SQL

```
CREATE OR REPLACE VIEW emp_per_dept AS
SELECT dept.deptno, dname, COUNT(emp.deptno) AS num_emps
FROM emp RIGHT JOIN dept ON emp.deptno = dept.deptno
GROUP BY dept.deptno, dname
ORDER BY dname ASC;

CREATE OR REPLACE PROCEDURE UPDATE_EMPLOYEE
(
        pEmpno IN NUMBER,
        pEname IN VARCHAR2,
        pJob IN VARCHAR2,
        pMgr IN NUMBER,
        pHiredate IN DATE,
        pSal IN NUMBER,
        pComm IN NUMBER,
        pDeptno IN NUMBER
)
IS
BEGIN
        UPDATE emp SET ename = pEname,
                job = pJob,
                mgr = pMgr,
                hiredate = pHiredate,
                sal = pSal,
                comm = pComm,
                deptno = pDeptno
        WHERE empno = pEmpno;
END;
/

CREATE OR REPLACE PACKAGE emp_dept_package IS
   PROCEDURE create_new_employee(new_empno IN NUMBER, new_ename IN VARCHAR2, new_job IN
         new_mgr IN VARCHAR2, new_hiredate IN DATE, new_sal IN NUMBER, new_comm IN NUMBI
         new_dept IN VARCHAR2, new_dept_loc IN VARCHAR2);
   PROCEDURE create_department(new_dname IN VARCHAR2, new_loc IN VARCHAR2);
   FUNCTION find_employee(ename_to_find VARCHAR2) RETURN NUMBER;
   FUNCTION find_department(dname_to_find IN VARCHAR2) RETURN NUMBER;
   END;
/

CREATE OR REPLACE PACKAGE BODY emp_dept_package IS
   last_deptno dept.deptno%TYPE;

   PROCEDURE create_department(new_dname IN VARCHAR2, new_loc IN VARCHAR2) IS
      new_deptno dept.deptno%TYPE;
   BEGIN
      SELECT MAX(deptno) INTO last_deptno FROM dept;
      new_deptno := last_deptno + 10;
      INSERT INTO dept (deptno, dname, loc) VALUES (new_deptno, new_dname, new_loc);
      last_deptno := new_deptno;
   END;
   FUNCTION find_department(dname_to_find IN VARCHAR2) RETURN NUMBER IS
      CURSOR dept_cursor IS SELECT deptno FROM dept WHERE dname = dname_to_find;
      found_deptno dept.deptno%TYPE;
   BEGIN
      OPEN dept_cursor;
      FETCH dept_cursor INTO found_deptno;
      IF dept_cursor%NOTFOUND THEN
         found_deptno := NULL;
      END IF;
      CLOSE dept_cursor;
      RETURN found_deptno;
   END;

   FUNCTION find_employee(ename_to_find IN VARCHAR2) RETURN NUMBER IS
      CURSOR emp_cursor IS SELECT empno FROM emp WHERE ename = ename_to_find;
      f_emp emp_cursor%ROWTYPE;
      found_empno emp.empno%TYPE;
   BEGIN
      OPEN emp_cursor;
      FETCH emp_cursor INTO f_emp;
      IF emp_cursor%FOUND THEN
         found_empno := f_emp.empno;
      ELSE
         found_empno := NULL;
      END IF;
      CLOSE emp_cursor;
      RETURN found_empno;
   END;

   PROCEDURE create_new_employee(new_empno IN NUMBER, new_ename IN VARCHAR2, new_job IN
         new_mgr IN VARCHAR2, new_hiredate IN DATE, new_sal IN NUMBER, new_comm IN NUMBE
         new_dept IN VARCHAR2, new_dept_loc VARCHAR2) IS
      new_deptno dept.deptno%TYPE;
   BEGIN
      new_deptno := find_department(new_dept);
      IF new_deptno IS NULL THEN
         create_department(new_dept, new_dept_loc);
         new_deptno := last_deptno;
      END IF;
      INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
      VALUES (new_empno, new_ename, new_job, find_employee(new_mgr), new_hiredate, new_s
   END;
END;
/
```

EditEmpProj Class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace EditEmpProj
{
    public class Employee
    {
        public int EmpNo { get; set; }
        public string Ename { get; set; }
        public string Job { get; set; }
        public int? Mgr { get; set; }
        public DateTime Hiredate { get; set; }
        public decimal Sal { get; set; }
        public decimal? Comm { get; set; }
        public int DeptNo { get; set; }

        public Employee(int EmpNo, string Ename, string Job, int? Mgr, DateTime Hiredate
        {
            this.EmpNo = EmpNo;
            this.Ename = Ename;
            this.Job = Job;
            this.Mgr = Mgr;
            this.Hiredate = Hiredate;
            this.Sal = Sal;
            this.Comm = Comm;
            this.DeptNo = DeptNo;
        }
    }
}
```

EditEmpProj DAO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.OracleClient;

namespace EditEmpProj
{
    public class EmployeeDAO
    {
        private string UserName { get; set; }
        private string Password { get; set; }

        public EmployeeDAO(string UserName, string Password)
        {
            this.UserName = UserName;
            this.Password = Password;
        }

        // The code here is used by several of the procedures below.  Note that the
        // SELECT statements used must include the SAME columns that are being read here
        private Employee FillFromDataRow(DataRow dr)
        {
            int empno = Convert.ToInt32(dr["empno"]);
            string ename = Convert.ToString(dr["ename"]);
            string job = Convert.ToString(dr["job"]);
            DateTime hiredate = Convert.ToDateTime(dr["hiredate"]);
            decimal sal = Convert.ToDecimal(dr["sal"]);
            int deptno = Convert.ToInt32(dr["deptno"]);
            int? mgr;
            decimal? comm;
            if (dr.IsNull("mgr"))
                mgr = null;
            else
                mgr = Convert.ToInt32(dr["mgr"]);
            if (dr.IsNull("comm"))
                comm = null;
            else
                comm = Convert.ToDecimal(dr["comm"]);
            return new Employee(empno, ename, job, mgr, hiredate, sal, comm, deptno);
        }

        public List<Employee> FillFromDataTable(DataTable dt)
        {
            List<Employee> employees = new List<Employee>();

            foreach (DataRow dr in dt.Rows)
            {
                Employee employee = FillFromDataRow(dr);
                employees.Add(employee);
            }
            return employees;
        }

        public List<Employee> LoadAll()
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("SELECT empno, ename, job, mgr, hired
            OracleDataAdapter da = new OracleDataAdapter(cmd);
            DataTable dt = new DataTable();

            da.Fill(dt);
            return FillFromDataTable(dt);
        }

        public List<Employee> FindByDepartmentNo(int DeptNo)
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("SELECT empno, ename, job, mgr, hired
            OracleDataAdapter da = new OracleDataAdapter(cmd);
            DataTable dt = new DataTable();

            cmd.Parameters.AddWithValue(":deptno", DeptNo);
            da.Fill(dt);
            return FillFromDataTable(dt);
        }

        public Employee FindByEmployeeNo(int EmpNo)
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("SELECT empno, ename, job, mgr, hired
            OracleDataAdapter da = new OracleDataAdapter(cmd);
            DataTable dt = new DataTable();

            cmd.Parameters.AddWithValue(":empno", EmpNo);
            da.Fill(dt);

            if (dt.Rows.Count > 0)
            {
                DataRow dr = dt.Rows[0];
                Employee employee = FillFromDataRow(dr);
                return employee;
            }
            else
                return null;
        }

        public void SaveEmployee(Employee employee)
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("UPDATE_EMPLOYEE", conn);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("pEmpno", employee.EmpNo);
            cmd.Parameters.AddWithValue("pEname", employee.Ename);
            cmd.Parameters.AddWithValue("pJob", employee.Job);

            if (employee.Mgr.HasValue)
                cmd.Parameters.AddWithValue("pMgr", employee.Mgr.Value);
            else
                cmd.Parameters.AddWithValue("pMgr", DBNull.Value);
            cmd.Parameters.AddWithValue("pHiredate", employee.Hiredate);
            cmd.Parameters.AddWithValue("pSal", employee.Sal);
            if (employee.Comm.HasValue)
                cmd.Parameters.AddWithValue("pComm", employee.Comm.Value);
            else
                cmd.Parameters.AddWithValue("pComm", DBNull.Value);
            cmd.Parameters.AddWithValue("pDeptno", employee.DeptNo);

            conn.Open();
            try
            {
                cmd.ExecuteNonQuery();
            }
            finally
            {
                conn.Close();
            }
        }

        public void CreateEmployee(int EmpNo, string Ename, string Job, string Mname, Da
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("emp_dept_package.create_new_employee"
            OracleCommand cmdFindDept = new OracleCommand("emp_dept_package.find_depart

            cmdFindDept.CommandType = CommandType.StoredProcedure;
            cmdFindDept.Parameters.AddWithValue("dname_to_find", Dname);
            cmdFindDept.Parameters.Add("found_deptno", OracleType.Int32).Direction = Pa

            conn.Open();
            try
            {
                if (null == Loc)
                {
                    cmdFindDept.ExecuteNonQuery();
                    if (Convert.IsDBNull(cmdFindDept.Parameters["found_deptno"].Value))
                        throw new ArgumentException("Invalid Department Location");
                }

                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("new_empno", EmpNo);
                cmd.Parameters.AddWithValue("new_ename", Ename.ToUpper());
                cmd.Parameters.AddWithValue("new_job", Job.ToUpper());

                cmd.Parameters.AddWithValue("new_mgr", Mname.ToUpper()); // A "" value
                cmd.Parameters.AddWithValue("new_hiredate", Hiredate);
                cmd.Parameters.AddWithValue("new_sal", Sal);
                if (Comm.HasValue)
                    cmd.Parameters.AddWithValue("new_comm", Comm.Value);
                else
                    cmd.Parameters.AddWithValue("new_comm", DBNull.Value);
                cmd.Parameters.AddWithValue("new_dept", Dname.ToUpper());
                cmd.Parameters.AddWithValue("new_dept_loc", Loc.ToUpper()); // A "" valu
                cmd.ExecuteNonQuery(); // Transaction is committed automatically (Auto
            }
            finally
            {
                conn.Close();
            }
        }
    }
}
```

Department Class

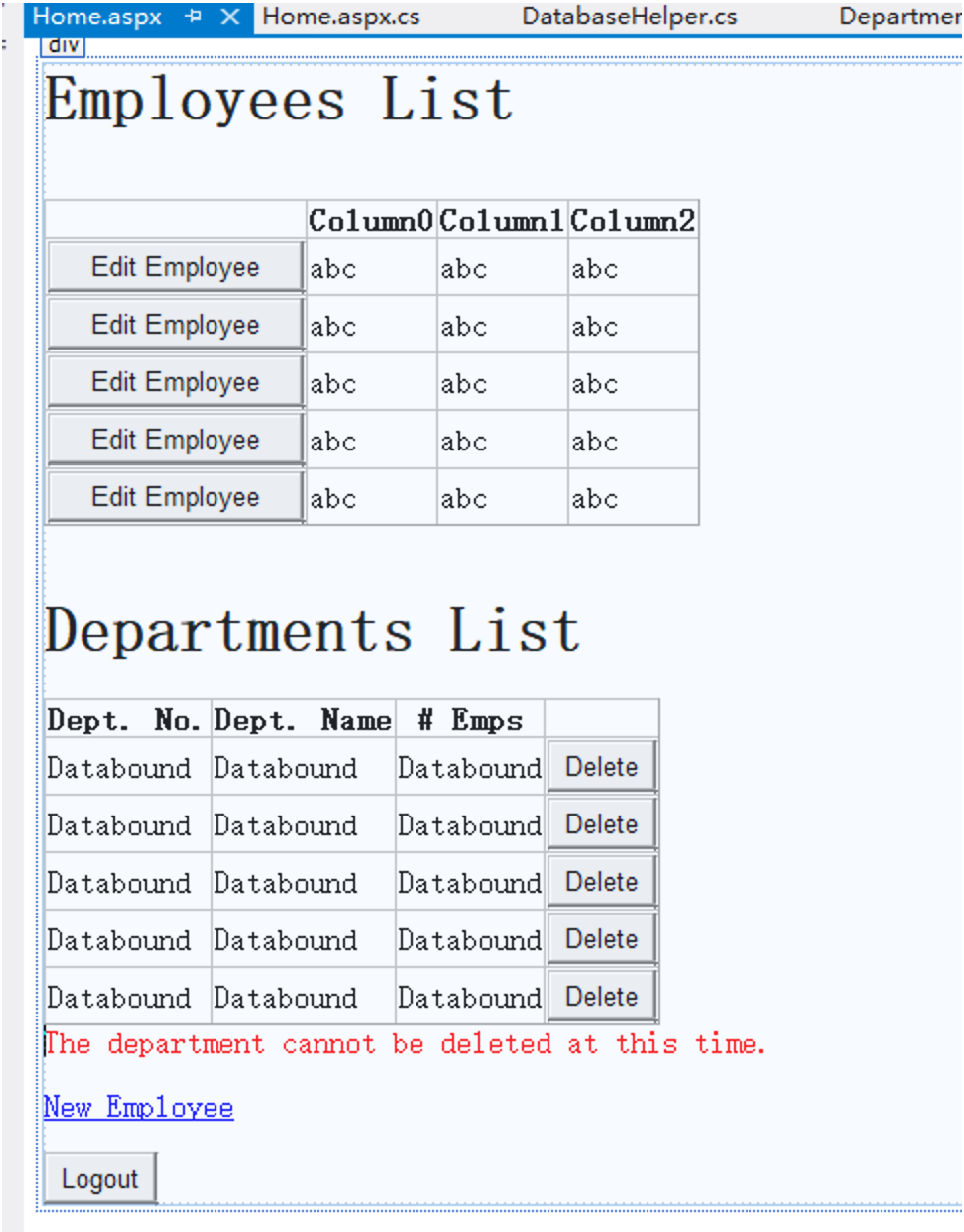
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace EditEmpProj
{
    public class Department
    {
        public int DeptNo { get; set; }
        public string Dname { get; set; }
        public string Loc { get; set; }

        public Department(int Deptno, string Dname, string Loc)
        {
            this.DeptNo = Deptno;
            this.Dname = Dname;
            this.Loc = Loc;
        }
    }
}
```

Departmemt DAO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.OracleClient;

namespace EditEmpProj
{
    public class DepartmentDAO
    {
        private string UserName { get; set; }
        private string Password { get; set; }

        public DepartmentDAO(string UserName, string Password)
        {
            this.UserName = UserName;
            this.Password = Password;
        }

        public List<Department> LoadAll()
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("SELECT deptno, dname, loc FROM dept",
            OracleDataAdapter da = new OracleDataAdapter(cmd);
            DataTable dt = new DataTable();
            List<Department> departments = new List<Department>();

            da.Fill(dt);

            foreach(DataRow dr in dt.Rows)
            {
                int deptno = Convert.ToInt32(dr["deptno"]);
                string dname = Convert.ToString(dr["dname"]);
                string loc = Convert.ToString(dr["loc"]);
                departments.Add(new Department(deptno, dname, loc));
            }
            return departments;
        }

        public List<DepartmentInfo> LoadAllDepartmentInfo()
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmd = new OracleCommand("SELECT deptno, dname, num_emps FROM e
            OracleDataAdapter da = new OracleDataAdapter(cmd);
            DataTable dt = new DataTable();
            List<DepartmentInfo> deptinfos = new List<DepartmentInfo>();

            da.Fill(dt);

            foreach (DataRow dr in dt.Rows)
            {
                int deptno = Convert.ToInt32(dr["deptno"]);
                string dname = Convert.ToString(dr["dname"]);
                int numemps = Convert.ToInt32(dr["num_emps"]);
                deptinfos.Add(new DepartmentInfo(deptno, dname, numemps));
            }
            return deptinfos;
        }

        public bool DeleteDepartment(int DeptNo)
        {
            OracleConnection conn = new OracleConnection(String.Format("Data Source=Nep
            OracleCommand cmdDeleteEmployees = new OracleCommand("DELETE FROM emp WHERE
            OracleCommand cmdDeleteDepartment = new OracleCommand("DELETE FROM dept WHE
            OracleTransaction trans;

            cmdDeleteEmployees.Parameters.AddWithValue(":deptno", DeptNo);
            cmdDeleteDepartment.Parameters.AddWithValue(":deptno", DeptNo);

            // Use transaction processing to make sure employees and department are dele
            // Note that employees in the department cannot be deleted if there is an e
            conn.Open();
            trans = conn.BeginTransaction();
            cmdDeleteEmployees.Transaction = trans;
            cmdDeleteDepartment.Transaction = trans;
            try
            {
                cmdDeleteEmployees.ExecuteNonQuery();
                cmdDeleteDepartment.ExecuteNonQuery();
                trans.Commit();
                return true;
            }
            catch (Exception)
            {
                trans.Rollback();
                return false;
            }
            finally
            {
                if (conn.State == ConnectionState.Open)
                    conn.Close();
            }
        }
    }
}
```

Home Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace EditEmpProj
{
    public partial class Home : BasePage
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                GetEmployeesData();
                GetDepartmentsData();
            }
        }

        protected void gvEmployees_RowCommand(object sender, GridViewCommandEventArgs e
        {
            if ("EDIT" == e.CommandName)
            {
                int index = Convert.ToInt32(e.CommandArgument); // The row of the butto
                int empno = Convert.ToInt32(gvEmployees.Rows[index].Cells[1].Text); // 
                Response.Redirect(String.Format("~/EditEmployee.aspx?empno={0}", empno)
            }
        }

        protected void btnLogout_Click(object sender, EventArgs e)
        {
            Session.Abandon();
            Response.Redirect("~/Home.aspx");
        }

        private void GetEmployeesData()
        {
            LoginInfo login = (LoginInfo)Session["login"];
            EmployeeDAO employeeDAO = new EmployeeDAO(login.UserName, login.Password);

            gvEmployees.DataSource = employeeDAO.LoadAll();
            gvEmployees.DataBind();
        }

        private void GetDepartmentsData()
        {
            LoginInfo login = (LoginInfo)Session["login"];
            DepartmentDAO departmentDAO = new DepartmentDAO(login.UserName, login.Passw

            gvDepartments.DataSource = departmentDAO.LoadAllDepartmentInfo();
            gvDepartments.Columns[0].Visible = true; // Make sure the department number
            gvDepartments.DataBind();
            gvDepartments.Columns[0].Visible = false; // Hide the department number colu
        }

        protected void gvDepartments_RowCommand(object sender, GridViewCommandEventArgs
        {
            if ("DELETE" == e.CommandName)
            {
                LoginInfo login = (LoginInfo)Session["login"];
                DepartmentDAO departmentDAO = new DepartmentDAO(login.UserName, login.Pa

                int index = Convert.ToInt32(e.CommandArgument); // The row of the butto
                int deptno = Convert.ToInt32(gvDepartments.Rows[index].Cells[0].Text); /

                if (departmentDAO.DeleteDepartment(deptno))
                {
                    // Must reload grid data to show employees and department have been
                    GetEmployeesData();
                    GetDepartmentsData();
                }
                else
                    lblNoDelete.Visible = true;
            }
        }

        protected void gvDepartments_RowDeleting(object sender, GridViewDeleteEventArgs
        {

        }
    }
}
```



Employees List

	Column0	Column1	Column2
Edit Employee	abc	abc	abc
Edit Employee	abc	abc	abc
Edit Employee	abc	abc	abc
Edit Employee	abc	abc	abc
Edit Employee	abc	abc	abc

Departments List

Dept. No.	Dept. Name	# Emps	
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete

The department cannot be deleted at this time.

New Employee

Logout

SYS_REFCURSOR

To create a SYS_REFCURSOR, define a SYS_REFCURSOR variable and then OPEN a SELECT statement for that cursor. E.g.

```
my_cursor SYS_REFCURSOR -- In the declarations section
BEGIN
   OPEN my_cursor FOR SELECT empno, ename, sal, job, deptno FROM emp ORDER BY ename ASC
   Return my_cursor;
END;
```

Notice in the above example that the cursor is opened but not closed. The active cursor is returned to the ASP.NET program. There, an OracleDataReader object is created using the cursor to read the results.

```
Dim cmd As New OracleCommand("emp_dept_package.get_employees", conn) ' Where conn is the
Dim dr As OracleDataReader

cmd.CommandType = CommandType.StoredProcedure
cmd.Parameters.Add("emp_cursor", OracleType.Cursor).Direction = ParameterDirection.Retu
conn.Open()
cmd.ExecuteNonQuery()

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
```

A data grid can be filled from the OracleDataReader in much the same way it was filled from the DataTable. That is

```
gv.DataSource = dr ' Where gv is the GridView control
gv.DataBind()
dr.Close() ' Also closes connection thanks to CommandBehaviour.CloseConnection
```

If OUT parameters are used, then multiple cursors can be returned from a stored procedure or function.