----BT VỀ THỦ TỤC----

Use bank

go

--Hãy tạo các thủ tục thực hiện các yêu cầu dưới đây

--1. Trả về tên chi nhánh ngân hàng nếu biết mã của nó.

go

create proc tenchinhanh (@ma char(5), @brname nvarchar(50) output)

as

begin

set @brname = (select br\_name

from Branch

where BR\_id=@ma)

end

--goi

declare @brname nvarchar(50)

exec tenchinhanh 'VN013', @brname out

print @brname

--2. Trả về tên, địa chỉ và số điện thoại của khách hàng nếu biết mã khách.

go

create proc khachhang (@mk char(6), @ten nvarchar(50) output, @dchi nvarchar(150) output, @sdt varchar(15) output)

as

begin

select @ten=Cust\_name, @dchi=Cust\_ad, @sdt=Cust\_phone

from customer

where @mk=Cust\_id

group by Cust\_name, Cust\_ad, Cust\_phone

end

--goi

declare @ten nvarchar(50), @dchi nvarchar(150), @sdt varchar(15)

exec khachhang '000003', @ten out, @dchi out, @sdt out

print @ten

print @dchi

print @sdt

--3. In ra danh sách khách hàng của một chi nhánh cụ thể nếu biết mã chi nhánh đó.

go

create function fList (@macn char(5))

returns table

as

return select cust\_id, cust\_name

from customer

where Br\_id=@macn

select \* from dbo.fList('VT010')

--4. Kiểm tra một khách hàng nào đó đã tồn tại trong hệ thống CSDL của ngân hàng chưa

--nếu biết: họ tên, số điện thoại của họ. Đã tồn tại trả về 1, ngược lại trả về 0

go

Alter proc ktra (@hoten nvarchar(50), @sdt varchar(15), @ret varchar(1) out)

as

begin

declare @temp int=0

select @temp=count(\*) from account join customer on account.cust\_id=customer.Cust\_id where Cust\_name=@hoten and Cust\_phone=@sdt

if @temp=0

set @ret='0'

else

set @ret='1'

end

declare @ret varchar(1)

exec spCheck N'Lê Anh Huy','0966654330', @ret out

print @ret

declare @ret varchar(1)

exec spCheck N'Trương Duy Tường','01645500071',@ret out

print @ret

-----

go

create proc TonTai (@ten nvarchar(100),@sdt varchar(15), @ret int output)

as

begin

declare @count int=0

select @count = count(cust\_id)

from customer

where @ten=cust\_name and @sdt=cust\_phone

if @count=0

set @ret=0

else

set @ret=1

end

declare @ret int

exec TonTai N'Hồ Trần Nhật Khánh','01648438000', @ret output

print @ret

--5. Cập nhật số tiền trong tài khoản nếu biết mã số tài khoản và số tiền mới.

--Thành công trả về 1, thất bại trả về 0

go

create proc capnhattien (@ac\_no varchar(50),@money numeric(15,0),@ret int output)

as

begin

IF EXISTS ( SELECT ac\_no FROM ACCOUNT WHERE ac\_no=@ac\_no)

begin

update ACCOUNT

set ac\_balance=@money

where ac\_no=@ac\_no

set @ret=1

end

else

set @ret=0

end

declare @ret int

exec capnhattien'10000000','-6107000',@ret out

print @ret

--6. Cập nhật địa chỉ của khách hàng nếu biết mã số của họ. Thành công trả về 1, thất bại trả về 0

go

CREATE PROC capnhatdchi (@ma VARCHAR(15),@dchi NVARCHAR(100),@RET INT OUTPUT)

AS

BEGIN

IF EXISTS (SELECT CUST\_ID FROM CUSTOMER WHERE CUST\_ID=@ma)

BEGIN

UPDATE CUSTOMER

SET CUST\_AD=@dchi

WHERE CUST\_ID=@ma

SET @RET=1

END

ELSE

SET @RET=0

END

DECLARE @RET INT

EXEC capnhatdchi '000001',N'NGUYỄN TIẾN DUẨN - THÔN 3 - XÃ DHÊYANG - EAHLEO - ĐĂKLĂK', @RET OUT

PRINT @RET

--7. Trả về số tiền có trong tài khoản nếu biết mã tài khoản.

go

create proc tientk (@matk char(10), @tientk numeric(15,0) output)

as

begin

set @tientk=(select ac\_balance

from account

where Ac\_no=@matk)

end

--goi

declare @tientk char(10)

exec tientk '1000000001', @tientk out

print @tientk

--8. Trả về số lượng khách hàng, tổng tiền trong các tài khoản nếu biết mã chi nhánh.

go

create proc khandtien (@macn char(5), @slkh int output, @tongtien int output)

as

begin

set @slkh=(select count(cust\_name)

from customer join Branch on customer.Br\_id=Branch.BR\_id

where Branch.BR\_id=@macn)

set @tongtien=(select sum(ac\_balance)

from Branch join customer on Branch.BR\_id=customer.Br\_id

join account on customer.Cust\_id=account.cust\_id

where Branch.BR\_id=@macn)

end

--goi

declare @slkh int, @tongtien int

exec khandtien 'VT009', @slkh out, @tongtien out

print @slkh

print @tongtien

--9. Kiểm tra một giao dịch có bất thường hay không nếu biết mã giao dịch.

--Giao dịch bất thường: giao dịch gửi diễn ra ngoài giờ hành chính, giao dịch rút diễn

--ra vào thời điểm 0am  3am

go

create proc kiemtraGDBT (@t\_id varchar(50), @ret nvarchar(50) output)

as

begin

if exists (select t\_id from transactions where t\_time between '00:00' and '03:00' and t\_id=@t\_id)

begin

set @ret=N'Giao dịch bất thường'

end

else

set @ret=N'Không có gì bất thường'

end

declare @ret nvarchar(50)

exec kiemtraGDBT '0000000264', @ret out

print @ret

--10. Trả về mã giao dịch mới. Mã giao dịch tiếp theo được tính như sau:

--MAX(mã giao dịch đang có) + 1. Hãy đảm bảo số lượng kí tự luôn đúng với quy định về mã giao dịch

go

alter proc Magdm (@ma char(10) output)

as

begin

select @ma = MAX(t\_id)+1

from transactions

set @ma = '0000000'+ @ma

end

declare @ma char(10)

exec Magdm @ma out

print @ma

-----------------

go

alter proc spNewTransID @new varchar(10) output

as

begin

declare @max int, @temp varchar(10)

set @max=(select max(cast(t\_id as int)) from transactions)

set @temp=cast(@max+1 as varchar)

set @new= replicate('0',10 - len(@temp))+ @temp

end

declare @new varchar(10)

exec spNewTransID @new out

print @new

--11. Thêm một bản ghi vào bảng TRANSACTIONS nếu biết các thông tin ngày giao dịch, thời gian giao dịch,

--số tài khoản, loại giao dịch, số tiền giao dịch. Công việc cần làm bao gồm:

--a. Kiểm tra ngày và thời gian giao dịch có hợp lệ không. Nếu không, ngừng xử lý

--b. Kiểm tra số tài khoản có tồn tại trong bảng ACCOUNT không? Nếu không, ngừng xử lý

--c. Kiểm tra loại giao dịch có phù hợp không? Nếu không, ngừng xử lý

--d. Kiểm tra số tiền có hợp lệ không (lớn hơn 0)? Nếu không, ngừng xử lý

--e. Tính mã giao dịch mới

--f. Thêm mới bản ghi vào bảng TRANSACTIONS

--g. Cập nhật bảng ACCOUNT bằng cách cộng hoặc trừ số tiền vừa thực hiện giao dịch tùy theo loại giao dịch

go

create proc UPDATE\_T\_AC (@t\_date date, @t\_time time, @ac\_no varchar(10),@t\_type int,@t\_amount numeric(15,0))

as

begin

--a

declare @maGDM varchar(10),@count varchar(10)

if @t\_date > getdate() or @t\_time between '07:00' and '11:30' and @t\_time between '11:30' and '16:00'

begin

print N'ngày giao dịch không hợp lệ'

return

end

--b

else if not exists (select ac\_no from account where ac\_no=@Ac\_no)

begin

print N'không tồn tại số tài khoản'

return

end

--c

else if @t\_type!=0 and @t\_type !=1

begin

print N'loại giao dịch không hợp lệ'

return

end

--d

else if @t\_amount <=0

begin

print N'số tiền GD không hợp lệ'

return

end

--e

select @count= MAX(t\_id)+1

from transactions

set @maGDM =REPLICATE ('0',10-len(@count))+@count

--f

INSERT INTO TRANSACTIONS VALUES(@maGDM,@t\_type,@t\_amount,@t\_date,@t\_time,@ac\_no)

--g

if @@ROWCOUNT>0

begin

IF @t\_type=0

UPDATE ACCOUNT

SET ac\_balance=ac\_balance-@t\_amount

WHERE ac\_no=@Ac\_no

ELSE

UPDATE ACCOUNT

SET ac\_balance=ac\_balance+@t\_amount

WHERE ac\_no=@Ac\_no

end

else

print N'UPDATE K THÀNH CÔNG'

end

EXEC UPDATE\_T\_AC '2022/10/17','12:00','1000000001',1,'1'

go

create proc Update\_T\_AC (@t\_date date , @t\_time time ,@ac\_no varchar(10),@t\_type int,@t\_amount numeric(15,0))

as

begin

declare @maGDN varchar(10)

--12. Thêm mới một tài khoản nếu biết: mã khách hàng, loại tài khoản, số tiền trong tài khoản.

--Bao gồm những công việc sau:

--a. Kiểm tra mã khách hàng đã tồn tại trong bảng CUSTOMER chưa? Nếu chưa, ngừng xử lý

--b. Kiểm tra loại tài khoản có hợp lệ không? Nếu không, ngừng xử lý

--c. Kiểm tra số tiền có hợp lệ không? Nếu NULL thì để mặc định là 50000, nhỏ hơn 0 thì ngừng xử lý.

--d. Tính số tài khoản mới. Số tài khoản mới bằng MAX(các số tài khoản cũ) + 1

--e. Thêm mới bản ghi vào bảng ACCOUNT với dữ liệu đã có.

go

create proc update\_tk (@makh char (6),@t\_type int, @loaitk int, @tientk numeric(15,0))

as

begin

if not exists (select cust\_id from customer where Cust\_id=@makh)

begin

print 'stop A'

return

end

else if @t\_type!=0 and @

end

go

create proc cau12 (@cust\_id char(6), @t\_type int,@ac\_balance numeric(15,0))

as

begin

if not exists (select cust\_id from customer where cust\_id=@cust\_id)

begin

print 'STOP A'

RETURN

end

RETURN

else if @t\_type!=0 and @t\_type!=1

begin

print 'STOP B'

end

else if @ac\_balance<=0

begin

print 'STOP C'

RETURN

end

else if @ac\_balance is null

begin

set @ac\_balance=50000

end

ELSE

BEGIN

SET @ac\_balance=@ac\_balance

END

declare @STK1 char(10)

select @STK1=MAX(AC\_NO)+1

from ACCOUNT

INSERT INTO ACCOUNT VALUES (@STK1,@ac\_balance,@t\_type,@cust\_id)

end

exec cau12'000001',1,NULL

---------------------------------HAM--------------------------------------

/\*

1.Kiểm tra thông tin kh đã tồn tại trong hệ thống hay chưa nếu biết tên và sdt.

Tồn tại trả về 1, k tồn tại trả về 0

--input: họ tên, sdt

--output: gtri

\*/

go

create function fThongTin (@ten nvarchar(50), @sdt varchar(15) )

returns varchar(50)

as

begin

DECLARE @temp varchar(1)

if exists (select cust\_name, Cust\_phone from customer where Cust\_name=@ten and cust\_phone=@sdt)

begin

set @temp=1

end

else

begin

set @temp=0

end

return @temp

end

print dbo.fThongTin(N'Hà Công Lực',

--FUNCTION

GO

create function fnewTranID():

returns varchar(10)

as

begin

declare @maMax varchar(15), @len varchar(15) ,@maMoi varchar(10)

set @maMax=(select max(t\_id) from transactions)

set @maMoi = @maMax + 1

set len =(select top 1 len(t\_id) from transactions)

set @maMoi= replicate('0',10 - len(@maMoi))+ @maMoi

return @maMoi

end

--test

print N'Giao dịch mới là'.dbo.fnewTranID()

select dbo.fnewTranID

--Trả về tên của khách hàng nếu biết mã khách

go

alter function fcau5(@maKH varchar(10))

returns nvarchar(100)

as

begin

declare @name nvarchar(100)

set @name=(select Cust\_name

from customer

where Cust\_id=@maKH)

return @name

end

--test

select \*from customer

print N'tên khách hàng ' +dbo.fcau5('000002')