



ANDREA NOELIA DESCOUS (31)



Time to complete: 38:16

Aprobado 7 (siete)

1. Seleccione la sintaxis correcta de los bloques de manejo de error correctos en un procedimiento almacenado:

5 / 5 pts

☐ TRY..END / CATCH..END☐ BEGIN..END / ERROR..END☒ BEGIN TRY..END TRY / BEGIN CATCH..END CATCH☐ Ninguna es correcta.

2. ¿Cuál función de SQL devuelve el número de error de la última sentencia ejecutada con error en la ejecución de un procedimiento?

5 / 5 pts

☐ @@ERROR☐ ISERROR()☐ ERROR_CODE()☒ ERROR_NUMBER()

3. La ejecución de la sentencia THROW 51000, 'Mensaje de error.', 1; en el bloque TRY de un procedimiento almacenado:

5 / 5 pts

☒ Genera una excepción y transfiere la ejecución a un bloque CATCH.☐ Genera una excepción y transfiere la ejecución a la aplicación que realiza la llamada.☐ Genera solo una excepción.☐ Ninguna de las anteriores.

4. Al ejecutar el siguiente código:

5 / 5 pts

```
DECLARE @DESCRIPCIONERROR VARCHAR(MAX)

SELECT @DESCRIPCIONERROR = 'Id inexistente'

IF NOT EXISTS (SELECT 1 FROM EMPLEADO WHERE
ID=2)

BEGIN

THROW 51000,@DESCRIPCIONERROR,1;

END
```

☐ Se genera el mensaje de error 'Id inexistente' si en la tabla empleado existe el empleado con ID con valor 2.

☒ Se genera el mensaje de error 'Id inexistente' si en la tabla empleado no existe el empleado con ID con valor 2. ✓

☐ Se genera el mensaje de error 'Id inexistente' si en la tabla empleado solo existe un empleado con ID con valor 2.

☐ Ninguna es correcta.

5. Considerando la siguiente cabecera del procedimiento SQL y los tipos de parámetros, indicar cuál es la llamada adecuada:

5 / 5 pts

```
CREATE PROCEDURE [dbo].[SP_EMPLEADO_ABM]
```

```
@ACCION CHAR(1),@ID NUMERIC(18,0),  
@NOMBRE VARCHAR(100),@APELLIDO VARCHAR(100),  
@RDO VARCHAR(MAX) OUTPUT
```

AS

☒ EXEC [dbo].[SP_EMPLEADO_ABM] 'A', 1,'nombre','apellido',
@RDO OUT ✓

☐ EXEC [dbo].[SP_EMPLEADO_ABM] 'A', 1,'nombre','apellido',
@RDO

☐ EXEC [dbo].[SP_EMPLEADO_ABM] 'A', 1,'nombre','apellido',
'Resultado'

☐ EXEC [dbo].[SP_EMPLEADO_ABM] 'A', 1,'nombre', @RDO
OUTPUT

6. Para consignar permanentemente los cambios de una transacción implícita o explícita se debe utilizar:

5 / 5 pts

☐ THROW

☐ TRIGGER

☐ ROLLBACK TRANSACTION

☒ COMMIT TRANSACTION ✓

☐ Ninguna es correcta.

7. ¿Cuál de las siguientes declaraciones de procedimientos almacenados posee un parámetro llamado Codigo de tipo INT opcional con valor predeterminado nulo?

5 / 5 pts

☐ ALTER PROC TarjetasCredito @Codigo INT DEFAULT NULL AS [..]

☐ ALTER PROC TarjetasCredito @Codigo INT IS NULL AS [..]

☒ ALTER PROC TarjetasCredito @Codigo INT=NULL AS [..] ✓

☐ Ninguna es correcta.

8. El funcionamiento del manejo de errores con el bloque TRY..CATCH permite que:

5 / 5 pts

☐ Si se produce un error en el bloque TRY, el control se transfiere a la capa superior.

☒ Si se produce un error en el bloque TRY, el control se transfiere al grupo de instrucciones del bloque CATCH. ✓

☐ Si se produce un error en el bloque CATCH, el control se transfiere al grupo de instrucciones del bloque TRY.

9. El tipo de Trigger de tipo INSTEAD OF se ejecuta:

5 / 5 pts

☐ Luego de la operación o de las operaciones para la cual se declara.

☐ Antes de la operación o de las operaciones para la cual se declara.

☒ En lugar de la operación o de las operaciones para la cual se declara. ✓

☐ Ninguna es correcta.

10. Un Trigger o disparador se crea:

5 / 5 pts

- ☐ Asociado a una tabla específica para uno o varios eventos de selección, inserción, eliminación y/o actualización.
- ☒ Asociado a una tabla específica para eventos de inserción, eliminación y/o actualización según como se configure. ✓
- ☐ Asociado a una tabla específica y es independiente de los eventos de inserción, eliminación y/o actualización.
- ☐ Ninguna es correcta.

11. Los Triggers:

0 / 5 pts

- ☒ Se ejecutan manualmente al intentar modificar los datos de una tabla para la que se ha definido el disparador. ✗
- ☐ Se ejecutan automáticamente al realizar una acción sobre los datos de la tabla para la que se ha definido el disparador. ✓
- ☐ Se ejecutan de forma automática o manual, según como se configure el disparador.
- ☐ Ninguna es correcta.

12. Cuáles son las tablas especiales que se crean en el Trigger ante una operación de UPDATE:

5 / 5 pts

- ☐ UPDATED
- ☐ INSERTED o DELETED según exista el registro a actualizar
- ☒ INSERTED y DELETED ✓
- ☐ Ninguna es correcta.

13. Para lograr configurar bajas lógicas sobre los registros de una tabla (y evitar la baja física de los mismos), se deben desarrollar:

0 / 5 pts

- ☐ Triggers del tipo INSTEAD OF sobre la operación de DELETE. ✓
- ☐ Triggers del tipo AFTER sobre la operación de DELETE.
- ☒ Restricciones de DEFAULTS en el campo que refleja la baja del registro. ✗
- ☐ Ninguna es correcta.

14. El siguiente código:

0 / 5 pts

```
SELECT @DESCRIPCIONERROR = 'Nombre inválido'

IF @NOMBRE IS NULL OR TRIM(@NOMBRE)=''

BEGIN

    THROW 51000,@DESCRIPCIONERROR,1;

END
```

- ☒ Valida que la variable @NOMBRE no sea nula ni sea una cadena vacía. ✕
- ☐ El código es incorrecto. ✓
- ☐ Valida que la variable @NOMBRE no sea nula.

15. El siguiente código en un Trigger asociado a INSERT y UPDATE:

0 / 5 pts

```
IF EXISTS(SELECT 1 FROM DELETED)

SET @Operation = 'M'

ELSE

SET @Operation = 'A'
```

- ☐ El código es incorrecto.
- ☐ Sirve para diferenciar el tipo de acción que disparó el Trigger. ✓
- ☒ Sirve para detectar si la acción se ejecutó correctamente. ✕

16. La siguiente afirmación:

0 / 5 pts

Un CURSOR LIBERA SU MEMORIA CON LA CLAUSULA

☐ OPEN <nombre_cursor>

☒ CLOSE <nombre cursor>



☐ DEALLOCATE <nombre_cursor>



☐ Un cursor no necesita una cláusula para liberar memoria.

17. ¿Qué isolation permite acceder a los datos sin efectuar ningún tipo de lockeo? Ejemplifique

5 / 10 pts

El READ UNCOMMITTED es el menos restrictivo , omite los bloqueos.

Faltó ejemplificar.

18. Declare y programe un cursor para la tabla Production.Product de la base de datos AdventureWorks y muestre por pantalla el valor de los registros ProductID, Name y ProductNumber.

4 / 10 pts

Declare c_prueba CURSOR FOR SELECT ProductID, Name, ProductNumber from Production.Product

open c_prueba

CLOSE c_prueba

DEALLOCATE c_prueba

No usás variables, no imprimis en pantalla ni ciclas con while para ir por cada registro. No cumplís con la expectativa del requerimiento.