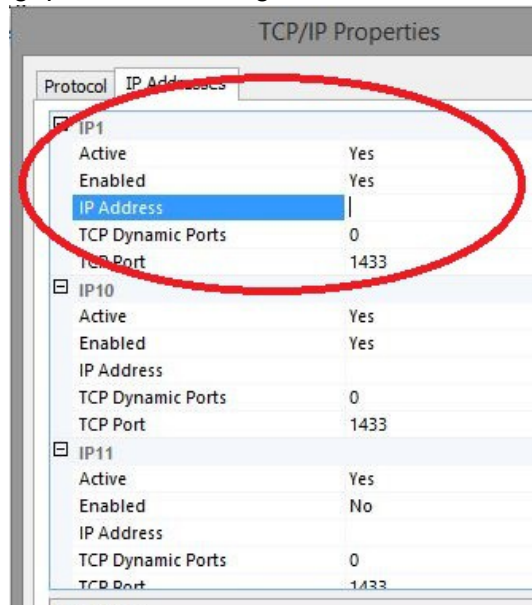
Wireless LAN adapter Local Area Connection* 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

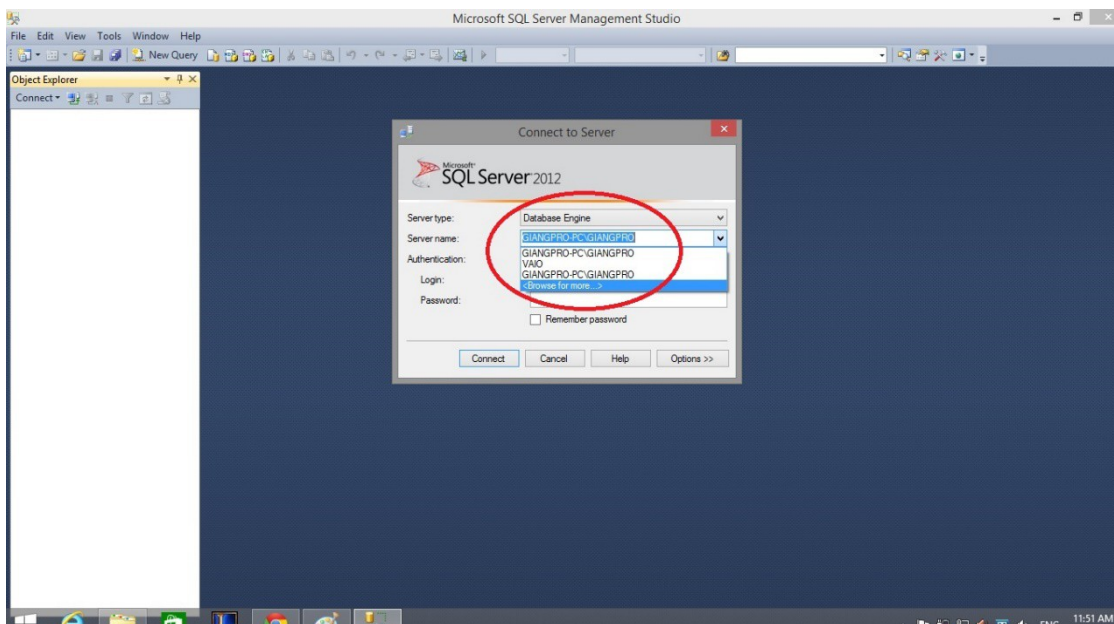
Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
```

Tiếp theo là tìm IP của máy Client và máy làm Server bằng cách vào CMD gõ ipconfig tiếp đó nhìn vào dòng Ipv4 Address và ghi nhớ nó.



Sau đó hãy viết IP đó vào IP address và để TCP Port là 1433 như hình. Và làm tương tự cho các cổng khác nhưng địa chỉ IP sẽ là máy muốn kết nối. Bây giờ là cách thiết lập trên máy trạm (client) máy chủ (Server) làm tương tự các bước như trên.



Sau khi làm xong các bước trên. Bắt đầu thử kết nối. Hãy mở SQL lên và xổ xuống như hình và chọn Brower for more ...

Sau khi chọn xong hãy chọn tab Network và đợi đôi chút và chọn tên Server muốn kết nối và nhấn OK. Và việc tiếp theo là chỉ cần nhập password và username là có thể kết nối.

VII.

VIII. Demo phần mềm

IX. Phân công công việc

STT	MSSV	Họ và tên	% đóng góp	Nội dung
1	12110193	Nguyễn Minh Tiến	50%	Công việc như nhau.
2	12110168	Lâm Khánh Tài	50%	Công việc như nhau

X. Kết luận

Ưu điểm: Đã nắm vững các kiến thức cần học liên quan đến thiết kế và quản lý cơ sở dữ liệu. Thành viên trong nhóm tích cực tìm hiểu, trao đổi thực hiện đề tài.

Khuyết điểm: Vì thời gian chuẩn bị không nhiều nên phần mềm còn tương đối đơn giản, tuy nhiên vẫn đảm bảo được các chức năng chính.

Hướng phát triển: Từ những chức năng cơ bản có thể mở rộng và nâng cấp để trở thành phần mềm hỗ trợ các giao dịch trong những ngân hàng vừa và nhỏ, đảm bảo được các yêu cầu về chuyên môn.