NOMBRE: \_\_\_\_\_\_Sánchez Ramírez Noel Adán\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. GRUPO: \_\_3TV2\_

Calificación: \_\_\_\_\_\_\_\_\_\_\_\_ FECHA: 25 de enero de 2021

**INSTRUCCIONES:** RESOLVER TODOS LOS REACTIVOS Y PROBLEMAS.  
 Escribir de forma clara y concisa (indicar procedimiento y resultado).  
 (el examen consta de 2 páginas).

FAVOR DE APAGAR EL CELULAR O COMPUTADORA, **DE NO ATENDER LA INDICACIÓN SE LE CANCELARA EL EXAMEN Y SU CALIFICACIÓN SERA DE CERO.**

1. Crear un Login llamado Fernando, el cual debe ser mapeado a la cuenta de usuarios, asignarle una contraseña

CREATE LOGIN [Fernando] WITH PASSWORD=N'123456'

CREATE USER usuarios

FOR LOGIN Fernando

1. El usuario Fernando debe tener autorización para consultar el esquema base\_upiita

CREATE SCHEMA base\_upiita AUTHORIZATION Usuarios;

1. Escriba las sentencias necesarias para cambiar el archivo Log de una base de datos distribuidas  
     
   ALTER DATABASE [UPIITA\_BASE] MODIFY FILE (NAME = UPIITA\_BASE, FILENAME = 'E:\SQLData\UPIITA\_BASE.mdf');  
     
   ALTER DATABASE tempdb MODIFY FILE (NAME = UPIITA\_BASELOG, FILENAME = 'E:\SQLData\UPIITA\_BASELOG.ldf');
2. Escriba las sentencias necesarias para redimensionar el crecimiento de la base de datos distribuidas  
     
   ALTER DATABASE [UPIITA\_BASE] MODIFY FILE ( NAME = N'UPIITA\_BASE', SIZE = 16384KB, MAXSIZE = 204800KB , FILEGROWTH = 131072KB )  
     
   ALTER DATABASE [UPIITA\_BASE] MODIFY FILE ( NAME = N'UPIITA\_BASE\_log', SIZE = 16384KB, FILEGROWTH = 131072KB )
3. ¿Cuál es la sentencia para REVOCAR los permisos de administrador al usuario: user\_conta  
   Permisos expuestos uno por uno  
   REVOKE ALL user\_conta FROM [UPIITA\_BASE];  
   Visto desde un rol.  
   ALTER ROLE [db\_ddladmin] DROP MEMBER user\_conta;  
   ALTER ROLE [db\_accessadmin] DROP MEMBER user\_conta;
4. Como se le asignarían permisos de SOLO LECTURA al usuario: user\_audit  
   Con esquema dbo

use [UPIITA\_BASE]  
GO

GRANT SELECT ON [dbo].[sysdiagrams] TO [usuarios]

Sin esquema dbo

use [UPIITA\_BASE]  
GO

GRANT SELECT ON [base\_upiita].[sysdiagrams] TO [usuarios]

O para cualquier cosa

USE [UPIITA\_BASE]; ALTER ROLE [db\_datareader] ADD MEMBER [usuarios];

1. Cree un procedimiento que reciba como parámetro el nombre de usuario, contraseña, base de datos y una opción:

1 -> El usuario tendrá permisos de administrador

2 -> El usuario tendrá permisos de sólo lectura

3 -> El usuario tendrá permisos de lectura y escritura

Este procedimiento, debe, con base en la opción recibida crear el login y el usuario solicitado y dar los permisos señalados en la base que recibió como parámetro.

CREATE OR ALTER PROCEDURE SP\_ROLES @USUARIO NVARCHAR(20), @PASS NVARCHAR(20), @BASE NVARCHAR(20), @OPCION INT

AS

BEGIN

DECLARE @QUERY NVARCHAR(MAX) = 'USE ' + @BASE + '; CREATE LOGIN ' + @USUARIO + ' WITH PASSWORD =''' + @PASS + '''; CREATE USER ' + @USUARIO + ' FOR LOGIN ' + @USUARIO + ' WITH DEFAULT\_SCHEMA = base\_upiita;';

IF @OPCION = 1

BEGIN

SET @QUERY = @QUERY + ' ALTER ROLE db\_ddladmin ADD MEMBER ' + @USUARIO + ';';

END

ELSE IF @OPCION = 2

BEGIN

SET @QUERY = @QUERY + ' ALTER ROLE db\_datareader ADD MEMBER ' + @USUARIO + ';';

END

ELSE IF @OPCION = 3

BEGIN

SET @QUERY = @QUERY + ' ALTER ROLE db\_datareader ADD MEMBER ' + @USUARIO + '; ALTER ROLE db\_datawriter ADD MEMBER ' + @USUARIO + ';';

END

PRINT @QUERY;

EXEC(@QUERY);

END

EXEC SP\_ROLES 'P','1234','UPIITA\_BASE',3

1. Se tiene un sistema, en el que se añadirá un control de usuarios que genere los usuarios a las bases de datos y los correspondientes logins, en la ventana de registro se solicitará:
2. Solicitud: Alta, Baja, Cambio en permisos
3. Nombre de Usuario
4. Password asignado
5. Nombre de base de datos
6. Permisos asignados

* Crear la o las tablas que recibirán dicha información, deberá incluir una tabla Bitacora, la cual será la última en llenarse.
* CREATE DATABASE [EXAMEN];
* USE [EXAMEN]
* GO
* -- Tablas
* CREATE TABLE CAT\_USUARIOS\_TYPE(
* CAT\_UST\_ID INT PRIMARY KEY,
* CAT\_UST\_DESC NVARCHAR(20)
* );
* CREATE TABLE CAT\_USUARIOS(
* CAT\_USU\_ID INT PRIMARY KEY,
* CAT\_USU\_NOMBRE NVARCHAR(20) NOT NULL,
* CAT\_USU\_PASS NVARCHAR(10) NOT NULL,
* CAT\_USU\_TYPE INT NOT NULL DEFAULT 2,
* CONSTRAINT FK\_USUARIOS\_TYPE FOREIGN KEY (CAT\_USU\_TYPE) REFERENCES CAT\_USUARIOS\_TYPE(CAT\_UST\_ID)
* );
* CREATE TABLE HIST\_REGISTROS\_BITACORA(
* HIST\_REG\_ID INT PRIMARY KEY,
* HIST\_USUARIO\_ID INT DEFAULT 1,
* HIST\_REG\_PERMISOS NVARCHAR(MAX),
* HIST\_REG\_BASE NVARCHAR(500) DEFAULT 'EXAMEN',
* HIST\_REG\_INIT DATETIME DEFAULT GETDATE(),
* HIST\_REG\_END DATETIME,
* CONSTRAINT FK\_USUARIOS FOREIGN KEY (HIST\_USUARIO\_ID) REFERENCES CAT\_USUARIOS(CAT\_USU\_ID)
* );
* Crear un trigger en la última tabla, Bitacora, el cual debe realizar la petición del usuario (Crear usuario, agregar o quitar permisos, etc)
* CREATE OR ALTER TRIGGER TG\_PERMISOS
* ON
* HIST\_REGISTROS\_BITACORA
* FOR INSERT
* AS
* BEGIN
* DECLARE @ID\_USUARIO INT;
* DECLARE @USUARIO NVARCHAR(30);
* DECLARE @PERMISOS NVARCHAR(20);
* DECLARE @BASE NVARCHAR(20);
* DECLARE @QUERY NVARCHAR(MAX);
* DECLARE @EVENTO NVARCHAR(20);
* SET @ID\_USUARIO = (SELECT HIST\_USUARIO\_ID FROM inserted);
* SET @USUARIO = (SELECT CAT\_USU\_NOMBRE FROM CAT\_USUARIOS);
* SET @BASE = (SELECT HIST\_REG\_BASE FROM inserted);
* SET @QUERY = 'USE ' + @BASE + '; ' + @PERMISOS + ' ON SCHEMA ::[dbo] to ' + @USUARIO + ';';
* EXEC(@QUERY);
* SET @EVENTO = 'EXITOSO ' + @USUARIO;
* INSERT INTO HIST\_AUDITORIA(HIST\_AUDI\_ID, HIST\_EVENTO)VALUES((SELECT ISNULL(MAX(HIST\_AUDI\_ID),0) + 1 FROM HIST\_AUDITORIA), @EVENTO)
* END
* El resultado (Exitoso, no exitoso) deberá registrarse en esa misma tabla.
* Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

  Descripción generada automáticamente
* Interfaz de usuario gráfica, Texto, Aplicación

  Descripción generada automáticamente
* Texto

  Descripción generada automáticamente

1. Crear un script que cree una base de datos con tres tablas y tres usuarios:

a) Administrador

b) Consultor

c) Analista

El usuario Administrador tendrá todos los privilegios, Consultor solo tendrá permisos de SELECT y Analista solo podrá realizar INSERT, UPDATE y DELETE.

CREATE DATABASE [EXAMEN];

USE [EXAMEN]

GO

-- Tablas

CREATE TABLE CAT\_USUARIOS\_TYPE(

CAT\_UST\_ID INT PRIMARY KEY,

CAT\_UST\_DESC NVARCHAR(20)

);

CREATE TABLE CAT\_USUARIOS(

CAT\_USU\_ID INT PRIMARY KEY,

CAT\_USU\_NOMBRE NVARCHAR(20) NOT NULL,

CAT\_USU\_PASS NVARCHAR(10) NOT NULL,

CAT\_USU\_TYPE INT NOT NULL DEFAULT 2,

CONSTRAINT FK\_USUARIOS\_TYPE FOREIGN KEY (CAT\_USU\_TYPE) REFERENCES CAT\_USUARIOS\_TYPE(CAT\_UST\_ID)

);

CREATE TABLE HIST\_REGISTROS\_BITACORA(

HIST\_REG\_ID INT PRIMARY KEY,

HIST\_USUARIO\_ID INT DEFAULT 1,

HIST\_REG\_PERMISOS NVARCHAR(MAX),

HIST\_REG\_BASE NVARCHAR(500) DEFAULT 'EXAMEN',

HIST\_REG\_INIT DATETIME DEFAULT GETDATE(),

HIST\_REG\_END DATETIME,

CONSTRAINT FK\_USUARIOS FOREIGN KEY (HIST\_USUARIO\_ID) REFERENCES CAT\_USUARIOS(CAT\_USU\_ID)

);

-- Usuario y logins

USE [EXAMEN]

CREATE LOGIN ADMINISTRADOR WITH PASSWORD='PASSWORD'

CREATE USER ADMINISTRADOR FOR LOGIN ADMINISTRADOR WITH

DEFAULT\_SCHEMA= ESQUEMA\_ADMINISTRADOR

CREATE LOGIN CONSULTOR WITH PASSWORD='PASSWORD'

CREATE USER CONSULTOR FOR LOGIN CONSULTOR WITH

DEFAULT\_SCHEMA= ESQUEMA\_CONSULTOR

CREATE LOGIN ANALISTA WITH PASSWORD='PASSWORD'

CREATE USER ANALISTA FOR LOGIN ANALISTA WITH

DEFAULT\_SCHEMA= ESQUEMA\_ANALISTA

-- Permisos

GRANT INSERT ON SCHEMA :: [dbo] to ADMINISTRADOR

GRANT SELECT ON SCHEMA :: [dbo] to ADMINISTRADOR

GRANT UPDATE ON SCHEMA :: [dbo] to ADMINISTRADOR

GRANT DELETE ON SCHEMA :: [dbo] to ADMINISTRADOR

GRANT SELECT ON SCHEMA :: [dbo] to CONSULTOR

GRANT INSERT ON SCHEMA :: [dbo] to ANALISTA

GRANT UPDATE ON SCHEMA :: [dbo] to ANALISTA

GRANT DELETE ON SCHEMA :: [dbo] to ANALISTA

* Escribir la sentencia para agregarle el permiso de insertar al Consultor

GRANT INSERT ON SCHEMA :: [dbo] to CONSULTOR

* Escribir la sentencia para quitarle el permiso de eliminar al Analista

REVOKE DELETE ON SCHEMA:: [dbo] to ANALISTA

1. Agregar una columna a la tabla Bitacora del ejercicio 4 llamada Folio, el cual se alimentará por una secuencia (SecBitac) que comenzará en 1 e irá incrementando en 1. Con un valor máximo de 999. (Nota: En caso de ejecutar la secuencia de forma externa a la tabla, ese valor se “ocupará”, es decir, la tabla lleva el folio 10, pero ejecutamos la secuencia fuera de la tabla y nos da el valor 11, al hacer un nuevo registro en la tabla deberá mostrar el folio 12)

ALTER TABLE HIST\_REGISTROS\_BITACORA ADD HIST\_REG\_FOLIO INTEGER DEFAULT 1;

CREATE SEQUENCE dbo.SECBITAC

AS INT

START WITH 1

INCREMENT BY 1

MINVALUE 1 MAXVALUE 999

CYCLE;

1. Crear una consulta a la tabla BITACORA con el usuario Analista, mostrar el mensaje enviado por SQL y posterior a eso crear la sentencia necesaria para llevar a cabo la consulta sin que muestre el mensaje de error.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente