Nombre de la práctica	Práctica de Administración de Bases de Datos con SQL Server			No.	
Asignatura:	Administración de Bases de datos	Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	2 horas

GRUPO: 3601
NOMBRE: Vanesa Hernández Martínez

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

-- 1. Crear base de datos con archivo principal de datos (.mdf) y archivo de registro (.ldf)

```
CREATE DATABASE PracticaAdminDB

ON PRIMARY

(

    -- Definimos el archivo de datos principal

    NAME = 'PracticaAdminDB_Data',  -- Nombre lógico del archivo de datos

    FILENAME = 'C:\Data\PracticaAdminDB.mdf',  -- Ruta física donde se almacenará el archivo .mdf

    SIZE = 25MB,  -- Tamaño inicial del archivo

    MAXSIZE = 100MB,  -- Tamaño máximo que puede alcanzar

    FILEGROWTH = 5MB  -- Crecimiento incremental del archivo cuando se llena

)

LOG ON

(

    -- Definimos el archivo de registro de transacciones

    NAME = 'PracticaAdminDB_Log',  -- Nombre lógico del archivo de log
```

```
    FILENAME = 'C:\Data\PracticaAdminDB.ldf',  -- Ruta física del archivo .ldf
    SIZE = 10MB,  -- Tamaño inicial del log
    MAXSIZE = 50MB,  -- Tamaño máximo del log
    FILEGROWTH = 2MB  -- Crecimiento incremental del log
);
GO
```

```
-- 2. Configurar opciones de la base de datos para optimizar comportamiento y rendimiento
ALTER DATABASE PracticaAdminDB SET RECOVERY SIMPLE;  -- Modo de recuperación simple (no conserva el log extensivamente)
ALTER DATABASE PracticaAdminDB SET AUTO_SHRINK OFF;  -- Desactiva el auto-encogimiento de archivos para evitar fragmentación
ALTER DATABASE PracticaAdminDB SET AUTO_CREATE_STATISTICS ON;  -- Habilita creación automática de estadísticas para el optimizador
ALTER DATABASE PracticaAdminDB SET AUTO_UPDATE_STATISTICS ON;  -- Permite la actualización automática de estadísticas
GO
```

```
-- 3. Crear un grupo de archivos secundario para distribuir datos en múltiples archivos físicos
ALTER DATABASE PracticaAdminDB
ADD FILEGROUP FG_SECUNDARIO;  -- Crea un nuevo grupo de archivos llamado FG_SECUNDARIO
GO
```

```
-- 4. Agregar un archivo de datos al grupo de archivos secundario
ALTER DATABASE PracticaAdminDB
```

```
ADD FILE
(
    NAME = 'PracticaAdminDB_Secundario',  -- Nombre lógico del nuevo archivo
    FILENAME = 'C:\Data\PracticaAdminDB_Secondary.ndf',  -- Ruta física del archivo secundario
    SIZE = 15MB,  -- Tamaño inicial del archivo secundario
    MAXSIZE = 50MB,  -- Tamaño máximo que puede alcanzar
    FILEGROWTH = 5MB  -- Crecimiento incremental cuando se llena
)
TO FILEGROUP FG_SECUNDARIO;  -- Se asigna el archivo al grupo FG_SECUNDARIO
GO
```

-- Ejercicio 2: Administración de Seguridad

-- 1. Crear roles

```
USE PracticaAdminDB;  -- Cambia el contexto a la base de datos 'PracticaAdminDB' para ejecutar el resto de los comandos en ella
GO

-- Creamos tres roles personalizados para agrupar permisos según el tipo de usuario
CREATE ROLE RolLectura;       -- Rol para usuarios que solo pueden leer (SELECT)
CREATE ROLE RolEscritura;     -- Rol para usuarios que pueden insertar, actualizar y eliminar datos
CREATE ROLE RolAdminDatos;    -- Rol con permisos más amplios sobre los objetos de datos
GO
```

-- 2. Crear usuarios y asignar roles

```
-- Usuario1: solo lectura

CREATE LOGIN Usuario1 WITH PASSWORD = 'P@ssw0rd1';  -- Crea un inicio de sesión (a nivel de servidor) con contraseña

CREATE USER Usuario1 FOR LOGIN Usuario1;            -- Crea el usuario dentro de la base de datos asociado al login

ALTER ROLE RolLectura ADD MEMBER Usuario1;          -- Asigna el usuario al rol de lectura


-- Usuario2: escritura (insertar, actualizar, eliminar)

CREATE LOGIN Usuario2 WITH PASSWORD = 'P@ssw0rd2';  -- Login para el segundo usuario

CREATE USER Usuario2 FOR LOGIN Usuario2;            -- Usuario correspondiente en la base de datos

ALTER ROLE RolEscritura ADD MEMBER Usuario2;        -- Se agrega al rol de escritura


-- AdminDatos: control total sobre los datos del esquema

CREATE LOGIN AdminDatos WITH PASSWORD = 'Adm1nD@t0s';  -- Login para un usuario administrador de datos

CREATE USER AdminDatos FOR LOGIN AdminDatos;          -- Usuario en la base de datos

ALTER ROLE RolAdminDatos ADD MEMBER AdminDatos;       -- Se asigna al rol administrativo

GO


-- 3. Configurar permisos específicos a los roles sobre el esquema 'dbo'

-- Al rol de lectura le damos permiso solo para leer datos del esquema 'dbo'

GRANT SELECT ON SCHEMA::dbo TO RolLectura;


-- Al rol de escritura le damos permiso para insertar, modificar y borrar datos en el esquema 'dbo'
```

GRANT INSERT, UPDATE, DELETE ON SCHEMA::dbo TO RolEscritura;

-- Al rol administrativo le damos CONTROL, que es el permiso más amplio (puede hacer de todo en el esquema)

GRANT CONTROL ON SCHEMA::dbo TO RolAdminDatos;

GO

-- 4. Crear usuario Auditor

-- Creamos un usuario con fines de auditoría: puede ver definiciones y consultar datos, pero no modificarlos

CREATE LOGIN Auditor WITH PASSWORD = 'Aud1t0r2023'; -- Login para el auditor

CREATE USER Auditor FOR LOGIN Auditor; -- Usuario en la base de datos

-- Permiso para ver la definición de los objetos (tablas, procedimientos, etc.), útil para auditoría

GRANT VIEW DEFINITION ON DATABASE::PracticaAdminDB TO Auditor;

-- Permiso para leer datos del esquema 'dbo'

GRANT SELECT ON SCHEMA::dbo TO Auditor;

GO

-- Ejercicio 3: Mantenimiento y Optimización

-- 1. Crear tabla

```
CREATE TABLE Clientes  -- Crea una nueva tabla llamada 'Clientes'
(
    ClienteID INT IDENTITY(1,1) PRIMARY KEY,  -- Columna de clave primaria con autoincremento (comienza en 1 y aumenta de 1 en 1)
    Nombre NVARCHAR(100) NOT NULL,           -- Nombre del cliente, campo obligatorio
    Apellido NVARCHAR(100) NOT NULL,         -- Apellido del cliente, campo obligatorio
    Email NVARCHAR(255) UNIQUE,              -- Dirección de correo electrónico, debe ser única en la tabla
    FechaRegistro DATETIME DEFAULT GETDATE(), -- Fecha y hora del registro, se asigna automáticamente la fecha actual
    Activo BIT DEFAULT 1,                    -- Indica si el cliente está activo (1 = sí, 0 = no), por defecto está activo
    INDEX IX_Clientes_Apellido NONCLUSTERED (Apellido) -- Crea un índice no agrupado en la columna Apellido para mejorar búsquedas
);
GO
```

-- 2. Insertar datos de ejemplo en la tabla Clientes

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

```
-- 3. Crear procedimiento almacenado para insertar un nuevo cliente
CREATE PROCEDURE sp_InsertarCliente
    @Nombre NVARCHAR(100),        -- Parámetro de entrada para el nombre
    @Apellido NVARCHAR(100),      -- Parámetro de entrada para el apellido
    @Email NVARCHAR(255)          -- Parámetro de entrada para el email
AS
BEGIN
    SET NOCOUNT ON;  -- Evita que SQL Server retorne el número de filas afectadas; mejora el rendimiento


    BEGIN TRY
        BEGIN TRANSACTION;  -- Inicia una transacción explícita


        -- Inserta los datos en la tabla Clientes utilizando los parámetros recibidos
        INSERT INTO Clientes (Nombre, Apellido, Email)
        VALUES (@Nombre, @Apellido, @Email);


        COMMIT TRANSACTION;  -- Si todo sale bien, confirma la transacción
    END TRY
    BEGIN CATCH
        -- Si ocurre un error y la transacción sigue activa, se revierte
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
```

```
        THROW;  -- Re-lanza el error para que sea manejado por el sistema o el cliente
    END CATCH
END;
GO
```

-- 4. Crear un índice filtrado que solo incluye clientes activos

```
CREATE INDEX IX_Clientes_Activos ON Clientes(ClienteID)
WHERE Activo = 1;  -- El índice se aplica solo a los registros donde Activo = 1, útil para consultas frecuentes sobre clientes activos
GO
```

-- 5. Actualizar estadísticas de la tabla Clientes

```
UPDATE STATISTICS Clientes WITH FULLSCAN;
-- Refresca las estadísticas de distribución de datos de la tabla 'Clientes' con un escaneo completo.
-- Esto ayuda al optimizador de consultas a generar planes más eficientes
GO
```

-- Ejercicio 4: Copias de Seguridad y Recuperación

-- 1. Backup completo

```
BACKUP DATABASE PracticaAdminDB  -- Realiza una copia de seguridad completa de la base de datos 'PracticaAdminDB'
TO DISK = 'C:\Backups\PracticaAdminDB_Full.bak'  -- Define la ubicación y el nombre del archivo de respaldo (.bak)
WITH
```

```
    INIT,        -- Sobrescribe el archivo de respaldo si ya existe (inicializa)
    COMPRESSION,  -- Comprime el respaldo para ahorrar espacio
    STATS = 10;   -- Muestra el progreso cada 10% durante la ejecución
GO


-- 2. Backup diferencial


-- Antes del backup diferencial, insertamos un nuevo registro en la tabla para generar cambios desde el último backup completo
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Ana', 'Martínez', 'ana.martinez@email.com');  -- Añade un nuevo cliente llamado Ana Martínez
GO

-- Backup diferencial que almacena solo los cambios realizados desde el último backup completo
BACKUP DATABASE PracticaAdminDB
TO DISK = 'C:\Backups\PracticaAdminDB_Diff.bak'  -- Archivo donde se guarda el respaldo diferencial
WITH
    DIFFERENTIAL,  -- Indica que es un respaldo diferencial (solo los cambios desde el backup completo más reciente)
    COMPRESSION,   -- Comprime el archivo
    STATS = 10;    -- Muestra progreso del respaldo cada 10%
GO
```

```
-- 3. Backup de log

-- Insertamos otro registro para generar más actividad antes de respaldar el log de transacciones
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Pedro', 'Sánchez', 'pedro.sanchez@email.com');  -- Añade al cliente Pedro Sánchez
GO

-- Realiza un respaldo del log de transacciones para conservar todas las operaciones desde el último backup de log
BACKUP LOG PracticaAdminDB
TO DISK = 'C:\Backups\PracticaAdminDB_Log.trn'  -- El archivo .trn guarda el log de transacciones
WITH
    COMPRESSION,  -- Comprime el archivo de log
    STATS = 10;   -- Muestra el progreso del respaldo cada 10%
GO

-- 4. Recuperación

-- Restaura la base de datos a partir del backup completo, pero sin finalizar la recuperación (NORECOVERY permite aplicar más backups después)
RESTORE DATABASE PracticaAdminDB_Test  -- Nombre de la nueva base de datos restaurada (para pruebas en este caso)
FROM DISK = 'C:\Backups\PracticaAdminDB_Full.bak'  -- Se usa el respaldo completo como base
WITH
    MOVE 'PracticaAdminDB_Data' TO 'C:\Data\PracticaAdminDB_Test.mdf',  -- Ruta para el
```

archivo de datos restaurado

MOVE 'PracticaAdminDB_Log' TO 'C:\Data\PracticaAdminDB_Test.ldf', -- Ruta para el archivo de log restaurado

MOVE 'PracticaAdminDB_Secundario' TO 'C:\Data\PracticaAdminDB_Test_Secondary.ndf', -- Ruta para el archivo del grupo secundario

REPLACE, -- Reemplaza la base de datos si ya existe

STATS = 10, -- Muestra progreso

NORECOVERY; -- Deja la base en estado de recuperación, lista para aplicar otro backup (como un diferencial o de log)

GO

-- Aplica el backup diferencial a la base restaurada para actualizarla hasta ese punto

RESTORE DATABASE PracticaAdminDB_Test

FROM DISK = 'C:\Backups\PracticaAdminDB_Diff.bak' -- Se usa el respaldo diferencial

WITH

RECOVERY, -- Finaliza la recuperación y deja la base operativa

STATS = 10; -- Muestra el progreso

GO

-- Ejercicio 5: Monitoreo y Resolución de Problemas

-- 1. Uso de espacio

EXEC sp_spaceused 'Clientes';

-- Ejecuta el procedimiento almacenado 'sp_spaceused' para mostrar estadísticas de espacio (tamaño total, datos, índice y espacio no usado)

-- Específicamente, lo hace sobre la tabla 'Clientes'

GO

```
-- 2. Consultas costosas

SELECT TOP 10  -- Selecciona las 10 consultas más costosas según uso de recursos
    qs.execution_count,  -- Cantidad de veces que la consulta ha sido ejecutada
    qs.total_logical_reads / qs.execution_count AS avg_logical_reads,
    -- Promedio de lecturas lógicas por ejecución (indicador de uso de CPU/memoria)

    qs.total_elapsed_time / qs.execution_count AS avg_elapsed_time,
    -- Promedio de tiempo total de ejecución por consulta (en microsegundos)

    SUBSTRING(qt.text, (qs.statement_start_offset/2)+1,
    ((CASE qs.statement_end_offset
        WHEN -1 THEN DATALENGTH(qt.text)  -- Si no hay fin definido, se toma toda la longitud del texto
        ELSE qs.statement_end_offset
    END - qs.statement_start_offset)/2)+1) AS query_text,
    -- Extrae el texto SQL exacto de la consulta

    qt.dbid,       -- ID de la base de datos donde se ejecutó
    qt.objectid    -- ID del objeto (procedimiento, función, etc.) si aplica
FROM sys.dm_exec_query_stats AS qs  -- Vista del sistema con estadísticas de ejecución acumuladas para cada consulta
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
-- Se usa CROSS APPLY para unir cada fila de estadísticas con el texto SQL correspondiente
ORDER BY qs.total_logical_reads DESC;
```

-- Ordena los resultados por el mayor número total de lecturas lógicas (es decir, las más costosas en lectura de datos)

GO

-- 3. Bloqueos

```
SELECT
    request_session_id AS spid,  -- ID de la sesión que tiene el bloqueo (process ID)
    resource_type,               -- Tipo de recurso bloqueado (página, fila, objeto, etc.)
    resource_database_id AS dbid,-- ID de la base de datos donde está ocurriendo el bloqueo
    request_mode,                -- Tipo de bloqueo solicitado (por ejemplo, Shared, Exclusive)
    request_status               -- Estado del bloqueo (por ejemplo, GRANT, WAIT)
FROM sys.dm_tran_locks           -- Vista dinámica que muestra los bloqueos actuales en el sistema
WHERE resource_database_id = DB_ID('PracticaAdminDB');
-- Filtra para mostrar solo los bloqueos dentro de la base de datos 'PracticaAdminDB'

GO
```

-- 4. Memoria

```
SELECT
    object_name,     -- Nombre del objeto relacionado con el contador de rendimiento
    counter_name,    -- Nombre del contador (por ejemplo, Buffer cache hit ratio)
    cntr_value       -- Valor actual del contador
```

```
FROM sys.dm_os_performance_counters  -- Vista que expone métricas de rendimiento en tiempo real

WHERE counter_name IN ('Buffer cache hit ratio', 'Page life expectancy')

-- Filtra solo dos contadores clave para analizar uso de memoria

-- 'Buffer cache hit ratio' mide la eficiencia del caché (cuánto se lee de memoria vs. disco)

-- 'Page life expectancy' indica cuánto tiempo viven las páginas en memoria (un indicador crítico de presión de memoria)


AND object_name LIKE '%Buffer Manager%';

-- Se asegura de tomar solo los contadores del componente "Buffer Manager"


GO
```