1. **một số câu lệnh cơ bản.**
   1. **Tạo bảng.**

cú pháp:

CREATE TABLE tên\_bảng

(

*tên\_cột thuộc\_tính\_cột các\_ràng\_buộc*

[ *, ...*

*, tên\_cột\_n thuộc\_tính\_cột\_n các\_ràng\_buộc\_cột\_n* ]

[ *, các\_ràng\_buộc\_trên\_bảng* ]

)

Trong đó:*tên\_bảng* Tên của bảng cần tạo (<=128 ký tự)

*tên\_cột* Tên của cột (field / trường) cần định nghĩa

*thuộc\_tính\_cột* Gồm:

• Kiểu dữ liệu của cột (field).

• Giá trị mặc định của cột (filed).

• IDENTITY - giá trị tự động tăng, dùng với field kiểu số.

• NULL / NOT NULL

*các\_ràng\_buộc* Các ràng buộc được sử dụng trên mỗi cột (field) hoặc trên bảng.

Vd: tạo bảng có tên user chứa thông tin tên và số điện thoại của người dùng.

CREATE TABLE user

(

Id [int] IDENTITY(1,1) PRIMARY KEY,

Name user NVARCHAR(50) NOTNULL,

PhoneNumber NVARCHAR(20) NOTNULL

)

* 1. **Câu lệnh SELECT**

Công dụng:

- Chọn các field từ bảng.

Cú pháp:

SELECT [ALL | DISTINCT][TOP n] danh\_sách\_chọn

[INTO *tên\_bảng\_mới*]

FROM *danh\_sách\_bảng* [ | *khung\_nhìn*]

[WHERE *điều\_kiện*]

[GROUP BY *danh\_sách\_cột*]

[HAVING *điều\_kiện*]

[ORDER BY *cột\_sắp\_xếp*]

[COMPUTE *danh\_sách\_hàm\_gộp* [BY *danh\_sách\_cột*]]

Chú ý: Các thành phần trong câu lệnh SELECT phải được sử dụng theo đúng thứ tự trên.

Vd: câu lệnh sau cho biết các thông tin trong bản user

SELECT \* FROM user

* 1. **Câu lệnh thêm dữ liệu INSET**

Công dụng:

- Thêm dòng dữ liệu (mẫu tin/record) vào bảng.

Cú pháp:

INSERT INTO *tên\_bảng*

[(*danh\_sách\_cột*)] VALUES(*danh\_sách\_trị*)

Vd: thêm một dòng vào bản user ;

INSERT INTO user Name user, PhoneNumber VALUES(‘ ten\_nguoi\_dung ’,’012345678’)

* 1. **Câu lệnh cập nhập dữ liệu UPDATE**

Công dụng:

- Cập nhật dữ liệu trong các bảng.

Cú pháp:

UPDATE tên\_bảng

SET *tên\_cột* = *biểu\_thức*

[ , ...

, *tên\_cột\_k* = *biểu\_thức\_k*]

[FROM *danh\_sách\_bảng*]

[WHERE *điều\_kiện*]

Vd: update với tên người dung mới cùng số điện thoại mới tại bảng user có id =1

UPDATE user SET Name user = ‘tên mới’, PhoneNumber = ‘số điện thoại mới’ WHERE Id=1

* 1. **Câu lệnh xóa dữ liệu DELETE**

Công dụng:

- Để xóa dữ liệu trong bảng.

Cú pháp:

DELETE FROM *tên\_bảng*

[FROM *danh\_sách\_bảng*]

[WHERE *điều\_kiện*]

Vd: Xóa tên người dùng có id =1;

DELETE FROM user WHERE Id=1

1. Một số hàm cơ bản sử dụng trong chương trình.
   1. Khai báo namespace dược sử dụng.

using System.Data.SqlClient;

* 1. Khởi tạo class Ketnoi.

public SqlConnection kn = new SqlConnection();

public void kn\_csdl()

{

string chuoikn = @" DataSource={SERVERNAME};Initial Catalog={DATABASENAME};Integrated Security=True; User ID={Username};Password={Password} ";

kn.ConnectionString = chuoikn;

kn.Open();

}

public void dongketnoi()

{

if (kn.State == ConnectionState.Open)

{ kn.Close(); }

* Hàm sử lý câu lệnh SQL

public int xulydulieu(string caulenhsql)

{

int kq = 0;

try

{

kn\_csdl();

SqlCommand lenh = new SqlCommand(caulenhsql, kn);

kq = lenh.ExecuteNonQuery();

}

catch (Exception ex)

{

//Thông báo lỗi ra!

kq = 0;

}

finally

{

dongketnoi();

}

return kq;

}

* Hàm lấy bảng dữ liệu

public DataTable laybang(string caulenh)

{

DataTable bangdulieu = new DataTable();

try

{

kn\_csdl();

SqlDataAdapter Adapter = new SqlDataAdapter(caulenh, kn);

DataSet ds = new DataSet();

Adapter.Fill(bangdulieu);

}

catch (System.Exception)

{

bangdulieu = null;

}

finally

{

dongketnoi();

}

return bangdulieu;

}

* 1. Sử dụng.
* Lấy dữ liệu cho datagripview

public void laydulieuchobang\_user()

{

KetNoi kn = new KetNoi();

DataTable dt = new DataTable();

dt = kn.laybang("select \* from tbl\_user");

dataGridView1.DataSource = dt;

}

* Hàm lấy danh sách máy:

private List<string> List\_may()

{

List<string> list = new List<string>();

try

{

KetNoi kn = new KetNoi();

DataTable dt = new DataTable();

dt = kn.laybang("SELECT \* FROM may4 ");

if (dt != null)

{

foreach (DataRow dr in dt.Rows)

{

termlist.Add(dr["MAY"].ToString());

}

}

}

catch (Exception)

{

MessageBox.Show("có lỗi xảy ra");

}

return list;

}

* Hàm xữ lý thủ tục sql trong C#;

public void laydulieuchobang\_user()

{

Try

{

KetNoi kn = new KetNoi();

SqlCommand cmd = new SqlCommand(caulenh, kn);// ten proc trong csdl

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@tenthamso",thamso));

cmd.ExecuteNonQuery();

cmd.Dispose();

}

Catch{

//báo lỗi.

}

}

* 1. Một số thủ tục trong sql server
* Thủ tục update trạng thái máy.

CREATE PROCEDURE [dbo].[Machine\_State]

(

@goidulieu varchar(500),--tham số

@goidulieu\_I varchar(max)

)

AS

BEGIN

UPDATE may1 set [trang\_thai] = @goidulieu where Id=774;

UPDATE may1 set [dong\_dien] = @goidulieu\_I where Id=774;

END

* Thủ tục update trạng thái máy cho điện thoại.

CREATE PROCEDURE [dbo].[Machine\_State\_Detail]

AS

BEGIN

declare @goidulieu varchar(500);

declare @goidulieu\_I varchar(max);

declare @step int=1;

declare @ketthuc int;

declare @byte char(1);

declare @tb table (ID int IDENTITY(1,1) NOT NULL,data varchar(30));

declare @dong\_I varchar(30);

set @goidulieu = ( select [trang\_thai] from may1 where Id=774);

set @goidulieu\_I = ( select [dong\_dien] from may1 where Id=774);

INSERT @tb (data) SELECT \* FROM [dbo].[SplitStringToTable](@goidulieu\_I, '/')

while (@step<=360)

begin;

set @byte =(SELECT SUBSTRING(@goidulieu,@step,1))

set @dong\_I=(select data from @tb where ID=@step)

UPDATE may4 SET [Dong\_dien] = @dong\_I WHERE Id=@step;

if (@byte='1' )

begin;

UPDATE may4 SET [Trang\_thai] = 'ON' WHERE Id=@step;

SET @step=@step+1;

end;

else

begin;

UPDATE may4 SET [Trang\_thai] = 'OFF' WHERE Id=@step;

SET @step=@step+1;

end;

end;

END

* Thủ tục lấy giá trị lô lớn nhất

CREATE PROCEDURE [dbo].[Get\_MAX\_LotIndex]

AS

BEGIN

SELECT MAX(LotNumber) AS MAX\_LotIndex FROM Tb\_Session

END

Quản trị sản xuất có tầm quan trọng đặc biệt trong hoạt động của doanh nghiệp nó chiếm đến 60% ảnh hưởng đến sự thành công của doanh nghiệp. Nếu quản trị tốt và ứng dụng các phương pháp quản trị khoa học thì tạo khả năng sinh lợi lớn cho doanh nghiệp. Ngược lại nếu quản trị không tốt sẽ làm cho doanh nghiệp bị thua lổ thậm chí bị phá sản

Dự toán vật liêu:

1. Sản xuất đúng thời hạn (Just in time - JIT) 1.1. Khái niệm JIT là một khái niệm trong sản xuất hiện đại, được hiểu ngắn gọn nhất "đúng sản phẩm - đúng số lượng - đúng nơi - đúng thời điểm cần thiết".(quản tri xa)
2. - Lãng phí do sản xuất dư thừa hoặc quá sớm.
3. - Lãng phí do lưu kho nhiều.