

PRACTICA DE ADMINISTRACION DE BASES DE DATOS CON SQL SERVER

Jocelin Reyes Rodriguez
Grupo 3601

--Ejercicio 1: Creación y Configuración de Base de Datos

-- 1. Crear base de datos

-- Esta instrucción crea una nueva base de datos llamada PracticaAdminDB

CREATE DATABASE PracticaAdminDB

ON PRIMARY

(

-- Nombre lógico del archivo de datos principal

NAME = 'PracticaAdminDB_Data',

-- Ruta y nombre del archivo físico en el disco

FILENAME = 'C:\recuperado\PracticaAdminDB.mdf',

-- Tamaño inicial del archivo de datos

SIZE = 25MB,

-- Tamaño máximo que puede alcanzar el archivo de datos

MAXSIZE = 100MB,

-- Cuánto crecerá el archivo cuando se llene (incremento)

FILEGROWTH = 5MB

)

LOG ON (

-- Nombre lógico del archivo de log

NAME = 'PracticaAdminDB_Log',

-- Ruta y nombre del archivo físico del log

FILENAME = 'C:\recuperado\PracticaAdminDB.ldf',

-- Tamaño inicial del archivo de log

SIZE = 10MB,

-- Tamaño máximo que puede alcanzar el archivo de log

MAXSIZE = 50MB,

-- Cuánto crecerá el archivo de log cuando se llene

FILEGROWTH = 2MB

);

GO -- Indica el fin del lote de instrucciones y ejecuta lo anterior

-- 2. Configurar opciones

-- Establece el modelo de recuperación de la base de datos en SIMPLE.

-- Esto significa que el archivo de log de transacciones no crecerá mucho, ya que no guarda todo el historial de transacciones.

-- Es útil en entornos de desarrollo o bases de datos donde la pérdida mínima de datos es aceptable.

ALTER DATABASE PracticaAdminDB SET RECOVERY SIMPLE;

-- Desactiva la opción de reducción automática de archivos.

-- Esto evita que SQL Server reduzca automáticamente el tamaño de los archivos de base de datos cuando no se necesita,

-- lo cual puede mejorar el rendimiento en algunas situaciones.

ALTER DATABASE PracticaAdminDB SET AUTO_SHRINK OFF;

-- Activa la creación automática de estadísticas.

-- SQL Server creará automáticamente estadísticas sobre columnas utilizadas en filtros o uniones para mejorar el rendimiento de las consultas.

ALTER DATABASE PracticaAdminDB SET AUTO_CREATE_STATISTICS ON;

-- Activa la actualización automática de estadísticas.

-- Esto permite que SQL Server actualice estadísticas desactualizadas para mantener un buen plan de ejecución en las consultas.

ALTER DATABASE PracticaAdminDB SET AUTO_UPDATE_STATISTICS ON;

-- Ejecuta el bloque de instrucciones anterior

GO

-- 3. Crear grupo de archivos secundario

-- Agrega un nuevo grupo de archivos llamado FG_SECUNDARIO a la base de datos PracticaAdminDB.

```

-- Los filegroups permiten organizar archivos de datos en grupos lógicos, lo cual puede mejorar la
administración y el rendimiento.
ALTER DATABASE PracticaAdminDB
ADD FILEGROUP FG_SECUNDARIO;
GO
-- Agrega un nuevo archivo físico de datos al grupo de archivos FG_SECUNDARIO.
-- Esto permite distribuir los datos de la base de datos entre varios archivos (y potencialmente
discos), lo cual puede mejorar el rendimiento.
ALTER DATABASE PracticaAdminDB
ADD FILE
(
    -- Nombre lógico del nuevo archivo de datos
    NAME = 'PracticaAdminDB_Secundario',
    -- Ruta y nombre del archivo físico donde se almacenarán los datos
    FILENAME = 'C:\recuperado\PracticaAdminDB_Secondary.ndf',
    -- Tamaño inicial del archivo
    SIZE = 15MB,
    -- Tamaño máximo que puede alcanzar el archivo
    MAXSIZE = 50MB,
    -- Tamaño del incremento cuando se necesite más espacio
    FILEGROWTH = 5MB
)
-- Indica que este archivo se agregará al grupo de archivos FG_SECUNDARIO
TO FILEGROUP FG_SECUNDARIO;
GO

```

-Ejercicio 2: Administración de Seguridad

```

-- 1. Crear rolUSE PracticaAdminDB;
-- Cambia el contexto a la base de datos PracticaAdminDB para que las siguientes instrucciones se
apliquen en ella.
USE PracticaAdminDB;
GO
-- Crea un rol de base de datos llamado RolLectura.
-- Este rol se puede usar para agrupar usuarios con permisos de solo lectura.
CREATE ROLE RolLectura;
-- Crea un rol de base de datos llamado RolEscritura.
-- Este rol se puede usar para agrupar usuarios con permisos para insertar, actualizar o eliminar
datos.
CREATE ROLE RolEscritura;
-- Crea un rol de base de datos llamado RolAdminDatos.
-- Este rol puede ser usado para usuarios con privilegios más avanzados, como la administración de
datos (sin ser sysadmin).
CREATE ROLE RolAdminDatos;
GO

-- 2. Crear usuarios y asignar roles
-- Crea un inicio de sesión a nivel de servidor llamado Usuario1 con la contraseña especificada.
CREATE LOGIN Usuario1 WITH PASSWORD = 'P@ssw0rd1';
-- Crea un usuario en la base de datos actual (PracticaAdminDB) asociado al login Usuario1.
CREATE USER Usuario1 FOR LOGIN Usuario1;
-- Agrega el usuario Usuario1 al rol RolLectura, dándole permisos de solo lectura según la
configuración del rol.
ALTER ROLE RolLectura ADD MEMBER Usuario1;
-- Crea un segundo login llamado Usuario2 con su respectiva contraseña.
CREATE LOGIN Usuario2 WITH PASSWORD = 'P@ssw0rd2';
-- Crea un usuario en la base de datos asociado a Usuario2.
CREATE USER Usuario2 FOR LOGIN Usuario2;
-- Agrega el usuario Usuario2 al rol RolEscritura, permitiéndole modificar datos según la configuración
del rol.
ALTER ROLE RolEscritura ADD MEMBER Usuario2;
-- Crea un login para un usuario con nivel administrativo de datos.
CREATE LOGIN AdminDatos WITH PASSWORD = 'Adm1nD@t0s';

```

```
-- Crea un usuario en la base de datos asociado al login AdminDatos.
CREATE USER AdminDatos FOR LOGIN AdminDatos;
-- Asigna a AdminDatos al rol RolAdminDatos, otorgándole privilegios más amplios (como gestión de
datos).
ALTER ROLE RolAdminDatos ADD MEMBER AdminDatos;
-- Ejecuta todas las instrucciones anteriores
GO

-- 3. Configurar permisos
-- Otorga permiso de lectura (SELECT) sobre todos los objetos del esquema "dbo" al rol RolLectura.
-- Los miembros de este rol podrán consultar (leer) datos, pero no modificarlos.
GRANT SELECT ON SCHEMA::dbo TO RolLectura;
-- Otorga permisos de inserción, actualización y eliminación (INSERT, UPDATE, DELETE)
-- sobre todos los objetos del esquema "dbo" al rol RolEscritura.
-- Los miembros de este rol podrán modificar los datos, pero no tendrán control total sobre los
objetos.
GRANT INSERT, UPDATE, DELETE ON SCHEMA::dbo TO RolEscritura;
-- Otorga permiso de CONTROL sobre el esquema "dbo" al rol RolAdminDatos.
-- Este es el permiso más amplio a nivel de esquema, permitiendo gestionar todos los objetos del
esquema,
-- incluidos otorgar o revocar permisos a otros usuarios.
GRANT CONTROL ON SCHEMA::dbo TO RolAdminDatos;
-- Ejecuta todas las instrucciones anteriores
GO

-- 4. Crear usuario Auditor
-- Crea un nuevo inicio de sesión a nivel del servidor llamado Auditor con la contraseña indicada.
CREATE LOGIN Auditor WITH PASSWORD = 'Aud1t0r2023';
-- Crea un usuario dentro de la base de datos PracticaAdminDB asociado al login Auditor.
CREATE USER Auditor FOR LOGIN Auditor;
-- Otorga permiso para ver la definición de objetos (como tablas, vistas, procedimientos) dentro de la
base de datos PracticaAdminDB.
-- Esto permite al usuario revisar el diseño de la base de datos, pero no modificarla.
GRANT VIEW DEFINITION ON DATABASE::PracticaAdminDB TO Auditor;
-- Otorga permiso de lectura (SELECT) sobre todos los objetos del esquema "dbo" al usuario Auditor.
-- Esto permite consultar los datos, pero no insertarlos, actualizarlos ni eliminarlos.
GRANT SELECT ON SCHEMA::dbo TO Auditor;
-- Ejecuta todas las instrucciones anteriores
GO
```

--Ejercicio 3: Mantenimiento y Optimización

-- 1. Crear tabla

-- Crea una tabla llamada "Clientes" en la base de datos actual.

```
CREATE TABLE Clientes
```

```
(
    -- Columna "ClienteID" de tipo entero, se autoincrementa empezando en 1 (IDENTITY).
    -- Es la clave primaria (PRIMARY KEY), lo que garantiza que cada cliente tenga un identificador
    único.
    ClienteID INT IDENTITY(1,1) PRIMARY KEY,
    -- Columna "Nombre", tipo cadena de texto Unicode con un máximo de 100 caracteres. No puede ser
    nula.
    Nombre NVARCHAR(100) NOT NULL,
    -- Columna "Apellido", igual que Nombre, no permite valores nulos.
    Apellido NVARCHAR(100) NOT NULL,
    -- Columna "Email", permite hasta 255 caracteres y debe ser única (UNIQUE).
    -- No se permite que dos clientes tengan el mismo correo electrónico.
    Email NVARCHAR(255) UNIQUE,
    -- Columna "FechaRegistro", de tipo DATETIME.
    -- Por defecto se establece con la fecha y hora actual al momento de insertar el registro.
    FechaRegistro DATETIME DEFAULT GETDATE(),
```

```

-- Columna "Activo", tipo BIT (booleano: 1 o 0).
-- Por defecto se establece en 1, lo que indica que el cliente está activo.
Activo BIT DEFAULT 1,
-- Se crea un índice no agrupado (NONCLUSTERED) sobre la columna "Apellido".
-- Esto mejora el rendimiento de las consultas que filtran o ordenan por Apellido.
INDEX IX_Clientes_Apellido NONCLUSTERED (Apellido)
);
-- Ejecuta la creación de la tabla
GO

-- 2. Insertar datos
-- Inserta nuevos registros en la tabla "Clientes".
-- Se especifican únicamente las columnas: Nombre, Apellido y Email.
-- Las demás columnas (FechaRegistro y Activo) tomarán sus valores por defecto definidos en la tabla.
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES
    -- Primer registro: Cliente Juan Pérez con su correo electrónico.
    ('Juan', 'Pérez', 'juan.perez@email.com'),
    -- Segundo registro: Cliente María Gómez.
    ('María', 'Gómez', 'maria.gomez@email.com'),
    -- Tercer registro: Cliente Carlos López.
    ('Carlos', 'López', 'carlos.lopez@email.com');
-- Ejecuta el lote de instrucciones de inserción
GO

-- 3. Crear procedimiento
-- Crea un procedimiento almacenado llamado "sp_InsertarCliente"
-- que permite insertar un nuevo cliente recibiendo como parámetros su nombre, apellido y correo
-- electrónico.
CREATE PROCEDURE sp_InsertarCliente
    @Nombre NVARCHAR(100),          -- Parámetro de entrada para el nombre del cliente
    @Apellido NVARCHAR(100),        -- Parámetro de entrada para el apellido del cliente
    @Email NVARCHAR(255)            -- Parámetro de entrada para el correo electrónico del cliente
AS
BEGIN
    -- Desactiva los mensajes que indican el número de filas afectadas por las operaciones,
    -- para que la ejecución sea más limpia, especialmente desde aplicaciones cliente.
    SET NOCOUNT ON;
    -- Inicio de un bloque TRY para manejar errores durante la ejecución
    BEGIN TRY
        -- Inicia una transacción para asegurar que la operación sea atómica (todo o nada)
        BEGIN TRANSACTION;
        -- Inserta el nuevo cliente usando los valores recibidos como parámetros
        INSERT INTO Clientes (Nombre, Apellido, Email)
        VALUES (@Nombre, @Apellido, @Email);
        -- Si no hay errores, confirma la transacción (hace permanentes los cambios)
        COMMIT TRANSACTION;
    END TRY
    -- Si ocurre un error durante la ejecución del bloque TRY, se ejecuta el siguiente bloque CATCH
    BEGIN CATCH
        -- Si la transacción sigue activa, realiza un rollback para deshacer los cambios
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
        -- Re-lanza el error para que pueda ser manejado externamente si es necesario
        THROW;
    END CATCH
END;
-- Ejecuta la creación del procedimiento
GO

-- 4. Índice filtrado

```

```
-- Crea un índice no agrupado llamado "IX_Clientes_Activos" en la tabla "Clientes".
-- Este índice está basado en la columna "ClienteID" pero solo para aquellos registros donde la columna
"Activo" sea igual a 1 (clientes activos).
CREATE INDEX IX_Clientes_Activos ON Clientes(ClienteID)
-- Se define una condición (WHERE) para incluir solo los registros donde la columna "Activo" sea
igual a 1.
-- Esto optimiza las consultas que filtran o buscan solo los clientes activos.
WHERE Activo = 1;
-- Ejecuta la creación del índice
GO

-- 5. Actualizar estadísticas
-- Actualiza las estadísticas de la tabla "Clientes".
-- Las estadísticas ayudan al optimizador de consultas a crear planes de ejecución eficientes.
-- En este caso, se usa la opción FULLSCAN, lo que significa que se escanearán todos los registros de
la tabla
-- para asegurar que las estadísticas estén completamente actualizadas y reflejen la distribución real
de los datos.
UPDATE STATISTICS Clientes WITH FULLSCAN;
-- Ejecuta la actualización de las estadísticas
GO
```

```
--Ejercicio 4: Copias de Seguridad y Recuperación
-- 1. Backup completo
-- Realiza una copia de seguridad completa de la base de datos "PracticaAdminDB".
-- La copia de seguridad se guarda en el archivo especificado en el disco, en este caso
'C:\recuperado\PracticaAdminDB_Full.bak'.
BACKUP DATABASE PracticaAdminDB
-- Especifica el destino del archivo de la copia de seguridad.
TO DISK = 'C:\recuperado\PracticaAdminDB_Full.bak'
-- La opción "WITH INIT" indica que se sobrescribirá el archivo de copia de seguridad si ya existe
en la ubicación especificada.
WITH INIT,
-- La opción "COMPRESSION" habilita la compresión de la copia de seguridad para reducir el espacio
de almacenamiento requerido.
COMPRESSION,
-- La opción "STATS = 10" muestra un mensaje de progreso cada 10% durante la creación de la copia
de seguridad.
STATS = 10;
-- Ejecuta el comando de copia de seguridad
GO

-- 2. Backup diferencial
-- Inserta un nuevo registro en la tabla "Clientes" con los valores especificados para Nombre, Apellido
y Email.
-- Los demás campos tomarán sus valores por defecto, como FechaRegistro y Activo.
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Ana', 'Martínez', 'ana.martinez@email.com');
-- Ejecuta la instrucción de inserción
GO
-- Realiza una copia de seguridad diferencial de la base de datos "PracticaAdminDB".
-- Esta copia de seguridad solo incluye los cambios realizados desde la última copia de seguridad
completa (FULL).
BACKUP DATABASE PracticaAdminDB
-- Especifica el destino del archivo de la copia de seguridad diferencial.
TO DISK = 'C:\recuperado\PracticaAdminDB_Diff.bak'
-- La opción "DIFFERENTIAL" indica que solo se respaldarán los cambios desde la última copia
completa.
WITH DIFFERENTIAL,
```

```

-- La opción "COMPRESSION" habilita la compresión de la copia de seguridad para reducir el tamaño
del archivo.
COMPRESSION,
-- La opción "STATS = 10" muestra un mensaje de progreso cada 10% durante la creación de la copia
de seguridad.
STATS = 10;
-- Ejecuta la instrucción de copia de seguridad
GO

-- 3. Backup de log
INSERT INTO Clientes (Nombre, Apellido, Email)
VALUES ('Pedro', 'Sánchez', 'pedro.sanchez@email.com');
GO
-- Respaldo completo de la base de datos
BACKUP DATABASE PracticaAdminDB
TO DISK = 'C:\recuperado\PracticaAdminDB_Full.bak'
WITH COMPRESSION, STATS = 10;

-- Ahora que tenemos un respaldo completo, podemos hacer el respaldo del log
BACKUP LOG PracticaAdminDB
TO DISK = 'C:\recuperado\PracticaAdminDB_Log.trn'
WITH COMPRESSION, STATS = 10;

-- 4. Recuperación
-- Restaura una base de datos desde el respaldo completo utilizando la opción MOVE para cambiar la
ubicación de los archivos.
-- En este caso, se está restaurando la base de datos "PracticaAdminDB_Test" desde el archivo de
respaldo completo.
RESTORE DATABASE PracticaAdminDB_Test
FROM DISK = 'C:\recuperado\PracticaAdminDB_Full.bak'
WITH
-- Mueve el archivo de datos principal de la base de datos (datos) a una nueva ubicación en el
disco.
MOVE 'PracticaAdminDB_Data' TO 'C:\recuperado\PracticaAdminDB_Test.mdf',
-- Mueve el archivo de log a una nueva ubicación en el disco.
MOVE 'PracticaAdminDB_Log' TO 'C:\recuperado\PracticaAdminDB_Test.ldf',
-- Mueve el archivo secundario (si existe) a una nueva ubicación.
MOVE 'PracticaAdminDB_Secundario' TO 'C:\recuperado\PracticaAdminDB_Test_Secondary.ndf',
-- La opción "REPLACE" sobrescribe cualquier base de datos existente con el mismo nombre.
REPLACE,
-- La opción "STATS = 10" muestra el progreso de la restauración cada 10%.
STATS = 10,
-- La opción "NORECOVERY" deja la base de datos en un estado de recuperación, permitiendo aplicar
más copias de seguridad (diferenciales o de log).
NORECOVERY;
GO
-- Restaura la base de datos utilizando un respaldo diferencial.
-- Después de restaurar el respaldo completo con NORECOVERY, se restaura el respaldo diferencial con la
opción RECOVERY.
RESTORE DATABASE PracticaAdminDB_Test
FROM DISK = 'C:\recuperado\PracticaAdminDB_Diff.bak'
WITH RECOVERY,
-- La opción "RECOVERY" finaliza el proceso de restauración, dejando la base de datos completamente
restaurada y lista para usar.
STATS = 10;
GO

```

```

-- Ejercicio 5: Monitoreo y Resolución de Problemas
-- 1. Uso de espacio

```

```

-- Ejecuta el procedimiento almacenado "sp_spaceused" para obtener información sobre el uso de espacio
de la tabla "Clientes".
-- Este comando devuelve estadísticas sobre el tamaño de la tabla, incluyendo el espacio total
utilizado y el espacio libre.
EXEC sp_spaceused 'Clientes';
-- Ejecuta la consulta para obtener información del espacio utilizado por la tabla "Clientes"
GO

-- 2. Consultas costosas
-- Realiza una consulta para obtener información sobre las consultas más costosas en términos de
lecturas lógicas.
-- Esto ayuda a identificar las consultas que consumen más recursos y podrían beneficiarse de
optimizaciones.
SELECT TOP 10
    -- Muestra el número de ejecuciones de la consulta.
    qs.execution_count,
    -- Calcula el promedio de lecturas lógicas por ejecución.
    qs.total_logical_reads / qs.execution_count AS avg_logical_reads,
    -- Calcula el tiempo promedio transcurrido por ejecución (en milisegundos).
    qs.total_elapsed_time / qs.execution_count AS avg_elapsed_time,
    -- Extrae el texto de la consulta SQL ejecutada, utilizando el manejo de los desplazamientos de
    inicio y fin.
    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,
    -- Devuelve el ID de la base de datos en la que se ejecutó la consulta.
    qt.dbid,
    -- Devuelve el ID del objeto que está relacionado con la consulta (por ejemplo, una tabla o
    índice).
    qt.objectid
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
-- Ordena los resultados por el número total de lecturas lógicas descendente.
ORDER BY qs.total_logical_reads DESC;
-- Ejecuta la consulta
GO

-- 3. Bloqueos
-- Realiza una consulta para obtener información sobre los bloqueos actuales en la base de datos
"PracticaAdminDB".
-- Esto ayuda a identificar las sesiones que están esperando o reteniendo bloqueos en la base de datos.
SELECT
    -- Muestra el ID de la sesión que está solicitando el bloqueo.
    request_session_id AS spid,
    -- Muestra el tipo de recurso sobre el cual se está solicitando el bloqueo (por ejemplo, página,
    objeto, tabla).
    resource_type,
    -- Muestra el ID de la base de datos sobre la que se solicita el bloqueo.
    resource_database_id AS dbid,
    -- Muestra el tipo de bloqueo solicitado (por ejemplo, compartido, exclusivo).
    request_mode,
    -- Muestra el estado actual del bloqueo (por ejemplo, concedido, en espera).
    request_status
-- Utiliza la vista de administración dinámica "sys.dm_tran_locks" para obtener detalles sobre los
bloqueos de transacciones.
FROM sys.dm_tran_locks
-- Filtra los bloqueos para la base de datos específica llamada "PracticaAdminDB".
WHERE resource_database_id = DB_ID('PracticaAdminDB');

```

```

-- Ejecuta la consulta
GO

-- 4. Memoria
-- Realiza una consulta para obtener información sobre ciertos contadores de rendimiento relacionados
con el manejador de búfer.
-- Esto ayuda a monitorear métricas clave relacionadas con el rendimiento del caché de búfer y la vida
útil de las páginas en memoria.
SELECT
    -- Muestra el nombre del objeto de rendimiento (por ejemplo, "Buffer Manager").
    object_name,
    -- Muestra el nombre del contador de rendimiento (por ejemplo, "Buffer cache hit ratio" o "Page
life expectancy").
    counter_name,
    -- Muestra el valor actual del contador.
    cntr_value
-- Utiliza la vista de administración dinámica "sys.dm_os_performance_counters" para obtener contadores
de rendimiento del sistema operativo.
FROM sys.dm_os_performance_counters
-- Filtra los resultados para obtener solo los contadores específicos de "Buffer cache hit ratio" y
"Page life expectancy".
WHERE counter_name IN ('Buffer cache hit ratio', 'Page life expectancy')
-- Filtra por objetos que contienen "Buffer Manager" en su nombre, para enfocarse en el administrador
de búfer.
AND object_name LIKE '%Buffer Manager%';
-- Ejecuta la consulta
GO

```

```

--Tareas Adicionales
-- Configurar un trabajo del Agente SQL para realizar backups automáticos cada noche.

-- Crear el trabajo de respaldo
EXEC msdb.dbo.sp_add_job
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @enabled = 1, -- Habilitar el trabajo
    @description = N'Backup completo de la base de datos cada noche';

-- Crear el paso del trabajo para el respaldo
EXEC msdb.dbo.sp_add_jobstep
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @step_name = N'Full Backup', -- Nombre del paso
    @subsystem = N'TSQL', -- Tipo de subsistema
    @command = N'BACKUP DATABASE PracticaAdminDB TO DISK =
''C:\recuperado\PracticaAdminDB_Full_$(ESCAPE_QUOTE(DATE)).bak'' WITH INIT, COMPRESSION, STATS = 10;',
-- Comando de respaldo
    @database_name = N'master', -- Base de datos en la que se ejecutará el comando
    @on_success_action = 1, -- Acción en caso de éxito (continuar con el siguiente
paso)
    @on_fail_action = 2; -- Acción en caso de error (detener el trabajo)

-- Crear el horario para el trabajo (cada noche a las 2:00 AM)
EXEC msdb.dbo.sp_add_schedule
    @schedule_name = N'Nightly Backup', -- Nombre del horario
    @enabled = 1, -- Habilitar el horario
    @freq_type = 4, -- Frecuencia diaria (4 significa diario)
    @freq_interval = 1, -- El trabajo debe ejecutarse todos los días
    @freq_subday_type = 8, -- Ejecutar a una hora específica
    @freq_subday_interval = 120, -- A las 2:00 AM (en minutos, 120 minutos después de
medianoche)
    @active_start_time = 20000; -- Hora de inicio (2:00 AM)

-- Asociar el horario con el trabajo

```



```

EXEC msdb.dbo.sp_attach_schedule
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @schedule_name = N'Nightly Backup';    -- Nombre del horario

-- Crear el trabajo de mantenimiento para reorganización de índices y actualización de estadísticas

-- Crear el horario para el trabajo (cada noche a las 2:00 AM)
EXEC msdb.dbo.sp_add_schedule
    @schedule_name = N'Nightly Maintenance', -- Nombre del horario
    @enabled = 1, -- Habilitar el horario
    @freq_type = 4, -- Frecuencia diaria (4 = diario)
    @freq_interval = 1, -- Frecuencia del intervalo (1 = cada día)
    @freq_subday_type = 8, -- Ejecutar a una hora específica
    @freq_subday_interval = 120, -- A las 2:00 AM (en minutos)
    @active_start_time = 20000; -- Hora de inicio (2:00 AM)

    -- Crear el trabajo
EXEC msdb.dbo.sp_add_job
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @enabled = 1, -- Habilitar el trabajo
    @description = N'Backup completo de la base de datos cada noche';

-- Crear el paso del trabajo para el respaldo
EXEC msdb.dbo.sp_add_jobstep
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @step_name = N'Full Backup', -- Nombre del paso
    @subsystem = N'TSQL', -- Tipo de subsistema
    @command = N'BACKUP DATABASE PracticaAdminDB TO DISK =
''C:\Backups\PracticaAdminDB_Full_$(ESCAPE_SQUOTE(DATE)).bak'' WITH INIT, COMPRESSION, STATS = 10;', -
- Comando de respaldo
    @database_name = N'master', -- Base de datos en la que se ejecutará el comando
    @on_success_action = 1, -- Acción en caso de éxito (continuar con el siguiente
paso)
    @on_fail_action = 2; -- Acción en caso de error (detener el trabajo)

-- Asociar el horario con el trabajo
EXEC msdb.dbo.sp_attach_schedule
    @job_name = N'Backup Database Nightly', -- Nombre del trabajo
    @schedule_name = N'Nightly Maintenance'; -- Nombre del horario

```