

Nombre de la práctica	Práctica de Administración de Bases de Datos con SQL Server			No.	1
Asignatura:	Administración de Base de Datos	Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	3 horas

GRUPO: 3601

NOMBRE: Shania Kinnereth Diaz Moya

-- EJERCICIO 1

-- 1. Crear una nueva base de datos llamada "PracticaAdminDB"

CREATE DATABASE PracticaAdminDB

ON PRIMARY -- Define el grupo de archivos primario donde estará el archivo principal de datos

(
NAME = 'PracticaAdminDB_Data', -- Nombre lógico del archivo de datos
FILENAME = 'C:\Data\PracticaAdminDB.mdf', -- Ruta física del archivo de datos
SIZE = 25MB, -- Tamaño inicial del archivo de datos
MAXSIZE = 100MB, -- Tamaño máximo del archivo de datos
FILEGROWTH = 5MB -- Incremento de tamaño cuando el archivo de datos se llena
)

LOG ON -- Define el archivo de log para las transacciones de la base de datos

(
NAME = 'PracticaAdminDB_Log', -- Nombre lógico del archivo de log
FILENAME = 'C:\Data\PracticaAdminDB.ldf', -- Ruta física del archivo de log
SIZE = 10MB, -- Tamaño inicial del archivo de log
MAXSIZE = 50MB, -- Tamaño máximo del archivo de log
FILEGROWTH = 2MB -- Incremento del archivo de log cuando se llena
);

GO

-- 2. Configurar algunas opciones de comportamiento de la base de datos

```
ALTER DATABASE PracticaAdminDB SET RECOVERY SIMPLE;
```

-- Establece el modelo de recuperación de la base de datos como "Simple", lo que significa que solo se guardan los cambios más recientes de la base de datos, sin hacer un seguimiento detallado de las transacciones.

```
ALTER DATABASE PracticaAdminDB SET AUTO_SHRINK OFF;
```

-- Desactiva la opción de "Auto-shrink", que reduce el tamaño de los archivos de la base de datos cuando se libera espacio. Esto podría afectar el rendimiento si se activa.

```
ALTER DATABASE PracticaAdminDB SET AUTO_CREATE_STATISTICS ON;
```

-- Permite que SQL Server cree estadísticas automáticamente para optimizar las consultas en la base de datos.

```
ALTER DATABASE PracticaAdminDB SET AUTO_UPDATE_STATISTICS ON;
```

-- Permite que SQL Server actualice automáticamente las estadísticas cuando detecte cambios en los datos, lo que mejora el rendimiento de las consultas.

GO

-- 3. Crear un grupo de archivos secundario para distribuir los datos

```
ALTER DATABASE PracticaAdminDB
```

```
ADD FILEGROUP FG_SECUNDARIO3;
```

-- Agrega un nuevo grupo de archivos llamado "FG_SECUNDARIO3" a la base de datos para distribuir los datos en

múltiples grupos de archivos.

GO

```
ALTER DATABASE PracticaAdminDB
```

```
ADD FILE
```

```
(
```

```
    NAME = 'PracticaAdminDB_Secundario3', -- Nombre lógico del nuevo archivo de datos
```

```
    FILENAME = 'C:\Data\PracticaAdminDB_Secondary3.ndf', -- Ruta física del archivo
```

```
    SIZE = 15MB, -- Tamaño inicial del archivo
```

```
    MAXSIZE = 50MB, -- Tamaño máximo del archivo
```

```
    FILEGROWTH = 5MB -- Incremento de tamaño cuando el archivo se llena
```

```
)
```

```
TO FILEGROUP FG_SECUNDARIO3;
```

-- Agrega un nuevo archivo de datos al grupo de archivos "FG_SECUNDARIO3" para distribuir la carga y mejorar el rendimiento.

GO

-- EJERCICIO 2

-- 1. Crear roles

USE PracticaAdminDB; -- Selecciona la base de datos PracticaAdminDB para ejecutar las siguientes instrucciones

GO

```
CREATE ROLE RolLectura; -- Crea un rol para permisos de solo lectura
```

CREATE ROLE RolEscritura; -- Crea un rol para permisos de escritura (INSERT, UPDATE, DELETE)

CREATE ROLE RolAdminDatos; -- Crea un rol para permisos administrativos sobre datos

GO

-- 2. Crear usuarios y asignar roles

CREATE LOGIN Usuario1 WITH PASSWORD = 'P@ssw0rd1';

-- Crea un login llamado "Usuario1" con una contraseña especificada

CREATE USER Usuario1 FOR LOGIN Usuario1;

-- Crea un usuario en la base de datos asociado al login "Usuario1"

ALTER ROLE RolLectura ADD MEMBER Usuario1;

-- Asigna el usuario "Usuario1" al rol "RolLectura" para que tenga permisos de solo lectura

CREATE LOGIN Usuario2 WITH PASSWORD = 'P@ssw0rd2';

-- Crea un login llamado "Usuario2" con una contraseña especificada

CREATE USER Usuario2 FOR LOGIN Usuario2;

-- Crea un usuario en la base de datos asociado al login "Usuario2"

ALTER ROLE RolEscritura ADD MEMBER Usuario2;

-- Asigna el usuario "Usuario2" al rol "RolEscritura" para que tenga permisos de escritura

CREATE LOGIN AdminDatos WITH PASSWORD = 'Adm1nD@t0s';

-- Crea un login llamado "AdminDatos" con una contraseña especificada

```
CREATE USER AdminDatos FOR LOGIN AdminDatos;
```

```
-- Crea un usuario en la base de datos asociado al login "AdminDatos"
```

```
ALTER ROLE RolAdminDatos ADD MEMBER AdminDatos;
```

```
-- Asigna el usuario "AdminDatos" al rol "RolAdminDatos" para que tenga permisos administrativos
```

```
GO
```

```
-- 3. Configurar permisos
```

```
GRANT SELECT ON SCHEMA::dbo TO RolLectura;
```

```
-- Otorga permisos de solo lectura (SELECT) sobre el esquema "dbo" al rol "RolLectura"
```

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

```
-- Otorga permisos de escritura (INSERT, UPDATE, DELETE) sobre el esquema "dbo" al rol "RolEscritura"
```

```
GRANT CONTROL ON SCHEMA::dbo TO RolAdminDatos;
```

```
-- Otorga control total sobre el esquema "dbo" al rol "RolAdminDatos"
```

```
GO
```

```
-- 4. Crear usuario Auditor
```

```
CREATE LOGIN Auditor WITH PASSWORD = 'Aud1t0r2023';
```

```
-- Crea un login para el usuario Auditor
```

CREATE USER Auditor FOR LOGIN Auditor;

-- Crea un usuario en la base de datos asociado al login "Auditor"

GRANT VIEW DEFINITION ON DATABASE::PracticaAdminDB TO Auditor;

-- Permite al Auditor ver la definición de los objetos en la base de datos

GRANT SELECT ON SCHEMA::dbo TO Auditor;

-- Permite al Auditor realizar consultas (SELECT) sobre el esquema "dbo"

GO

-- EJERCICIO 3

-- 1. Crear tabla

CREATE TABLE Clientes -- Crea una nueva tabla llamada "Clientes"

(

 ClienteID INT IDENTITY(1,1) PRIMARY KEY, -- Columna para el ID del cliente, autoincremental, clave primaria

 Nombre NVARCHAR(100) NOT NULL, -- Columna para el nombre del cliente, no permite valores nulos

 Apellido NVARCHAR(100) NOT NULL, -- Columna para el apellido, no permite valores nulos

 Email NVARCHAR(255) UNIQUE, -- Columna para el correo electrónico, debe ser único

 FechaRegistro DATETIME DEFAULT GETDATE(), -- Columna para la fecha de registro, valor predeterminado es la fecha y hora actuales

 Activo BIT DEFAULT 1, -- Columna booleana para indicar si el cliente está activo, valor predeterminado es 1 (activo)

 INDEX IX_Clientes_Apellido NONCLUSTERED (Apellido) -- Crea un índice no agrupado sobre la columna Apellido

);

GO

-- 2. Insertar datos

INSERT INTO Clientes (Nombre, Apellido, Email) -- Inserta un nuevo registro en la tabla "Clientes"

VALUES

('Juan', 'Pérez', 'juan.perez@email.com'), -- Primer cliente

('María', 'Gómez', 'maria.gomez@email.com'), -- Segundo cliente

('Carlos', 'López', 'carlos.lopez@email.com'); -- Tercer cliente

GO

-- 3. Crear procedimiento

CREATE PROCEDURE sp_InsertarCliente -- Crea un procedimiento almacenado para insertar un cliente

@Nombre NVARCHAR(100), -- Parámetro para el nombre

@Apellido NVARCHAR(100), -- Parámetro para el apellido

@Email NVARCHAR(255) -- Parámetro para el correo electrónico

AS

BEGIN

SET NOCOUNT ON; -- Evita que SQL Server muestre el número de filas afectadas

BEGIN TRY -- Comienza el bloque de manejo de errores

BEGIN TRANSACTION; -- Inicia una transacción

INSERT INTO Clientes (Nombre, Apellido, Email) -- Inserta los datos del cliente

VALUES (@Nombre, @Apellido, @Email);

COMMIT TRANSACTION; -- Si todo sale bien, confirma la transacción

END TRY

BEGIN CATCH -- Si ocurre un error, se maneja aquí

IF @@TRANCOUNT > 0 -- Verifica si hay una transacción abierta

ROLLBACK TRANSACTION; -- Revierte la transacción en caso de error

THROW; -- Relanza el error para que se propague

END CATCH

END;

GO

-- 4. Índice filtrado

CREATE INDEX IX_Clientes_Activos ON Clientes(ClienteID) -- Crea un índice sobre la columna ClienteID

WHERE Activo = 1; -- El índice solo se aplicará a los clientes activos (Activo = 1)

GO

-- 5. Actualizar estadísticas

UPDATE STATISTICS Clientes WITH FULLSCAN; -- Actualiza las estadísticas de la tabla "Clientes" usando un escaneo completo

GO

-- EJERCICIO 4

-- 1. Backup completo

BACKUP DATABASE PracticaAdminDB -- Realiza un respaldo completo de la base de datos PracticaAdminDB

TO DISK = 'C:\Backups\PracticaAdminDB_Full.bak' -- Guarda el respaldo en la ruta especificada

WITH INIT, COMPRESSION, STATS = 10; -- Sobrescribe el archivo de respaldo, lo comprime y muestra el progreso cada 10%

GO

-- 2. Backup diferencial

INSERT INTO Clientes (Nombre, Apellido, Email) -- Inserta un nuevo cliente

VALUES ('Ana', 'Martínez', 'ana.martinez@email.com'); -- Datos del cliente

GO

BACKUP DATABASE PracticaAdminDB -- Realiza un respaldo diferencial de la base de datos PracticaAdminDB

TO DISK = 'C:\Backups\PracticaAdminDB_Diff.bak' -- Guarda el respaldo diferencial en la ruta especificada

WITH DIFFERENTIAL, COMPRESSION, STATS = 10; -- Solo guarda los cambios desde el último respaldo completo, comprime y muestra progreso

GO

-- 3. Backup de log

INSERT INTO Clientes (Nombre, Apellido, Email) -- Inserta un nuevo registro en la tabla Clientes

VALUES ('Pedro', 'Sánchez', 'pedrossss.sanchez@email.com'); -- Datos del nuevo cliente insertado

GO

ALTER DATABASE PracticaAdminDB SET RECOVERY FULL;

-- Establece el modelo de recuperación de la base de datos como "Full". Esto es necesario para poder hacer backups del log de transacciones, ya que el modelo "Full" permite la recuperación a cualquier punto en el tiempo.

GO

BACKUP LOG PracticaAdminDB -- Realiza una copia de seguridad del log de transacciones de la base de datos

"PracticaAdminDB"

TO DISK = 'C:\Backups\PracticaAdminDB_Log.trn' -- Guarda el respaldo del log en la ruta especificada
(C:\Backups\PracticaAdminDB_Log.trn)

WITH COMPRESSION, STATS = 10; -- Comprime el respaldo y muestra el progreso cada 10% para monitorear el avance del proceso

GO

RESTORE FILELISTONLY

FROM DISK = 'C:\Backups\PracticaAdminDB_Full.bak';

-- Muestra los archivos contenidos en el respaldo completo sin realizar la restauración, lo que permite verificar los archivos antes de restaurar la base de datos.

GO

-- 4. Recuperación

RESTORE DATABASE PracticaAdminDB_Test -- Restaura la base de datos "PracticaAdminDB_Test" desde el respaldo completo

FROM DISK = 'C:\Backups\PracticaAdminDB_Full.bak' -- Usa el archivo de respaldo completo como fuente

WITH

MOVE 'PracticaAdminDB_Data' TO 'C:\Data\PracticaAdminDB_Test_Data.mdf', -- Mueve el archivo de datos principal a una nueva ubicación

MOVE 'PracticaAdminDB_Log' TO 'C:\Data\PracticaAdminDB_Test_Log.ldf', -- Mueve el archivo de log a una nueva ubicación

MOVE 'PracticaAdminDB_Secundario3' TO 'C:\Data\PracticaAdminDB_Test_Secundario3.ndf', -- Mueve un archivo secundario a una nueva ubicación

REPLACE, -- Sobrescribe la base de datos "PracticaAdminDB_Test" si ya existe

STATS = 10, -- Muestra el progreso cada 10%

NORECOVERY; -- Deja la base de datos en un estado donde se pueden aplicar más respaldos (como un respaldo

diferencial o del log)

GO

-- EJERCICIO 5

-- 1. Uso de espacio

EXEC sp_spaceused 'Clientes'; -- Muestra el espacio utilizado por la tabla "Clientes", incluyendo el tamaño de los datos y el índice.

GO

-- 2. Consultas costosas

SELECT TOP 10 -- Muestra las 10 consultas más costosas en términos de lecturas lógicas

qs.execution_count, -- Muestra cuántas veces se ha ejecutado cada consulta

qs.total_logical_reads/qs.execution_count AS avg_logical_reads, -- Muestra el promedio de lecturas lógicas por ejecución

qs.total_elapsed_time/qs.execution_count AS avg_elapsed_time, -- Muestra el tiempo promedio de ejecución por consulta

SUBSTRING(qt.text, (qs.statement_start_offset/2)+1, -- Extrae el texto de la consulta SQL

((CASE qs.statement_end_offset -- Verifica el final de la consulta

WHEN -1 THEN DATALENGTH(qt.text) -- Si no tiene fin, toma el tamaño completo del texto

ELSE qs.statement_end_offset -- Si tiene fin, usa el offset de fin

END - qs.statement_start_offset)/2)+1) AS query_text, -- Extrae el texto SQL de la consulta

qt.dbid, -- ID de la base de datos donde se ejecutó la consulta

qt.objectid -- ID del objeto que generó la consulta (como el procedimiento almacenado o tabla)

```
FROM sys.dm_exec_query_stats AS qs -- Vista dinámica que muestra las estadísticas de ejecución de las consultas
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt -- Une las estadísticas de consulta con el texto SQL
ORDER BY qs.total_logical_reads DESC; -- Ordena por las consultas con más lecturas lógicas (más costosas)
GO
```

-- 3. Bloqueos

```
SELECT
request_session_id AS spid, -- ID de la sesión que solicita el bloqueo
resource_type, -- Tipo de recurso bloqueado (página, fila, base de datos, etc.)
resource_database_id AS dbid, -- ID de la base de datos
request_mode, -- Modo del bloqueo (lectura, escritura, etc.)
request_status -- Estado de la solicitud de bloqueo (esperando, concedido, etc.)
FROM sys.dm_tran_locks -- Vista que muestra información sobre los bloqueos activos
WHERE resource_database_id = DB_ID('PracticaAdminDB'); -- Filtra los bloqueos solo para la base de datos
"PracticaAdminDB"
GO
```

-- 4. Memoria

```
SELECT
object_name, -- Nombre del objeto del contador de rendimiento
counter_name, -- Nombre del contador de memoria
cntr_value -- Valor actual del contador
FROM sys.dm_os_performance_counters -- Vista que muestra los contadores de rendimiento del sistema operativo
WHERE counter_name IN ('Buffer cache hit ratio', 'Page life expectancy') -- Filtra por los contadores de caché de búfer
y vida útil de la página
AND object_name LIKE '%Buffer Manager%'; -- Asegura que los contadores sean del "Buffer Manager" (gestor de
```

caché de SQL Server)

GO

-- TAREAS ADICIONALES

-- Configurar un trabajo del Agente SQL para realizar backups automáticos

-- Crear un nuevo trabajo llamado BackupNocturno

-- Crea un nuevo trabajo en SQL Server Agent llamado 'BackupNocturno'

EXEC msdb.dbo.sp_add_job

@job_name = N'BackupNocturno'; -- Nombre del trabajo a crear

-- Agrega un paso al trabajo para realizar un respaldo completo de la base de datos

EXEC msdb.dbo.sp_add_jobstep

@job_name = N'BackupNocturno', -- Asocia el paso al trabajo creado

@step_name = N'Respaldo Completo', -- Nombre del paso

@subsystem = N'TSQL', -- Tipo de comando (T-SQL)

@command = N'

BACKUP DATABASE PracticaAdminDB

TO DISK = "C:\Backups\PracticaAdminDB_Auto.bak"

WITH INIT, COMPRESSION, STATS = 10;

', -- Comando de respaldo con compresión e inicialización

@database_name = N'master'; -- Base utilizada para ejecutar el comando

-- Programa el trabajo para que se ejecute todos los días a las 12:00 AM

EXEC msdb.dbo.sp_add_schedule

@schedule_name = N'BackupNocturno_Horario', -- Nombre del horario

@freq_type = 4, -- Diario

@freq_interval = 1, -- Cada 1 día

@active_start_time = 120000; -- Hora de inicio: 12:00:00 AM

-- Asocia el horario al trabajo definido

EXEC msdb.dbo.sp_attach_schedule

@job_name = N'BackupNocturno',

@schedule_name = N'BackupNocturno_Horario'; -- Asocia el horario al trabajo

-- Habilita el trabajo para ejecutarse en el servidor actual

EXEC msdb.dbo.sp_add_jobserver

@job_name = N'BackupNocturno'; -- Asocia el trabajo al servidor SQL

-- Reorganiza los índices fragmentados de todas las tablas en la base de datos PracticaAdminDB

USE PracticaAdminDB;

GO

EXEC sp_MSforeachtable @command1 =

'ALTER INDEX ALL ON ? REORGANIZE'; -- Reorganiza todos los índices de cada tabla

GO

-- Actualiza estadísticas de todas las tablas para mejorar rendimiento del motor de consultas

EXEC sp_MSforeachtable @command1 =

'UPDATE STATISTICS ? WITH FULLSCAN'; -- Actualiza estadísticas con escaneo completo

GO