

## Actividad evaluable UT2

### Invocar procedimientos almacenados

#### I. Procedimiento sin parámetros: “p1#EmpleadosPorCargo”

##### 1. Descripción

Este procedimiento almacenado devuelve, a través de registros, el número de empleados por cargo.

A través de un *INNER JOIN* unimos la tabla *employee* y la tabla *jobs* a través de su campo común *job\_id*, el cual es una clave foránea (*FK*) de la tabla *employee*. Una vez unidos los registros, los agrupamos a través de un *GROUP BY* por el campo *job\_desc* de la tabla *jobs*. A continuación mostramos este campo junto al número de empleados que hay en dicho campo a través del *COUNT*. Finalmente ordenamos los registros por el campo *job\_desc*.

##### 2. Código Transact-SQL

```
-- numero de empleados por cargo
create procedure p1#EmpleadosPorCargo
as
    select j.job_desc as 'Cargo', count(e.lname) as 'Empleados'
    from employee e
    inner join jobs j on e.job_id = j.job_id
    group by j.job_desc
    order by j.job_desc;
go
```

##### 3. Prueba

```
exec pubs.dbo.p1#EmpleadosPorCargo;
```

| Results |                             | Messages  |
|---------|-----------------------------|-----------|
|         | Cargo                       | Empleados |
| 1       | Acquisitions Manager        | 4         |
| 2       | Business Operations Manager | 1         |
| 3       | Chief Executive Officer     | 1         |
| 4       | Chief Financial Officer     | 1         |
| 5       | Designer                    | 3         |
| 6       | Editor                      | 3         |
| 7       | Managing Editor             | 4         |
| 8       | Marketing Manager           | 4         |
| 9       | Operations Manager          | 4         |
| 10      | Productions Manager         | 4         |
| 11      | Public Relations Manager    | 4         |
| 12      | Publisher                   | 7         |
| 13      | Sales Representative        | 3         |

#### 4. Código VB .Net

**A. Versión 1:** Conectando nuestra aplicación con la base de datos a través de un dsn de archivo (ODBC) y usando un DataReader (Modo Conectado) para almacenar los resultados.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim cn As New OdbcConnection
    cn.ConnectionString = "filedsn=|DataDirectory|\dsnPubs.dsn"
    Try
        cn.Open()
        Dim cmd As OdbcCommand = New OdbcCommand("p1#EmpleadosPorCargo", cn)
        cmd.CommandType = CommandType.StoredProcedure

        Dim dr As OdbcDataReader = cmd.ExecuteReader()
        Dim dt As New DataTable
        dt.Load(dr)
        DataGridView1.DataSource = dt

        dr.Close()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
        cn.Close()
    End Try
End Sub
```

**B. Versión 2:** Haciendo la conexión a través del proveedor de datos de SQL Server (nativo) y usando un DataSet (Modo desconectado)

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim cn As New SqlConnection
    cn.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=pubs;Integrated Security=True"
    Try
        Dim da As SqlDataAdapter = New SqlDataAdapter("p1#EmpleadosPorCargo", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure

        Dim ds As New DataSet
        da.Fill(ds, "EmpleadosCargo")

        DataGridView1.DataSource = ds
        DataGridView1.DataMember = "EmpleadosCargo"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

**Resultado de ejecución:**

|   | Cargo                 | Empleados |
|---|-----------------------|-----------|
| ► | Acquisitions Man...   | 4         |
|   | Business Operati...   | 1         |
|   | Chief Executive ...   | 1         |
|   | Chief Financial Of... | 1         |
|   | Designer              | 3         |
|   | Editor                | 2         |

## II. Procedimiento con parámetro de entrada: “p2#EmpleadoCargo”

### 1. Descripción

Este procedimiento muestra el nombre de un empleado, cuyo id se debe proporcionar, y su cargo.

A través del *INNER JOIN* unimos las tablas *employee* y *jobs* a través de su campo en común *job\_id*, mostrando así sus registros comunes. Y aplicamos un filtro para que, a través del *WHERE*, indiquemos el empleado a través de su id (*emp\_id*).

### 2. Código Transact-SQL

```
create procedure p2#EmpleadoCargo
    @idEmpleado char(9)
as
    select e.lname as 'Empleado', j.job_desc as 'Cargo'
    from employee e
    inner join jobs j on e.job_id = j.job_id
    where e.emp_id = @idEmpleado;
go
```

### 3. Prueba

```
exec pubs.dbo.p2#EmpleadoCargo 'PMA42628M';
```

| Results |          | Messages             |
|---------|----------|----------------------|
|         | Empleado | Cargo                |
| 1       | Accorti  | Sales Representative |

#### 4. Código VB .Net

**A. Versión 1:** Conectando nuestra aplicación con la base de datos a través del proveedor de SQL Server (nativo) y guardando el resultado en un DataSet (modo desconectado).

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim cn As New SqlConnection
    cn.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=pubs;Integrated Security=True"
    Try
        Dim da As SqlDataAdapter = New SqlDataAdapter("p2#EmpleadoCargo", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure

        Dim param As SqlParameter = New SqlParameter("@idEmpleado", SqlDbType.Char, 9)
        param.Direction = ParameterDirection.Input
        param.Value = TextBox1.Text
        da.SelectCommand.Parameters.Add(param)

        Dim ds As New DataSet
        da.Fill(ds, "EmpleadoCargo")

        DataGridView1.DataSource = ds
        DataGridView1.DataMember = "EmpleadoCargo"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

**B. Versión 2:** Haciendo la conexión a través del proveedor de datos de OleDb y guardando el resultado en un DataReader (modo conectado).

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Dim cn As New OleDbConnection
    cn.ConnectionString = "Provider=SQLNCLI11;Data Source=(LocalDB)\MSSQLLocalDB;Integrated Security=SSPI;Initial Catalog=pubs"
    Try
        cn.Open()
        Dim cmd As OleDbCommand = New OleDbCommand("p2#EmpleadoCargo", cn)
        cmd.CommandType = CommandType.StoredProcedure

        Dim param As OleDbParameter = New OleDbParameter("@idEmpleado", OleDbType.Char, 9)
        param.Direction = ParameterDirection.Input
        param.Value = TextBox1.Text
        cmd.Parameters.Add(param)

        Dim dr As OleDbDataReader = cmd.ExecuteReader()
        Dim dt As New DataTable
        dt.Load(dr)
        DataGridView1.DataSource = dt

        dr.Close()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
        cn.Close()
    End Try
End Sub
```

**Resultado de ejecución:**

Invocar procedimiento  
p1#EmpleadosPorCargo

Conectado ODBC Desconectado SQL

Invocar procedimiento  
p2#EmpleadoCargo

ID del Empleado:  
PMA42628M

Desconectado SQL Conectado OleDb

|   | Empleado | Cargo              |
|---|----------|--------------------|
| ▶ | Accorti  | Sales Represent... |
| * |          |                    |

### III. Procedimiento con parámetro de entrada y salida: “p3#EmpleadosEditorial”

#### 1. Descripción

Este procedimiento devuelve, a través de un parámetro de salida, la cantidad de empleados que trabajan para una editorial concreta (proporcionada como parámetro de entrada).

Mediante el INNER JOIN unimos las tablas *employee* y *publishers* a través de su campo común *pub\_id*. Devolvemos la cantidad a través del *COUNT*. Y aplicamos un filtro para que, a través del *WHERE*, indiquemos la editorial a través de su nombre (*pub\_name*).

#### 2. Código Transact-SQL

```
create procedure p3#EmpleadosEditorial
    @nombrePub varchar(40),
    @numEmpleados int output
as
    set @numEmpleados = (
        select count(e.emp_id)
        from employee e
        inner join publishers p on e.pub_id = p.pub_id
        where p.pub_name = @nombrePub);
go
```

#### 3. Prueba

```
declare @num int
exec p3#EmpleadosEditorial 'New Moon Books', @num output
print 'Num de empleados: ' + cast(@num as varchar)
```

Messages

Num de empleados: 10

## 4. Código VB .Net

**A. Versión 1:** Conectando nuestra aplicación con la base de datos a través del proveedor de datos de Ole DB y, empleando el modo conectado, mostrar el valor devuelto en un MsgBox.

```
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    Dim cn As New OleDbConnection
    cn.ConnectionString = "Provider=SQLNCLI11;Data Source=(LocalDB)\MSSQLLocalDB;Integrated Security=SSPI;Initial Catalog=pubs"
    Try
        cn.Open()
        Dim cmd As OleDbCommand = New OleDbCommand("p3#EmpleadosEditorial", cn)
        cmd.CommandType = CommandType.StoredProcedure

        Dim param As OleDbParameter = New OleDbParameter("@nombrePub", OleDbType.VarChar, 40)
        param.Direction = ParameterDirection.Input
        param.Value = TextBox2.Text
        cmd.Parameters.Add(param)

        Dim param2 As OleDbParameter = New OleDbParameter("@numEmpleados", OleDbType.Integer)
        param2.Direction = ParameterDirection.Output
        cmd.Parameters.Add(param2)

        cmd.ExecuteScalar()
        cn.Close()

        Dim num As Integer = cmd.Parameters("@numEmpleados").Value
        MsgBox("La editorial " + TextBox2.Text + " tiene " + num.ToString() + " empleados")
    Catch ex As Exception
        MsgBox(ex.Message)
        cn.Close()
    End Try
End Sub
```

**B. Versión 2:** Haciendo la conexión a través del proveedor de datos de SQL Server y empleando un DataAdapter para ejecutar el método ExecuteScalar(), pero abriendo la conexión ya que este lo requiere.

```
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    Dim cn As New SqlConnection
    cn.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=pubs;Integrated Security=True"
    Try
        Dim da As SqlDataAdapter = New SqlDataAdapter("p3#EmpleadosEditorial", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure

        Dim p1 As SqlParameter = New SqlParameter("@nombrePub", SqlDbType.VarChar, 40)
        p1.Direction = ParameterDirection.Input
        p1.Value = TextBox2.Text
        da.SelectCommand.Parameters.Add(p1)

        Dim p2 As SqlParameter = New SqlParameter("@numEmpleados", SqlDbType.Int)
        p2.Direction = ParameterDirection.Output
        da.SelectCommand.Parameters.Add(p2)

        cn.Open()
        da.SelectCommand.ExecuteScalar()
        cn.Close()

        Dim num As Integer = da.SelectCommand.Parameters("@numEmpleados").Value
        MsgBox("La editorial " + TextBox2.Text + " tiene " + num.ToString() + " empleados")
    Catch ex As Exception
        MsgBox(ex.Message)
        cn.Close()
    End Try
End Sub
```

**Resultado de ejecución:**

