

Universidad Peruana Los Andes
Facultad de Ingeniería
Escuela profesional de Ingeniería de Sistemas



PRACTICA DE LA SEMANA 10

Curso: Base de datos II

Docente: Raul Enrique Fernandez Bejarano

Estudiante: Sarmiento Mosquera Yeims Abraham

Ciclo: V - Código: s03807f

Huancayo - 2025

Practica de la semana 10

Proyecto 1

Proyecto 1: Creación y distribución de archivos físicos de la base QhatuPeru

Como parte de la expansión comercial de QhatuPeru, la empresa solicita la creación de una nueva base de datos que distribuya los datos entre un archivo primario, un archivo secundario y el archivo de registro de transacciones, asegurando mejor rendimiento y seguridad en el almacenamiento.

Ejercicio práctico:

- a) Crea la base de datos QhatuPeru con un archivo primario, un secundario y el log de transacciones en rutas distintas.
- b) Consulta los archivos físicos asociados a la base QhatuPeru.

Script de la solución:

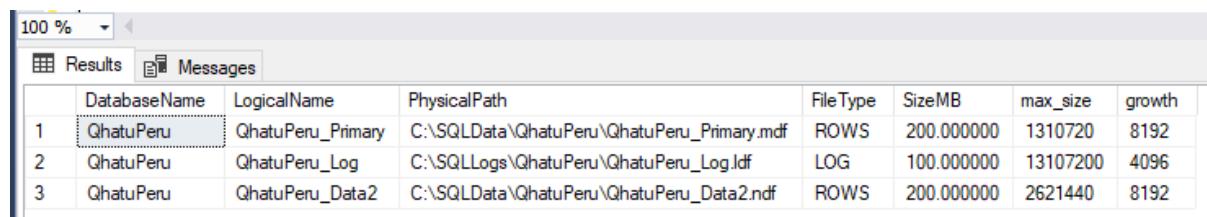
```

-- Crear base de datos QhatuPeru con archivo primario, secundario y log en rutas distintas
CREATE DATABASE QhatuPeru
ON
PRIMARY (
    NAME = QhatuPeru_Primary,
    FILENAME = 'C:\SQLData\QhatuPeru\QhatuPeru_Primary.mdf',
    SIZE = 200MB,
    MAXSIZE = 10GB,
    FILEGROWTH = 64MB
),
FILEGROUP FG_DATA2 (
    NAME = QhatuPeru_Data2,
    FILENAME = 'C:\SQLData\QhatuPeru\QhatuPeru_Data2.ndf',
    SIZE = 200MB,
    MAXSIZE = 20GB,
    FILEGROWTH = 64MB
)
LOG ON (
    NAME = QhatuPeru_Log,
    FILENAME = 'C:\SQLLogs\QhatuPeru\QhatuPeru_Log.ldf',
    SIZE = 100MB,
    MAXSIZE = 100GB,
    FILEGROWTH = 32MB
);
GO

-- Consultar archivos físicos creados a QhatuPeru
USE QhatuPeru;
GO
SELECT
    db_name(database_id) AS DatabaseName,
    name AS LogicalName,
    physical_name AS PhysicalPath,
    type_desc AS FileType,
    size * 8 / 1024.0 AS SizeMB,
    max_size,
    growth
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru');
GO

```

Resultado



The screenshot shows a results grid from SSMS with the following data:

	DatabaseName	LogicalName	PhysicalPath	FileType	SizeMB	max_size	growth
1	QhatuPeru	QhatuPeru_Primary	C:\SQLData\QhatuPeru\QhatuPeru_Primary.mdf	ROWS	200.000000	1310720	8192
2	QhatuPeru	QhatuPeru_Log	C:\SQLLogs\QhatuPeru\QhatuPeru_Log.ldf	LOG	100.000000	13107200	4096
3	QhatuPeru	QhatuPeru_Data2	C:\SQLData\QhatuPeru\QhatuPeru_Data2.ndf	ROWS	200.000000	2621440	8192

Justificación

Creamos la base de datos en 2 carpetas distintas en el disco C, los archivos primarios y secundarios en la carpeta SQLData y el log en SQLLogs, se especifican los tamaños de los archivos y su ruta.

Para la consulta de los archivos usamos sys.master_files que nos brinda información sobre archivos físicos del sistema

Explicación de las buenas prácticas

Separar los archivos de la base de datos nos da una mejor administración sobre estos, además que cuando definimos los tamaños iniciales nos ayuda a controlar el rendimiento

Proyecto 2: Ajuste de configuración y validación de propiedades de QhatuPeru

"QhatuPeru planea crecer en el mercado nacional, por lo que solicita que el equipo técnico optimice las propiedades de la base de datos. Debe cambiar la colación para soportar tildes y configurar el crecimiento automático del archivo de datos, evitando saturaciones inesperadas."

Ejercicio práctico:

- Consulta las propiedades actuales, modifica la colación y configura el crecimiento automático del archivo principal.
- Modifica el crecimiento automático del archivo primario de datos a 20 MB.

Script de la solución

```
-- Verificar las propiedades actuales
USE QhatuPeru;
GO
SELECT
    file_id,
    name AS LogicalName,
    physical_name AS PhysicalPath,
    type_desc AS FileType,
    size * 8 / 1024.0 AS SizeMB,
    max_size,
    growth
FROM sys.database_files;
GO

--cambiar la colacion
ALTER DATABASE QhatuPeru COLLATE Latin1_General_CI_AS;
GO

--Configurar crecimiento automático del archivo primario de datos a 20 MB
USE master;
GO
ALTER DATABASE QhatuPeru
MODIFY FILE (
    NAME = N'QhatuPeru_Primary',
    FILEGROWTH = 20MB
);
GO

--verificar cambios
SELECT
    db_name(database_id) AS DatabaseName,
    name AS LogicalName,
    physical_name AS PhysicalPath,
    type_desc AS FileType,
    size * 8 / 1024.0 AS SizeMB,
    growth
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru');
GO
```

Resultado

	DatabaseName	LogicalName	PhysicalPath	FileType	SizeMB	growth
1	QhatuPeru	QhatuPeru_Primary	C:\SQLData\QhatuPeru\QhatuPeru_Primary.mdf	ROWS	200.000000	2560
2	QhatuPeru	QhatuPeru_Log	C:\SQLLogs\QhatuPeru\QhatuPeru_Log.ldf	LOG	100.000000	4096
3	QhatuPeru	QhatuPeru_Data2	C:\SQLData\QhatuPeru\QhatuPeru_Data2.ndf	ROWS	200.000000	8192

Justificación

Primero verificamos las propiedades actuales, para cambiar la colacion se debe desconectar a los usuarios, pero siendo solo un ejemplo no afectara en nada así que podemos desconectarla sin problema, seguidamente para configurar el crecimiento lo hacemos usando el ALTER DATABASE y colocando en FILEGROWTH el tamaño deseado, finalmente para ver los cambios que realizamos nuevamente usamos el sys.master_files para ver los archivos

Explicación de buenas prácticas

Con el ajuste de autorecimiento de 20MB ayuda a controlar como el archivo va aumentando, controlándolo de mejor manera haciendo uso de una cantidad en específico y no en porcentajes.

Cambiar la colación ayuda a manejar de mejor manera los datos ingresados como por ejemplo con tildes, preparando a la base de datos para que pueda recibirlos

Verificar los cambios ayuda a tener conocimiento como los cambios han sido aplicados, es recomendable verificar los archivos antes y después de la modificación

Proyecto 3: Definición de modelo de recuperación y respaldo para QhatuPeru

"En el área de operaciones de QhatuPeru se identificaron diferentes necesidades de recuperación de datos según el proceso. Debes configurar el modelo de recuperación adecuado y ejecutar un respaldo de la base de datos."

Ejercicio práctico:

- Cambia el modelo de recuperación de QhatuPeru a Simple y luego a Bulk-Logged, explicando la diferencia práctica entre ambos modelos.
- Realiza un respaldo completo después de cambiar al modelo FULL.

Script de la solución

```

-- Cambiar al modelo SIMPLE
ALTER DATABASE QhatuPeru SET RECOVERY SIMPLE;
GO

-- Cambiar al modelo BULK_LOGGED
ALTER DATABASE QhatuPeru SET RECOVERY BULK_LOGGED;
GO

-- Cambiar al modelo FULL
ALTER DATABASE QhatuPeru SET RECOVERY FULL;
GO

-- Realizar respaldo completo
BACKUP DATABASE QhatuPeru
TO DISK = 'C:\SQLBackups\QhatuPeru_FullBackup.bak'
WITH FORMAT,
      NAME = 'Respaldo completo QhatuPeru',
      SKIP,
      INIT,
      STATS = 10;
GO

```

Resultado

```

100 % < 
Messages
20 percent processed.
41 percent processed.
50 percent processed.
60 percent processed.
73 percent processed.
81 percent processed.
92 percent processed.
100 percent processed.
Processed 592 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 40 pages for database 'QhatuPeru', file 'QhatuPeru_Data2' on file 1.
Processed 1 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP DATABASE successfully processed 633 pages in 0.021 seconds (235.188 MB/sec).

Completion time: 2025-11-06T10:17:23.8412653-05:00

```

Justificación

El backup simple no guarda detalles de las transacciones y tampoco permite backups de logs, a comparación del backup bulk_logged que, si guarda transacciones normales, pero minimiza los logs en transacciones masivas.

Usar el backup full que guarda todas las transacciones y es la más segura es la mejor opción, por lo que se guarda en una carpeta aparte de backups de la base de datos

Explicación de buenas prácticas

Usar el backup full nos permite tener un respaldo de la base de datos con todas las transacciones, por lo que es el más seguro y confiable de usar cuando se trata de una

gran base de datos, además ordenar los backups en diferentes carpetas ayuda a la fácil administración de los archivos

Proyecto 4: Implementación de roles y usuarios para seguridad en QhatuPeru

"Con el crecimiento de los equipos de ventas y atención al cliente, QhatuPeru solicita la creación de usuarios con roles diferenciados: cajeros accesan solo a consulta, administradores tienen control total y el gerente requiere acceso a reportes."

Ejercicio práctico:

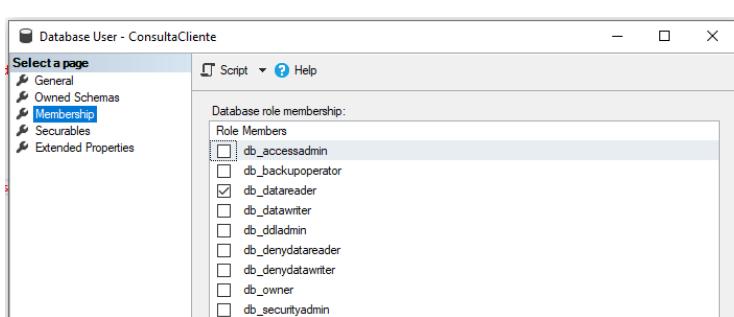
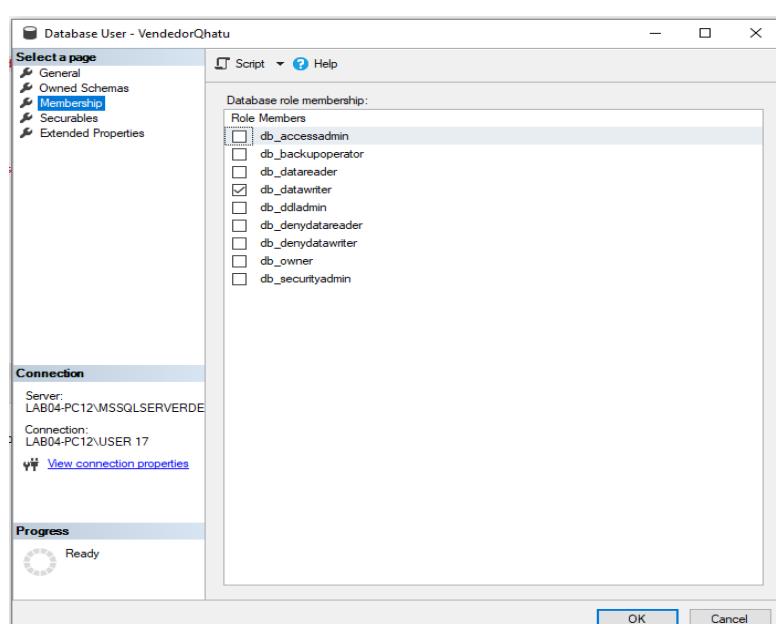
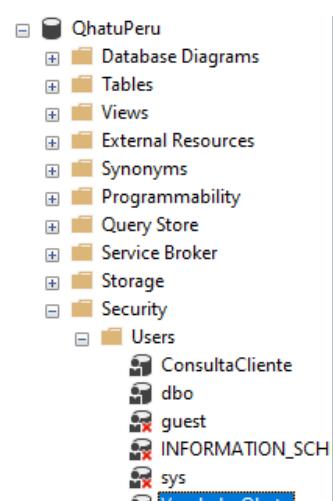
- Crea el usuario VendedorQhatu y asígnalo al rol db_datawriter para registrar ventas.
- Crea el usuario ConsultaCliente y asígnalo solo al rol db_datareader.

Script de la solución

```
USE QhatuPeru;
GO
-- Crear el usuario vendedor
CREATE USER VendedorQhatu WITHOUT LOGIN;
GO
-- Asignar al rol db_datawriter
EXEC sp_addrolemember 'db_datawriter', 'VendedorQhatu';
GO

-- Crear el usuario del cliente
CREATE USER ConsultaCliente WITHOUT LOGIN;
GO
-- Asignar al rol db_datareader
EXEC sp_addrolemember 'db_datareader', 'ConsultaCliente';
GO
```

RESULTADO



Justificación

Para crear el usuario de ventas y el del cliente es facil, esta vez se crea dentro de la base de datos con el comando WITHOUT LOGIN para hacer la prueba, pero se puede crear de diferente manera si se requiere el login, se le asigna los roles de escritura y lectura dependiendo al usuario.

Explicación de buenas prácticas

Asignar los roles de escritura y lectura ayuda a manejar los usuarios que se quieran y las funciones que se les piensa dar, además se debe dar permisos dependiendo a su rol y solo los necesarios

Proyecto 5: Configuración granular de permisos en el módulo de ventas de QhatuPeru

"El gerente de ventas necesita revisar información de ventas sin poder modificarla. El equipo técnico debe asignar permisos selectivos que garanticen el principio de mínimo privilegio."

Ejercicio práctico:

- a) Otorga a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes
- b) Revoca a CajeroQhatu el permiso UPDATE sobre la tabla Ventas.

Script de la solución

```

-- Crear el usuario si no existe (Gerente)
CREATE USER GerenteQhatu WITHOUT LOGIN;
GO

--tabla de ejemplo Reportes
CREATE TABLE dbo.Reportes (
    ReporteID INT IDENTITY(1,1) PRIMARY KEY,
    FechaReporte DATE NOT NULL,
    TipoReporte VARCHAR(50) NOT NULL,
    TotalVentas DECIMAL(12,2) NOT NULL,
    TotalTransacciones INT NOT NULL,
    GeneradoPor NVARCHAR(100) NOT NULL,
    Observaciones NVARCHAR(MAX) NULL
);
GO

-- Otorgar permiso SELECT sobre la tabla Reportes
GRANT SELECT ON dbo.Reportes TO GerenteQhatu;
GO

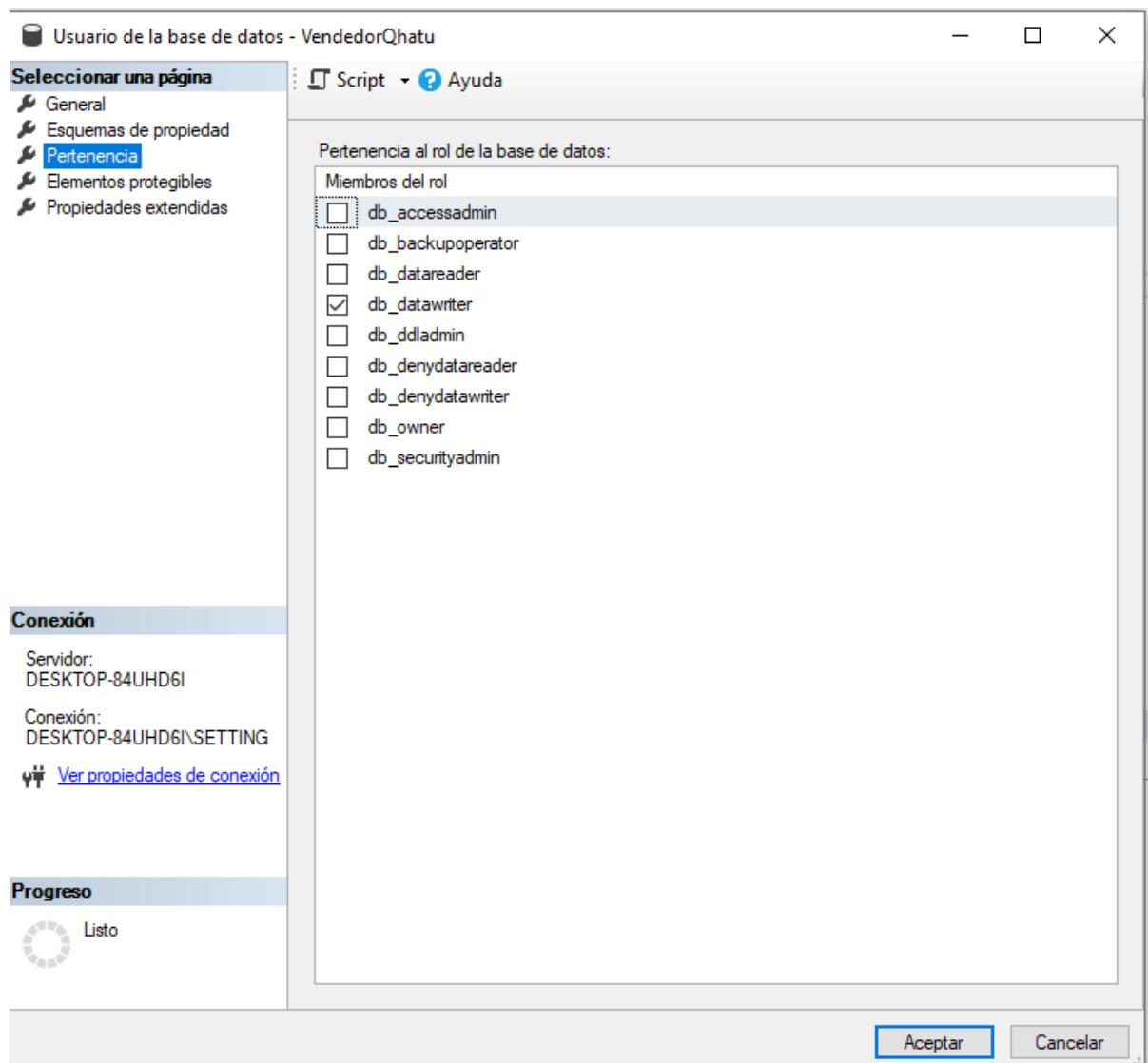
--tabla de ejemplo Ventas
CREATE TABLE dbo.Ventas (
    VentaID INT IDENTITY(1,1) PRIMARY KEY,
    FechaVenta DATETIME NOT NULL,
    ClienteID INT NOT NULL,
    ProductoID INT NOT NULL,
    Cantidad INT CHECK (Cantidad > 0),
    PrecioUnitario DECIMAL(10,2) CHECK (PrecioUnitario >= 0),
    TotalVenta AS (Cantidad * PrecioUnitario) PERSISTED,
    Vendedor NVARCHAR(100) NOT NULL,
    MetodoPago VARCHAR(50) NOT NULL,
    Observaciones NVARCHAR(MAX) NULL
);
GO

-- Crear el usuario si no existe (Cajero)
CREATE USER CajeroQhatu WITHOUT LOGIN;
GO

-- Revocar permiso UPDATE sobre la tabla Ventas
REVOKE UPDATE ON dbo.Ventas FROM CajeroQhatu;
GO

```

RESULTADO



Justificación

Usamos GRANT SELECT para que el usuario pueda consultar datos, pero no modificarlos ni eliminarlos, y REMOVE UPDATE para que el usuario no pueda modificar registros

Explicación de buenas prácticas

Se cumple el requisito de mínimo privilegio que indica que se le debe dar solo los requisitos necesarios al usuario, además la separación de roles ayuda a tener un mejor manejo de los roles

Proyecto 6: Identificación y solución de procesos lentos en QhatuPeru

"Durante una campaña promocional, el sistema presenta lentitud. El equipo técnico debe identificar los procesos con más uso de CPU y sugerir optimizaciones utilizando Activity Monitor y consultas en SQL Server."

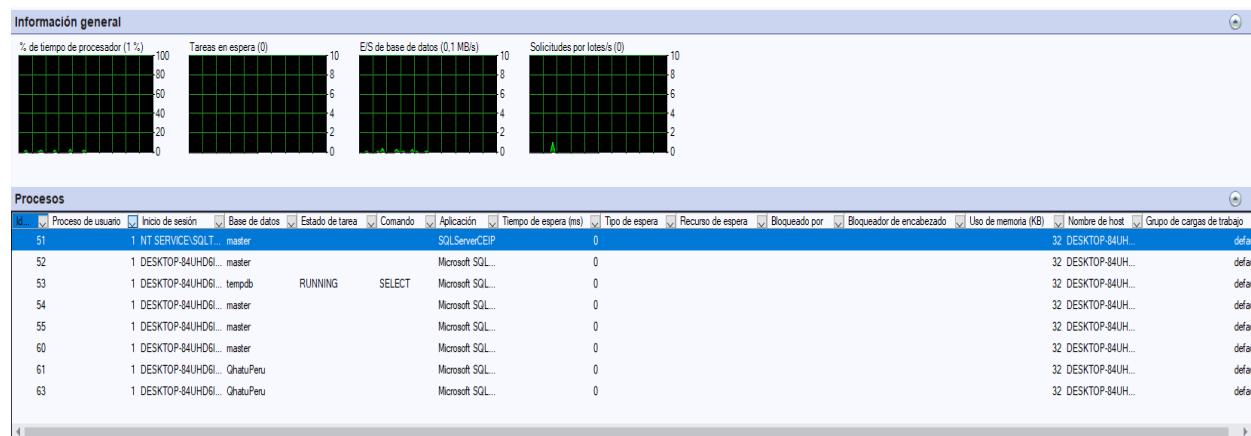
Ejercicio práctico:

- Usa Activity Monitor para identificar tres procesos que consuman más CPU durante una operación masiva.
- Consulta desde T-SQL los bloqueos actuales en la base de datos.

Script de la solución

```
1 USE QhatuPeru;
2 GO
3
4
5     SELECT
6         r.session_id AS SessionID,
7         r.status,
8         r.blocking_session_id AS BlockedBy,
9         r.wait_type,
10        r.wait_time,
11        r.cpu_time,
12        r.total_elapsed_time,
13        t.text AS QueryText,
14        s.login_name,
15        s.host_name,
16        s.program_name
17     FROM sys.dm_exec_requests r
18     JOIN sys.dm_exec_sessions s ON r.session_id = s.session_id
19     CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) t
20     WHERE r.blocking_session_id <> 0;
21 GO
22
```

Resultado



Justificación

Se usa el activity monitor para poder vigilar los recursos que se usan (en este caso importa el ms) o se puede hacer por scripts, aunque es más fácil el activity monitor.

Las consultas con sys.dm_exec_requests y sys.dm_tran_locks permiten ver sin necesidad de la interfaz grafica

Explicación de buenas prácticas

El uso de estas herramientas ayuda a detectar cuellos de botella u operaciones arduas y lentas que se pueden corregir y optimizar, ademas cuando se usa scripts analizar wait_type y cpu_time: ayuda a distinguir si el problema es por CPU, I/O, o bloqueos lógicos.

Proyecto 7: Automatización de respaldos y limpieza del sistema QhatuPeru

"Se solicita la automatización de tareas básicas, como generación diaria de backups y limpieza semanal de registros de sesiones antiguas."

Ejercicio práctico:

- a) Crea un Job en SQL Server Agent que realice un respaldo diario automático de QhatuPeru.
(El paso se hace en la interfaz gráfica, pero el script es:)
- b) Diseña y programa un Job que elimine registros de la tabla Sesiones con más de 15 días de antigüedad cada semana.

Script de la solución

```
1  --script para la creeacion del job que realiza backups automaticos
2  BACKUP DATABASE QhatuPeru
3  TO DISK = 'E:\SQLBackups\QhatuPeru_Diario.bak'
4  WITH FORMAT,
5      NAME = 'Respaldo diario QhatuPeru',
6      INIT,
7      SKIP,
8      STATS = 10;
9  GO
10
11
12  -- creamos la tabla sesiones
13  CREATE TABLE dbo.Sesiones (
14      SesionID INT IDENTITY(1,1) PRIMARY KEY,
15      Usuario NVARCHAR(100) NOT NULL,
16      FechaSesion DATETIME NOT NULL,
17      IPOrigen VARCHAR(45) NULL,
18      EstadoSesion VARCHAR(20) CHECK (EstadoSesion IN ('Activa', 'Finalizada', 'Expirada'))
19      Observaciones NVARCHAR(MAX) NULL
20  );
21  GO
22
23  --script para programar un job que elimine registros de la tabla sesion
24  DELETE FROM dbo.Sesiones
25  WHERE FechaSesion < DATEADD(DAY, -15, GETDATE());
26  GO
```

Resultado

100 % ✓ No se encontraron problemas.

Mensajes

```

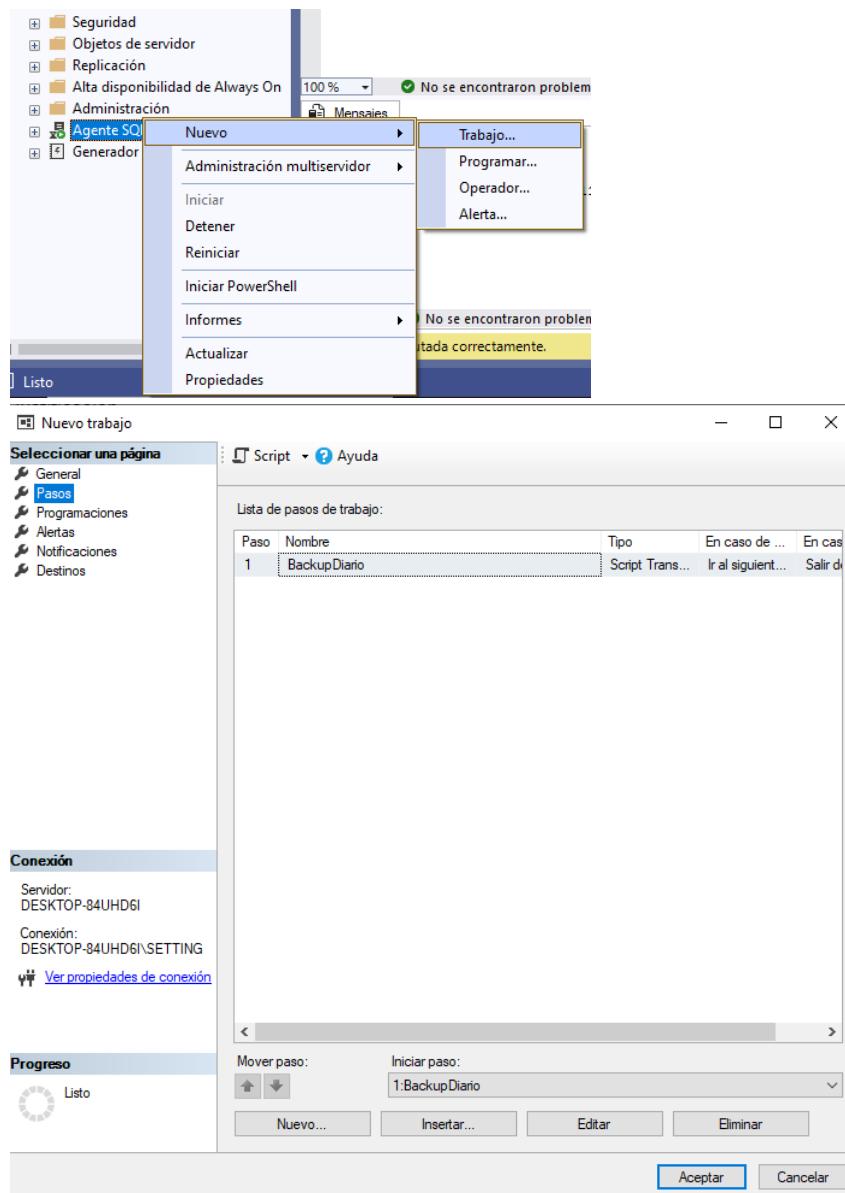
18por ciento procesado.
37por ciento procesado.
47por ciento procesado.
50por ciento procesado.
62por ciento procesado.
70por ciento procesado.
81por ciento procesado.
90por ciento procesado.
100por ciento procesado.
Se han procesado 672 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Primary' en el archivo 1.
Se han procesado 40 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Data2' en el archivo 1.
Se han procesado 1 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Log' en el archivo 1.
BACKUP DATABASE procesó correctamente 713 páginas en 1.700 segundos (3.272 MB/s).

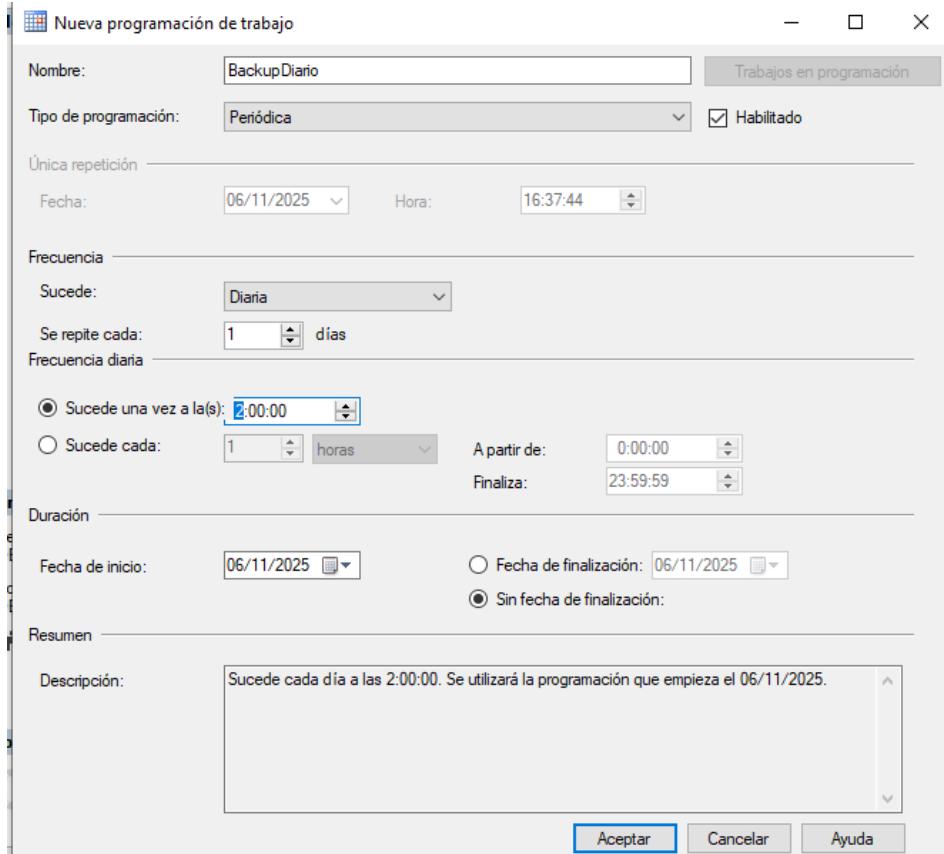
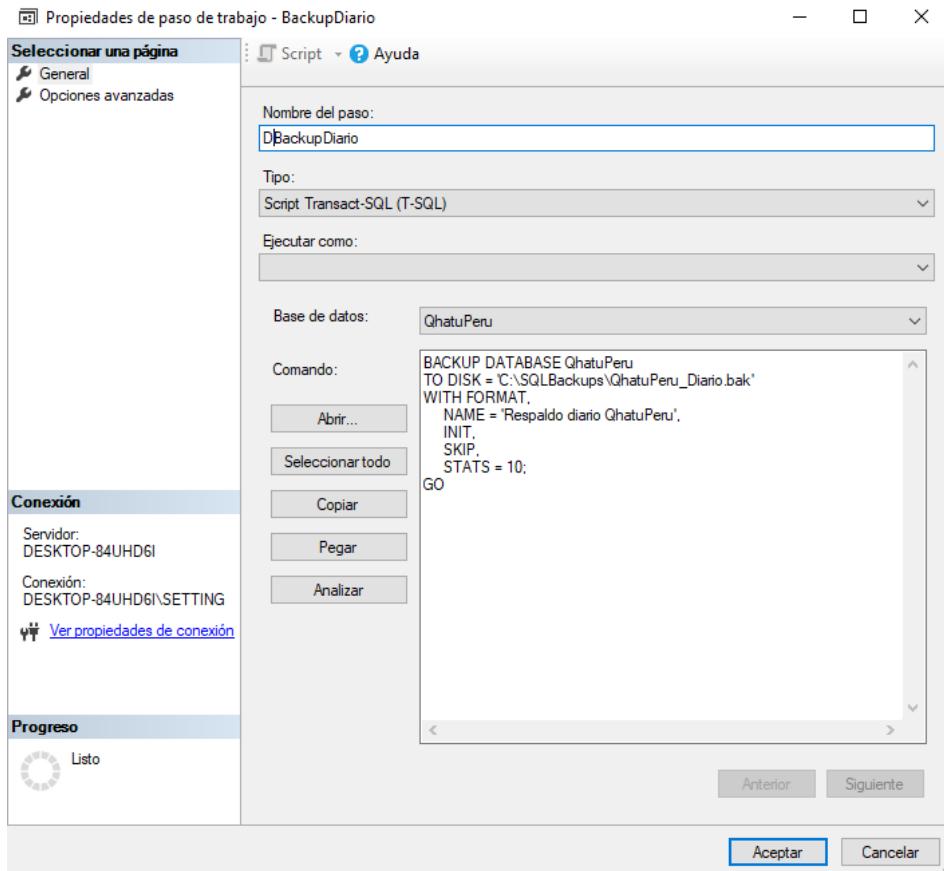
```

Hora de finalización: 2025-11-06T16:25:07.8720362-05:00

Justificación

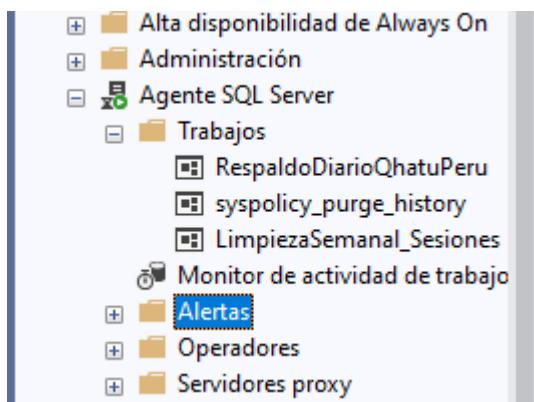
El job se puede crear con la interfaz gráfica y también a través de scripts de la siguiente manera:





Y de igual manera se haria con el otro job (trabajo en la version español)

De esa manera ya tenemos ambos jobs



Explicación de buenas prácticas

La programación de los jobs se hace en un momento en que nadie probablemente esté usando la db como por ejemplo a las 3 de madrugada los domingos (en caso del job de limpieza de registros), además separar los jobs por sus respectivos trabajos permite tener un mejor control sobre lo que se ha automatizado, el job que usa DATEADD también nos permite eliminar los registros antiguos y guardar solo los recientes

Proyecto 8: Registro de estados en pedidos en QhatuPeru

"Con la finalidad de optimizar la trazabilidad de los envíos, se solicita agregar una columna 'EstadoEnvio' en la tabla Pedidos, permitiendo identificar el estado actual de cada pedido."

Ejercicio práctico:

- Agrega la columna "Prioridad" tipo INT a la tabla Pedidos.
- Elimina la columna "EstadoEnvio" de la tabla Pedidos.

Script de la solución

```
1 USE QhatuPeru;
2 GO
3
4 -- teniendo esta tabla de base
5 CREATE TABLE dbo.Pedidos (
6     PedidoID INT IDENTITY(1,1) PRIMARY KEY,
7     FechaPedido DATE NOT NULL,
8     ClienteID INT NOT NULL,
9     ProductoID INT NOT NULL,
10    Cantidad INT CHECK (Cantidad > 0),
11    EstadoEnvio VARCHAR(30) CHECK (EstadoEnvio IN ('Pendiente', 'En tránsito', 'Entregado', 'Cancelado')),
12    Observaciones NVARCHAR(MAX) NULL
13 );
14 GO
15
16 -- agregaremos la columna prioridad
17 ALTER TABLE dbo.Pedidos
18 ADD Prioridad INT NULL;
19 GO
20
21 -- ahora eliminamos la columna EstadoEnvio
22 ALTER TABLE dbo.Pedidos
23 DROP COLUMN EstadoEnvio;
24 GO
```

Resultado

DESKTOP-84U... dbo.Pedidos				SQLQuery8.sql...SETTING (73))*
	Nombre de columna	Tipo de datos	Permitir val...	
▶	PedidoID	int	<input type="checkbox"/>	
	FechaPedido	date	<input type="checkbox"/>	
	ClienteID	int	<input type="checkbox"/>	
	ProductoID	int	<input type="checkbox"/>	
	Cantidad	int	<input checked="" type="checkbox"/>	
	Observaciones	nvarchar(MAX)	<input checked="" type="checkbox"/>	
	Prioridad	int	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

Justificación

Usamos el ALTER TABLE para modificar las tablas existentes, también DROP COLUMN para eliminar las columnas que queramos y ADD para agregar otra columna

Explicación de buenas prácticas

La eliminación de columnas debe darse solo cuando es necesario o no se use la columna, esto de manera controlada y segura, como por ejemplo usando sys.check_constraints, de igual manera separar los pasos ayuda a reducir y/o evitar errores

Proyecto 9: Implementación de registros automáticos de modificaciones en QhatuPeru

"Es necesario auditar todas las modificaciones realizadas en la tabla Clientes para cumplir con políticas internas de seguridad y protección de datos."

Ejercicio práctico:

- a) Crea una tabla AuditoriaClientes para registrar cambios en Clientes.
- b) Desarrolla un trigger que registre en AuditoriaClientes cada eliminación de registros en Clientes.

Script de la solución

```

1 USE QhatuPeru;
2 GO
3
4 -- creamos la tabla auditoria clientes
5 CREATE TABLE dbo.AuditoriaClientes (
6     AuditoriaID INT IDENTITY(1,1) PRIMARY KEY,
7     ClienteID INT NOT NULL,
8     Nombre NVARCHAR(100),
9     DocumentoIdentidad VARCHAR(20),
10    FechaEliminacion DATETIME NOT NULL DEFAULT GETDATE(),
11    UsuarioEliminacion NVARCHAR(100),
12    Motivo NVARCHAR(MAX) NULL
13 );
14 GO
15
16 -- creamos la tabla clientes
17 CREATE TABLE dbo.Clientes (
18     ClienteID INT IDENTITY(1,1) PRIMARY KEY,
19     Nombre NVARCHAR(100) NOT NULL,
20     DocumentoIdentidad VARCHAR(20) NOT NULL UNIQUE,
21     FechaRegistro DATE NOT NULL DEFAULT GETDATE(),
22     CorreoElectronico NVARCHAR(100) NULL,
23     Telefono VARCHAR(15) NULL,
24     Direccion NVARCHAR(200) NULL,
25     EstadoCliente VARCHAR(20) CHECK (EstadoCliente IN ('Activo', 'Inactivo', 'Suspendido')),
26     Observaciones NVARCHAR(MAX) NULL
27 );
28 GO
-- Creamos el trigger que registre cada eliminacion de registros
CREATE TRIGGER trg_Auditoria_Eliminacion_Clientes
ON dbo.Clientes
AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO dbo.AuditoriaClientes (
        ClienteID,
        Nombre,
        DocumentoIdentidad,
        FechaEliminacion,
        UsuarioEliminacion,
        Motivo
    )
    SELECT
        d.ClienteID,
        d.Nombre,
        d.DocumentoIdentidad,
        GETDATE(),
        SYSTEM_USER,
        'Eliminación registrada automáticamente por trigger'
    FROM deleted d;
END;
GO

```

Resultado (ejemplo de eliminación de 2 usuarios)

```

-- ejemplo del uso
-- INSERT INTO dbo.Clientes (Nombre, DocumentoIdentidad, CorreoElectronico, Telefono, Direccion, EstadoCliente, Observaciones)
VALUES
('María López', 'DNI12345678', 'maria.lopez@example.com', '987654321', 'Av. Perú 123', 'Activo', 'Cliente frecuente'),
('Carlos Ruiz', 'DNI87654321', 'carlos.ruiz@example.com', '912345678', 'Jr. Lima 456', 'Activo', 'Solicitó cambio de dirección'),
('Ana Torres', 'DNI11223344', 'ana.torres@example.com', '999888777', 'Calle Sol 789', 'Inactivo', NULL);
GO

-- Eliminar a Carlos Ruiz y Ana Torres
DELETE FROM dbo.Clientes
WHERE DocumentoIdentidad IN ('DNI87654321', 'DNI11223344');
GO

SELECT * FROM dbo.AuditoriaClientes;
GO

```

The screenshot shows a SQL Server Management Studio window. At the top, there's a status bar with '100 %' and a green checkmark icon followed by the text 'No se encontraron problemas.' Below the status bar is a toolbar with icons for 'Resultados' (Results) and 'Mensajes' (Messages). The main area is a table titled 'Resultados' with the following data:

	AuditoriaID	ClienteID	Nombre	DocumentoIdentidad	FechaEliminacion	UsuarioEliminacion	Motivo
1	1	2	Carlos Ruiz	DNI87654321	2025-11-06 17:32:44.240	DESKTOP-84UHD6I\SETTING	Eliminación registrada automáticamente por trigger
2	2	3	Ana Torres	DNI11223344	2025-11-06 17:32:44.240	DESKTOP-84UHD6I\SETTING	Eliminación registrada automáticamente por trigger

Justificación

Usamos un trigger para poder capturar la información de cuando se realiza una eliminación de un registro de la tabla clientes con AFTER DELETE, se asegura que el trigger solo se ejecute cuando se elimina un registro y agarra de deleted los datos antes que ya no existan, además del campo motivo donde ponemos el mensaje del registro del trigger

Explicación de buenas prácticas

El uso de AFTER DELETE permite que solo se ejecute cuando ocurra la eliminación que es lo correcto, además la separación de auditoria permite que no se mezcle la información con otra antigua, y con SYSTEM_USER ayuda en la trazabilidad y en la seguridad

Proyecto 10: Simulación de restauración tras un incidente en QhatuPeru

"Por un error humano se eliminó información de la tabla Clientes. El equipo debe restaurar la base usando el respaldo más reciente y validar la recuperación exitosa de los datos."

Ejercicio práctico:

- a) Simula la eliminación de registros de Clientes y la posterior restauración desde backup.
- b) Verifica la existencia y correcta restauración de los datos con una consulta.

Script de la solución

```

1 USE QhatuPeru;
2 GO
3
4 -- Eliminar todos los registros de la tabla Clientes
5 DELETE FROM dbo.Clientes;
6 GO
7
8 SELECT * FROM dbo.Clientes;
9 GO
10
11 -- Restaurar la base QhatuPeru desde el respaldo
12 USE master;
13 GO
14
15 -- Cerrar conexiones activas
16 ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
17 GO
18
19 -- Restaurar desde archivo de respaldo
20 RESTORE DATABASE QhatuPeru
21 FROM DISK = 'E:\SQLBackups\QhatuPeru_Diario.bak'
22 WITH REPLACE,
23      STATS = 10;
24 GO
25
26 -- Reabrir la base para múltiples usuarios
27 ALTER DATABASE QhatuPeru SET MULTI_USER;
28 GO
29
30 -- Verificar los archivos restaurados
31 SELECT * FROM dbo.Clientes;
32 GO
33

```

Resultado

100% □ 3 ▲ 0 ↑ ↓

ClienteID	Nombre	DocumentoIdentidad	FechaRegistro	CorreoElectronico	Teléfono	Dirección	EstadoCliente	Observaciones

18
19 -- Restaurar desde archivo de respaldo
20 RESTORE DATABASE QhatuPeru
21 FROM DISK = 'E:\SQLBackups\QhatuPeru_Diario.bak'
22 WITH REPLACE,
23 STATS = 10;
24 GO
25
26 -- Reabrir la base para múltiples usuarios
27 ALTER DATABASE QhatuPeru SET MULTI_USER;
28 GO
29
30 -- Verificar los archivos restaurados
31 SELECT * FROM dbo.Clientes;
32 GO

% □ 3 ▲ 0 ↑ ↓

Mensajes

18por ciento procesado.
37por ciento procesado.
55por ciento procesado.
74por ciento procesado.
92por ciento procesado.
100por ciento procesado.
Se han procesado 672 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Primary' en el archivo 1.
Se han procesado 40 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Data2' en el archivo 1.
Se han procesado 1 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_Log' en el archivo 1.
RESTORE DATABASE procesó correctamente 713 páginas en 22.627 segundos (0.245 MB/s).

Hora de finalización: 2025-11-06T17:44:31.4514948-05:00

```
29  
30  
31  
32  
33      -- Verificar los archivos restaurados  
SELECT * FROM dbo.Clientes;  
GO
```

100 % 3 ▲ 0 ↑ ↓

Resultados Mensajes

	ClienteID	Nombre	DocumentoIdentidad	FechaRegistro	CorreoElectronico	Teléfono	Dirección	EstadoCliente	Observaciones
1	1	Maria López	DNI12345678	2025-11-06	maria.lopez@example.com	987654321	Av. Perú 123	Activo	Cliente frecuente
2	2	Carlos Ruiz	DNI87654321	2025-11-06	carlos.ruiz@example.com	912345678	Jr. Lima 456	Activo	Solicitó cambio de dirección
3	3	Ana Torres	DNI11223344	2025-11-06	ana.torres@example.com	999888777	Calle Sol 789	Inactivo	NULL

Justificación

Primero eliminamos los registros que tenemos (3) para luego realizar un backup de los datos que se tenían, demorará un poco dependiendo de los datos que se tenga y luego se volverá a tener los datos correctamente

Explicación de buenas prácticas

Primero cerramos la base de datos para no afectar a los demás usuarios y realizamos el backup, posteriormente lo volvemos a multiusuarios para ver que si se recuperaron los datos gracias al backup que se realizó