

## **Laboratorio 6: Ejercitación de Procedimientos Almacenados**

Dada la base de datos **AdventureWorks**, realizar los siguientes enunciados:

- 1- **p\_InsCulture(id,name,date)**: Este sp debe permitir dar de alta un nuevo registro en la tabla Production.Culture. Los tipos de datos de los parámetros deben corresponderse con la tabla. Para ayudarse, se podrá ejecutar el procedimiento `sp_help "esquema.objeto"`.
- 2- **p\_SelCulture(id)**: Este sp devolverá el registro completo según el id enviado.
- 3- **p\_DelCulture(id)**: Este sp debe borrar el id enviado por parámetro de la tabla Production.Culture.
- 4- **p\_UpdCulture(id)**: Dado un id debe permitirme cambiar el campo name del registro.
- 5- **sp\_CantCulture (cant out)**: Realizar un sp que devuelva la cantidad de registros en Culture. El resultado deberá colocarlo en una variable de salida.
- 6- **sp\_CultureAsignadas** : Realizar un sp que devuelva solamente las Culture's que estén siendo utilizadas en las tablas (Verificar qué tabla/s la están referenciando). Sólo debemos devolver id y nombre de la Cultura.
- 7- **p\_ValCulture(id,name,date,operación, valida out)**: Este sp permitirá validar los datos enviados por parámetro. En el caso que el registro sea válido devolverá un 1 en el parámetro de salida valida ó 0 en caso contrario. El parámetro operación puede ser "U" (Update), "I" (Insert) ó "D" (Delete).  
Lo que se debe validar es:
  - Si se está insertando no se podrá agregar un registro con un id existente, ya que arrojará un error.
  - Tampoco se puede agregar dos registros Cultura con el mismo Name, ya que el campo Name es un unique index.
  - Ninguno de los campos debería estar vacío.
  - La fecha ingresada no puede ser menor a la fecha actual.
- 8- **p\_SelCulture2(id out, name out, date out)**: A diferencia del sp del punto 2, este debe emitir todos los datos en sus parámetros de salida. ¿Cómo se debe realizar la llamada del sp para testear este sp?
- 9- Realizar una modificación al sp `p_InsCulture` para que valide los registros ingresados. Por lo cual, deberá invocar al sp `p_ValCulture`. Sólo se insertará si la validación es correcta.
- 10- Idem con el sp `p_UpdCulture`. Validar los datos a actualizar.
- 11- En `p_DelCulture` se deberá modificar para que valide que no posea registros relacionados en la tabla que lo referencia. Investigar cuál es la tabla referenciada e incluir esta validación. Si se está utilizando, emitir un mensaje que no se podrá eliminar.
- 12- **p\_CrearCultureHis**: Realizar un sp que permita crear la siguiente tabla histórica de Cultura. Si existe deberá eliminarse. Ejecutar el procedimiento para que se pueda crear:

```
CREATE TABLE Production.CultureHis (  
    CultureID nchar(6) NOT NULL,  
    Name [dbo].[Name] NOT NULL,  
    ModifiedDate datetime NOT NULL CONSTRAINT  
        DF_CultureHis_ModifiedDate DEFAULT (getdate()),  
    CONSTRAINT PK_CultureHis_IDDate PRIMARY KEY CLUSTERED (CultureID,  
    ModifiedDate)  
)
```

- ¿Qué tipo de datos posee asignado el campo Name?
- ¿Qué sucede si no se inserta el campo ModifiedDate?

- 13-Dada la tabla histórica creada en el punto 12, se desea modificar el procedimiento **p\_UpdCulture** creado en el punto 4. La modificación consiste en que cada vez que se cambia algún valor de la tabla Culture se desea enviar el registro anterior a una tabla histórica. De esta forma, en la tabla Culture siempre tendremos el último registro y en la tabla CutureHis cada una de las modificaciones realizadas.
- 14-**p\_UserTables(opcional esquema)**: Realizar un procedimiento que liste las tablas que hayan sido creadas dentro de la base de datos con su nombre, esquema y fecha de creación. En el caso que se ingrese por parámetro el esquema, entonces mostrar únicamente dichas tablas, de lo contrario, mostrar todos los esquemas de la base.
- 15-**p\_GenerarProductoxColor()**: Generar un procedimiento que divida los productos según el color que poseen. Los mismos deben ser insertados en diferentes tablas según el color del producto. Por ejemplo, las tablas podrían ser Product\_Black, Product\_Silver, etc... Estas tablas deben ser generadas dinámicamente según los colores que existan en los productos, es decir, si genero un nuevo producto con un nuevo color, al ejecutar el procedimiento debe generar dicho color. Cada vez que se ejecute este procedimiento se recrearán las tablas de colores. Los productos que no posean color asignados, no se tendrán en cuenta para la generación de tablas y no se insertarán en ninguna tabla de color.
- 16-**p\_UltimoProducto(param)**: Realizar un procedimiento que devuelva en sus parámetros (output), el último producto ingresado.
- 17-**p\_TotalVentas(fecha)**: Realizar un procedimiento que devuelva el total facturado en un día dado. El procedimiento, simplemente debe devolver el total monetario de lo facturado (Sales).