



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

namespace CaseStudy

{

class DataSetDemo

{

static void Main()

{

SqlConnection con = new SqlConnection();

con.ConnectionString = @"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practicedb; integrated security = true";

SqlDataAdapter ada = new SqlDataAdapter("Select \* from Employee", con);

DataSet ds = new DataSet();

ada.Fill(ds, "Employee");

//Console.WriteLine(ds.Tables.Count);

foreach (DataRow dr in ds.Tables["Employee"].Rows)

{

foreach (DataColumn dc in ds.Tables[0].Columns)

{

Console.WriteLine(dr[dc]);

}

}

}

}

}

<appSettings>

<add key="connection" value="data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practicedb; integrated security = true"></add>

</appSettings>

Checked keyword is used to explicitly check overflow and conversion of integral type values at compile time

Checked unchecked

checked

{

int val = int.MaxValue;

Console.WriteLine(val + 2);

}

-- Joins

Default Join :: Inner join

select \* from empdata

select \* from empdat1

-- Natural Join and inner join

-- In natural join the common columns will come once

-- supported in sql server

-- select \* from empdata natural join empdat1

select \* from empdata inner join empdat1 on empdata.id = empdat1.id

select \* from empdata left outer join empdat1 on empdata.id = empdat1.id

select \* from empdata full outer join empdat1 on empdata.id = empdat1.id

select a.name , b.name from employee a join employee b where a.managerid=b.employeeid