

- 1) Implemente y explique un Script para crear una **vista** para crear utilizando tres tablas

```
CREATE VIEW ResumenVentas AS
SELECT
    v.ID,
    c.Nombre AS NombreCliente,
    p.Nombre AS NombreProducto,
    p.Precio,
    v.FechaVenta,
    v.MontoTotal
FROM Ventas v
JOIN Clientes c ON v.ClienteID = c.ID
JOIN Productos p ON v.ProductoID = p.ID;
```

- 2) Implemente y explique un Script para crear un **procedimiento almacenado** para insertar datos a su base de datos.

```
CREATE PROCEDURE NuevaVenta
    @Cliente INT,
    @Producto INT,
    @Cantidad INT
AS
BEGIN
    DECLARE @PrecioTotal DECIMAL(10,2);
    SET @PrecioTotal = @Cantidad * (SELECT Precio FROM Productos WHERE ID = @Producto);

    INSERT INTO Ventas VALUES (@Cliente, @Producto, @Cantidad, @PrecioTotal, GETDATE());
    UPDATE Productos SET Stock = Stock - @Cantidad WHERE ID = @Producto;
END
```

- 3) Implemente y explique un Script para crear un **procedimiento almacenado** para eliminar datos a su base de datos

```
CREATE PROCEDURE BorrarCliente
    @ID INT
AS
BEGIN
    DELETE FROM Ventas WHERE ClienteID = @ID;
    DELETE FROM Clientes WHERE ID = @ID;
END
```

- 4) Implemente y explique un Script para crear un **procedimiento almacenado** para actualizar datos a su base de datos

```

CREATE PROCEDURE ActualizarInventario
    @ID INT,
    @NuevoPrecio DECIMAL(10,2),
    @NuevoStock INT
AS
BEGIN
    UPDATE Productos
    SET Precio = @NuevoPrecio,
        Stock = @NuevoStock
    WHERE ID = @ID;
END

```

- 5) Implemente y explique un Script para crear un **procedimiento almacenado** para realizar cálculos matemáticos de una columna de su base de datos.

```

CREATE PROCEDURE CalcularVentasMensuales
    @Mes INT,
    @Año INT
AS
BEGIN
    SELECT
        SUM(MontoTotal) AS Total,
        COUNT(*) AS NumeroVentas,
        AVG(MontoTotal) AS PromedioVenta
    FROM Ventas
    WHERE MONTH(FechaVenta) = @Mes
    AND YEAR(FechaVenta) = @Año;
END

```

- 6) Implemente y explique un Script para crear un **disparador** para ingresar un registro automáticamente en una tabla de su base de datos.

```

CREATE TRIGGER RegistroNuevoProducto
ON Productos
AFTER INSERT
AS
BEGIN
    INSERT INTO LogProductos (Accion, Fecha, ProductoID)
    SELECT 'Nuevo Producto', GETDATE(), ID
    FROM inserted;
END

```

- 7) Implemente y explique un Script para crear un **disparador** para elimine un registro automáticamente en una tabla de su base de datos.

```
CREATE TRIGGER LimpiezaVentas
ON Clientes
AFTER DELETE
AS
BEGIN
    DELETE FROM Ventas
    WHERE ClienteID IN (SELECT ID FROM deleted);
END
```

- 8) Implemente y explique un Script para crear un **disparador** para actualice un registro automáticamente en una tabla de su base de datos.

```
CREATE TRIGGER ActualizarInventarioVenta
ON Ventas
AFTER INSERT
AS
BEGIN
    UPDATE Productos
    SET Stock = Stock - i.Cantidad
    FROM Productos p
    JOIN inserted i ON p.ID = i.ProductoID;
END
```

- 9) Implemente y explique un Script para crear un **disparador** para verificar el control de datos (Ejemplo: que la nota ingresada este entre 0 y 20)

```
CREATE TRIGGER ValidarVenta
ON Ventas
AFTER INSERT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE MontoTotal <= 0)
    BEGIN
        ROLLBACK;
        RAISERROR('El monto debe ser positivo', 16, 1);
    END
END
```

- 10) Utilizando Script Crear 03 usuarios con nombres de sus compañeros y uno suyo

```
CREATE LOGIN alberto WITH PASSWORD = 'Pass123';
CREATE LOGIN maria WITH PASSWORD = 'Pass456';
CREATE LOGIN carlos WITH PASSWORD = 'Pass789';
```

- 11) Utilizando un script, copiar la base de datos (creada anteriormente) y compartir en cada uno de los usuarios

```
USE MiBaseDatos;
CREATE USER alberto FOR LOGIN alberto;
CREATE USER maria FOR LOGIN maria;
CREATE USER carlos FOR LOGIN carlos;
```

- 12) Utilizando un script, generar una copia de seguridad de la base de datos y compartir a cada uno de los usuarios

```
BACKUP DATABASE MiBaseDatos
TO DISK = 'C:\Backups\backup_completo.bak'
WITH DESCRIPTION = 'Backup completo';
```

- 13) Utilizando un script, encriptar una de las tablas para que no se puedan ver los datos

```
-- Crear clave maestra
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'MiClaveSegura123';

-- Crear certificado
CREATE CERTIFICATE CertSeguridad
WITH SUBJECT = 'Certificado para datos sensibles';

-- Encriptar columna
ALTER TABLE Clientes
ADD TelefonoEncriptado varbinary(256);
```

- 14) Utilizando un script, aplique la seguridad a nivel de columna, restringiendo el acceso a la columna que contiene la clave primaria de una de las tablas de su base de datos

```
CREATE ROLE AccesoLimitado;
DENY SELECT ON Clientes(DNI) TO AccesoLimitado;
```

- 15) Utilizando un script, implementé seguridad a nivel de columna restringiendo el acceso a una de las columnas de una tabla.

```
DENY SELECT ON Empleados(Salario) TO AccesoLimitado;
```

- 16) Utilizando un script, realice el cifrado transparente de datos (TDE) para una las tablas.

```
-- Requiere versión Enterprise
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE CertSeguridad;
```

- 17) Utilizando un script, configure el usuario con el nombre de su compañero para otorgar permisos de SELECT, INSERT, UPDATE y DELETE en la base de datos.

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON Productos
TO maria;
```

- 18) Utilizando un Scripts realice la validación y filtración de entradas del usuario para evitar caracteres maliciosos (Ejemplo: ', --, ;)

```
CREATE PROCEDURE ValidarTexto
    @Texto varchar(100)
AS
BEGIN
    IF @Texto LIKE '%[^a-zA-Z0-9 ]%'
        RETURN 'Caracteres no permitidos';
    RETURN 'OK';
END
```

- 19) Realice un script que verifiquen que los datos ingresados cumplan con formatos esperados (ej.: números en lugar de texto, longitud máxima).

```
CREATE PROCEDURE ValidarTelefono
    @Telefono varchar(15)
AS
BEGIN
    IF @Telefono LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
        RETURN 'Válido';
    RETURN 'Inválido';
END
```

- 20) Utilizando un script, configure la auditoría para el seguimiento y registro de acciones en la base de datos

```
CREATE SERVER AUDIT MiAuditoria
TO FILE (FILEPATH = 'C:\Logs\');

CREATE DATABASE AUDIT SPECIFICATION AuditoriaTablas
FOR SERVER AUDIT MiAuditoria
ADD (SELECT ON Clientes BY PUBLIC);
```

- 21) Utilizando un script, configure de la memoria y el disco duro

```
EXEC sp_configure 'max server memory', 4096; -- 4GB
EXEC sp_configure 'min server memory', 1024; -- 1GB
```

- 22) Utilizando un script, genere una copia de seguridad de la base de datos

```
BACKUP DATABASE MiBaseDatos
TO DISK = 'C:\Backups\backup_diario.bak';
```

- 23) Realice un script para programar backups automatizados de su base de datos

```
-- Crear trabajo programado
CREATE PROCEDURE BackupDiario
AS
BEGIN
    BACKUP DATABASE MiBaseDatos
    TO DISK = 'C:\Backups\auto_' +
    CONVERT(varchar(8), GETDATE(), 112) + '.bak';
END
```

- 24) Utilizando un script, genere la restauración de la base de datos

```
RESTORE DATABASE MiBaseDatos
FROM DISK = 'C:\Backups\backup_diario.bak'
WITH REPLACE;
```

- 25) Utilizando un script, cree un espejo de la base de datos

```
-- En servidor principal
ALTER DATABASE MiBaseDatos
SET PARTNER = 'TCP://ServidorEspejo:5022';
```

- 26) Utilizando un script, para enviar datos a la base de datos espejo creada

```
-- Verificar estado
SELECT mirroring_state_desc
FROM sys.database_mirroring
WHERE database_id = DB_ID('MiBaseDatos');
```

- 27) Utilizando un script, de permiso a un usuario por un determinado tiempo

```
CREATE PROCEDURE AccesoTemporal
    @Usuario varchar(50)
AS
BEGIN
    GRANT SELECT ON Ventas TO @Usuario;
    WAITFOR DELAY '01:00:00';
    REVOKE SELECT ON Ventas FROM @Usuario;
END
```

- 28) Utilizando un script, realice la replicación de bases de datos

```
-- Configurar publicador
EXEC sp_replicationdboption
    @dbname = 'MiBaseDatos',
    @optname = 'publish',
    @value = 'true';
```

- 29) Explique que es Always On Availability Groups

Es como tener varias copias de tu base de datos en diferentes servidores que trabajan juntas. Si uno falla, otro toma el control automáticamente, como tener un plan B siempre listo.

- 30) Explique que es Log Shipping

Es como hacer fotocopias del diario de la base de datos y enviarlas a otro servidor regularmente.
Si el servidor principal falla, tienes una copia casi actualizada en otro lugar.