Práctica de Administración de Bases de Datos con SQL Server

Objetivo

Desarrollar habilidades prácticas en la administración de bases de datos SQL Server, incluyendo creación, configuración, seguridad, mantenimiento y optimización.

Ejercicio 1: Creación y Configuración de Base de Datos

Instrucciones:

1. Crear una base de datos llamada "PracticaAdminDB" con:
    - Archivo de datos principal de 25MB con crecimiento de 5MB (máximo 100MB)
    - Archivo de log de 10MB con crecimiento de 2MB (máximo 50MB)
2. Configurar las siguientes opciones:
    - Modelo de recuperación: Simple
    - Desactivar Auto-Shrink
    - Activar creación y actualización automática de estadísticas
3. Crear un grupo de archivos secundario llamado "FG_SECUNDARIO" con un archivo de 15MB

Solución:

```
-- 1. Crear base de datos
CREATE DATABASE PracticaAdminDB
ON PRIMARY
(
    NAME = 'PracticaAdminDB_Data',
    FILENAME = 'C:\Data\PracticaAdminDB.mdf',
    SIZE = 25MB,
    MAXSIZE = 100MB,
```

```
    FILEGROWTH = 5MB
)
LOG ON
(
    NAME = 'PracticaAdminDB_Log',
    FILENAME = 'C:\Data\PracticaAdminDB.ldf',
    SIZE = 10MB,
    MAXSIZE = 50MB,
    FILEGROWTH = 2MB
);
GO

-- 2. Configurar opciones
ALTER DATABASE PracticaAdminDB SET RECOVERY SIMPLE;
ALTER DATABASE PracticaAdminDB SET AUTO_SHRINK OFF;
ALTER DATABASE PracticaAdminDB SET AUTO_CREATE_STATISTICS ON;
ALTER DATABASE PracticaAdminDB SET AUTO_UPDATE_STATISTICS ON;
GO

-- 3. Crear grupo de archivos secundario
ALTER DATABASE PracticaAdminDB
ADD FILEGROUP FG_SECUNDARIO;
GO

ALTER DATABASE PracticaAdminDB
ADD FILE
(
    NAME = 'PracticaAdminDB_Secundario',
```

```
    FILENAME = 'C:\Data\PracticaAdminDB_Secondary.ndf',
    SIZE = 15MB,
    MAXSIZE = 50MB,
    FILEGROWTH = 5MB
)
TO FILEGROUP FG_SECUNDARIO;
GO
```

Ejercicio 2: Administración de Seguridad

Instrucciones:

1. Crear los siguientes roles de base de datos:
    - ✓ RolLectura (permisos de solo lectura)
    - ✓ RolEscritura (permisos de escritura)
    - ✓ RolAdminDatos (permisos completos)
2. Crear tres usuarios asociados a logins y asignarlos a los roles:
    - ✓ Usuario1 → RolLectura
    - ✓ Usuario2 → RolEscritura
    - ✓ AdminDatos → RolAdminDatos
3. Configurar permisos a nivel de esquema para cada rol
4. Crear un usuario "Auditor" con permisos para ver definiciones y leer datos

Solución:

```
-- 1. Crear roles
USE PracticaAdminDB;
GO

CREATE ROLE RolLectura;
CREATE ROLE RolEscritura;
```

```
CREATE ROLE RolAdminDatos;
GO

-- 2. Crear usuarios y asignar roles
CREATE LOGIN Usuario1 WITH PASSWORD = 'P@ssw0rd1';
CREATE USER Usuario1 FOR LOGIN Usuario1;
ALTER ROLE RolLectura ADD MEMBER Usuario1;

CREATE LOGIN Usuario2 WITH PASSWORD = 'P@ssw0rd2';
CREATE USER Usuario2 FOR LOGIN Usuario2;
ALTER ROLE RolEscritura ADD MEMBER Usuario2;

CREATE LOGIN AdminDatos WITH PASSWORD = 'Adm1nD@t0s';
CREATE USER AdminDatos FOR LOGIN AdminDatos;
ALTER ROLE RolAdminDatos ADD MEMBER AdminDatos;
GO

-- 3. Configurar permisos
GRANT SELECT ON SCHEMA::dbo TO RolLectura;
GRANT INSERT, UPDATE, DELETE ON SCHEMA::dbo TO RolEscritura;
GRANT CONTROL ON SCHEMA::dbo TO RolAdminDatos;
GO

-- 4. Crear usuario Auditor
CREATE LOGIN Auditor WITH PASSWORD = 'Aud1t0r2023';
CREATE USER Auditor FOR LOGIN Auditor;
GRANT VIEW DEFINITION ON DATABASE::PracticaAdminDB TO Auditor;
GRANT SELECT ON SCHEMA::dbo TO Auditor;
```

GO

Ejercicio 3: Mantenimiento y Optimización

Instrucciones:

1. Crear una tabla "Clientes" con:
   - Campos: ClienteID (PK autoincremental), Nombre, Apellido, Email (único), FechaRegistro (default), Activo (default)
   - Índice no agrupado en Apellido
2. Insertar 3 registros de prueba
3. Crear un procedimiento almacenado "sp_InsertarCliente" con manejo de transacciones y errores
4. Crear un índice filtrado para clientes activos
5. Actualizar estadísticas de la tabla

```
-- 1. Crear tabla
CREATE TABLE Clientes
(
    ClienteID INT IDENTITY(1,1) PRIMARY KEY,
    Nombre NVARCHAR(100) NOT NULL,
    Apellido NVARCHAR(100) NOT NULL,
    Email NVARCHAR(255) UNIQUE,
    FechaRegistro DATETIME DEFAULT GETDATE(),
    Activo BIT DEFAULT 1,
    INDEX IX_Clientes_Apellido NONCLUSTERED (Apellido)
);
GO

-- 2. Insertar datos
```

```
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES
('Juan', 'Pérez', 'juan.perez@email.com'),
('María', 'Gómez', 'maria.gomez@email.com'),
('Carlos', 'López', 'carlos.lopez@email.com');
GO

-- 3. Crear procedimiento
CREATE PROCEDURE sp_InsertarCliente
    @Nombre NVARCHAR(100),
    @Apellido NVARCHAR(100),
    @Email NVARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        INSERT INTO Clientes (Nombre, Apellido, Email)
        VALUES (@Nombre, @Apellido, @Email);

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
```

```
        THROW;
    END CATCH
END;
GO

-- 4. Índice filtrado
CREATE INDEX IX_Clientes_Activos ON Clientes(ClienteID)
WHERE Activo = 1;
GO

-- 5. Actualizar estadísticas
UPDATE STATISTICS Clientes WITH FULLSCAN;
GO
```

Ejercicio 4: Copias de Seguridad y Recuperación

Instrucciones:

1. Realizar un backup completo de la base de datos con compresión
2. Insertar un nuevo cliente y realizar un backup diferencial
3. Insertar otro cliente y realizar un backup del log de transacciones
4. Simular un proceso de recuperación:
    - ✓ Restaurar el backup completo en una nueva base de datos "PracticaAdminDB_Test"
    - ✓ Aplicar el backup diferencial

Solución:

```
-- 1. Backup completo
BACKUP DATABASE PracticaAdminDB
TO DISK = 'C:\Backups\PracticaAdminDB_Full.bak'
```

```
WITH INIT, COMPRESSION, STATS = 10;
GO

-- 2. Backup diferencial
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Ana', 'Martínez', 'ana.martinez@email.com');
GO

BACKUP DATABASE PracticaAdminDB
TO DISK = 'C:\Backups\PracticaAdminDB_Diff.bak'
WITH DIFFERENTIAL, COMPRESSION, STATS = 10;
GO

-- 3. Backup de log
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Pedro', 'Sánchez', 'pedro.sanchez@email.com');
GO

BACKUP LOG PracticaAdminDB
TO DISK = 'C:\Backups\PracticaAdminDB_Log.trn'
WITH COMPRESSION, STATS = 10;
GO

-- 4. Recuperación
RESTORE DATABASE PracticaAdminDB_Test
FROM DISK = 'C:\Backups\PracticaAdminDB_Full.bak'
WITH
   MOVE 'PracticaAdminDB_Data' TO 'C:\Data\PracticaAdminDB_Test.mdf',
```

```
    MOVE 'PracticaAdminDB_Log' TO 'C:\Data\PracticaAdminDB_Test.ldf',
    MOVE 'PracticaAdminDB_Secundario' TO 'C:\Data\PracticaAdminDB_Test_Secondary.ndf',
    REPLACE, STATS = 10, NORECOVERY;
GO

RESTORE DATABASE PracticaAdminDB_Test
FROM DISK = 'C:\Backups\PracticaAdminDB_Diff.bak'
WITH RECOVERY, STATS = 10;
GO
```

Ejercicio 5: Monitoreo y Resolución de Problemas

Instrucciones:

1. Consultar el uso de espacio de la tabla Clientes
2. Identificar las 10 consultas con mayor uso de recursos lógicos
3. Verificar bloqueos actuales en la base de datos
4. Monitorear el uso de memoria (ratio de aciertos de buffer y esperanza de vida de páginas)

Solución:

```
-- 1. Uso de espacio
EXEC sp_spaceused 'Clientes';
GO

-- 2. Consultas costosas
SELECT TOP 10
    qs.execution_count,
    qs.total_logical_reads/qs.execution_count AS avg_logical_reads,
    qs.total_elapsed_time/qs.execution_count AS avg_elapsed_time,
    SUBSTRING(qt.text, (qs.statement_start_offset/2)+1,
```

```
        ((CASE qs.statement_end_offset
          WHEN -1 THEN DATALENGTH(qt.text)
         ELSE qs.statement_end_offset
         END - qs.statement_start_offset)/2)+1) AS query_text,
    qt.dbid,
    qt.objectid
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
ORDER BY qs.total_logical_reads DESC;
GO

-- 3. Bloqueos
SELECT
    request_session_id AS spid,
    resource_type,
    resource_database_id AS dbid,
    request_mode,
    request_status
FROM sys.dm_tran_locks
WHERE resource_database_id = DB_ID('PracticaAdminDB');
GO

-- 4. Memoria
SELECT
    object_name,
    counter_name,
    cntr_value
FROM sys.dm_os_performance_counters
```

```
WHERE counter_name IN ('Buffer cache hit ratio', 'Page life expectancy')
AND object_name LIKE '%Buffer Manager%';
GO
```

Tareas Adicionales

1. Configurar un trabajo del Agente SQL para realizar backups automáticos cada noche.
2. Crear un plan de mantenimiento que incluya reorganización de índices y actualización de estadísticas.

Evaluación

- Correcta creación y configuración de la base de datos (20%)
- Implementación adecuada de seguridad y roles (20%)
- Estrategia de mantenimiento y optimización (20%)
- Plan de copias de seguridad y recuperación (20%)
- Monitoreo y resolución de problemas (20%)