

UNIVERSIDAD PERUANA LOS ANDES
FACULTAD INGENIERIA
PROGRAMA DE ESTUDIOS INGENIERIA DE
SISTEMAS Y COMPUTACIÓN



Asignatura:

BASE DE DATOS II

Tema:

PRACTICA 10

Docente:

Bejarano Raúl Fernández

Estudiante:

Breña Quispe Megam

HUANCAYO-2025

Practica 10

ACTIVIDAD

Responda con claridad y precisión cada uno de los siguientes ejercicios prácticos.

utilice el siguiente formato estructurado:

1. Enunciado del ejercicio
2. Script de la solución en T-SQL
3. Justificación técnica de la solución aplicada
4. Explicación de las buenas prácticas utilizadas en el proyecto

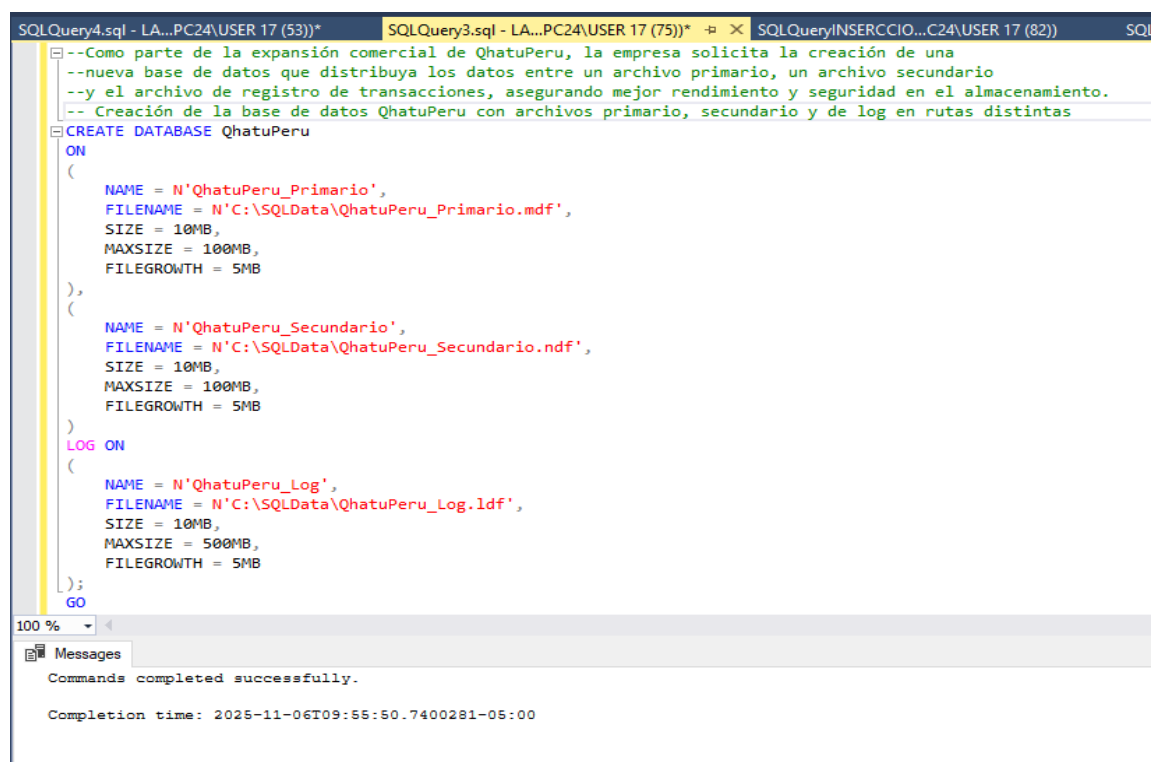
Proyecto 1: Creación y distribución de archivos físicos de la base QhatuPeru

1. Enunciado del ejercicio

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:

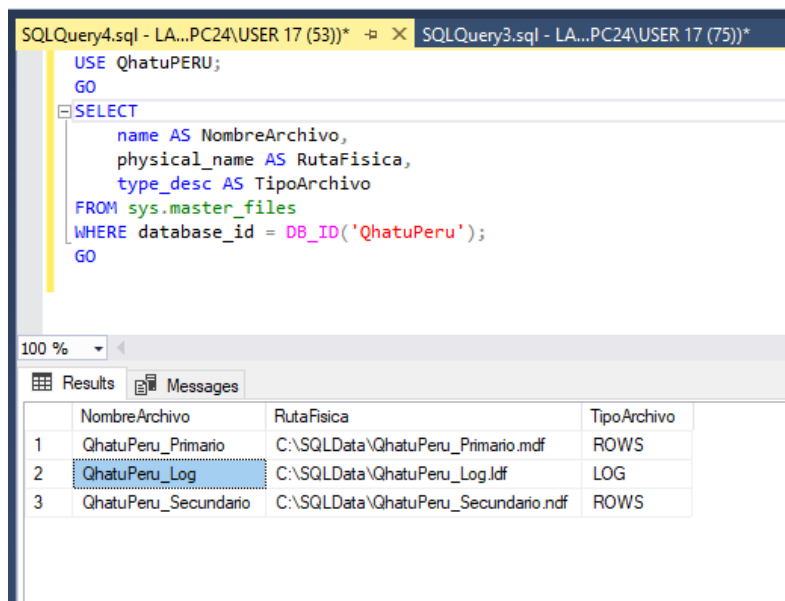
a) Crea la base de datos QhatuPeru con un archivo primario, un secundario y el log de transacciones en rutas distintas.



```
SQLQuery4.sql - LA...PC24\USER 17 (53))*  SQLQuery3.sql - LA...PC24\USER 17 (75))*  SQLQuery1\INSERCCIO...C24\USER 17 (82))  SQL
--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.
-- Creación de la base de datos QhatuPeru con archivos primario, secundario y de log en rutas distintas
CREATE DATABASE QhatuPeru
ON
(
    NAME = N'QhatuPeru_Primary',
    FILENAME = N'C:\SQLData\QhatuPeru_Primary.mdf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 5MB
),
(
    NAME = N'QhatuPeru_Secundario',
    FILENAME = N'C:\SQLData\QhatuPeru_Secundario.ndf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 5MB
)
LOG ON
(
    NAME = N'QhatuPeru_Log',
    FILENAME = N'C:\SQLData\QhatuPeru_Log.ldf',
    SIZE = 10MB,
    MAXSIZE = 500MB,
    FILEGROWTH = 5MB
);
GO

100 %
Messages
Commands completed successfully.
Completion time: 2025-11-06T09:55:50.7400281-05:00
```

b) Consulta los archivos físicos asociados a la base QhatuPeru.



The screenshot shows a SQL query window with the following code:

```
USE QhatuPERU;
GO
SELECT
    name AS NombreArchivo,
    physical_name AS RutaFisica,
    type_desc AS TipoArchivo
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru');
GO
```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	NombreArchivo	RutaFisica	TipoArchivo
1	QhatuPeru_Primary	C:\SQLData\QhatuPeru_Primary.mdf	ROWS
2	QhatuPeru_Log	C:\SQLData\QhatuPeru_Log.ldf	LOG
3	QhatuPeru_Secundario	C:\SQLData\QhatuPeru_Secundario.ndf	ROWS

2. Script de la solución en T-SQL

```
CREATE DATABASE QhatuPeru
```

```
ON
```

```
(
```

```
    NAME = N'QhatuPeru_Primary',
```

```
    FILENAME = N'C:\SQLData\QhatuPeru_Primary.mdf',
```

```
    SIZE = 10MB,
```

```
    MAXSIZE = 100MB,
```

```
    FILEGROWTH = 5MB
```

```
),
```

```
(
```

```
    NAME = N'QhatuPeru_Secundario',
```

```
    FILENAME = N'C:\SQLData\QhatuPeru_Secundario.ndf',
```

```
    SIZE = 10MB,
```

```
    MAXSIZE = 100MB,
```

```
    FILEGROWTH = 5MB
```

```
)
```

```
LOG ON
```

```
(
```

```
    NAME = N'QhatuPeru_Log',
```

```
    FILENAME = N'C:\SQLData\QhatuPeru_Log.ldf',
```

```
    SIZE = 10MB,
```

```
    MAXSIZE = 500MB,
```

```
    FILEGROWTH = 5MB
```

```
);
```

GO

```
USE QhatuPERU;
GO
SELECT
    name AS NombreArchivo,
    physical_name AS RutaFisica,
    type_desc AS TipoArchivo
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru');
GO
```

3. Justificación técnica de la solución aplicada

- **Consulta de crecimiento de archivos:** La consulta utiliza la vista `sys.master_files` para verificar la configuración de crecimiento de los archivos de la base de datos. `growth` devuelve el tamaño de incremento en bytes, y la conversión a megabytes permite una visualización más comprensible.
- **Validación del crecimiento automático:** Validar el crecimiento automático de los archivos es importante para garantizar que la base de datos podrá manejar aumentos en el volumen de datos sin interrupciones.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- **Revisión periódica de configuraciones:** Es importante revisar las configuraciones de crecimiento para asegurarse de que la base de datos está configurada para manejar de manera eficiente los cambios en el volumen de datos, evitando sobresaturaciones o escasez de espacio.
- **Uso de vistas del sistema:** Como se mostró en los ejercicios anteriores, utilizar las vistas del sistema como `sys.master_files` proporciona información clave

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

1. Enunciado del ejercicio

"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:

a) Consulta las propiedades actuales, modifica la colación y configura el crecimiento automático del archivo principal.

The screenshot displays three SQL queries executed in SQL Server Enterprise Manager, along with their results.

Query 1: Consultar las propiedades actuales de la base de datos

```
-- 1@ Consultar las propiedades actuales de la base de datos
SELECT
    name AS NombreBaseDatos,
    collation_name AS ColacionActual
FROM sys.databases
WHERE name = 'QhatuPeru';
GO
```

Results:

	NombreBaseDatos	ColacionActual
1	QhatuPeru	Modem_Spanish_CI_AS

Query 2: Cambiar la colación para soportar tildes y acentos

```
-- 2@ Cambiar la colación para soportar tildes y acentos
ALTER DATABASE QhatuPeru
COLLATE Modern_Spanish_CI_AS; -- Español moderno, insensible a mayúsculas, sensible a acentos
GO

-- 3@ Configurar crecimiento automático del archivo de datos principal (10MB inicialmente)
ALTER DATABASE QhatuPeru
MODIFY FILE
(
    NAME = N'QhatuPeru_Primary',
    FILEGROWTH = 10MB
);
GO
```

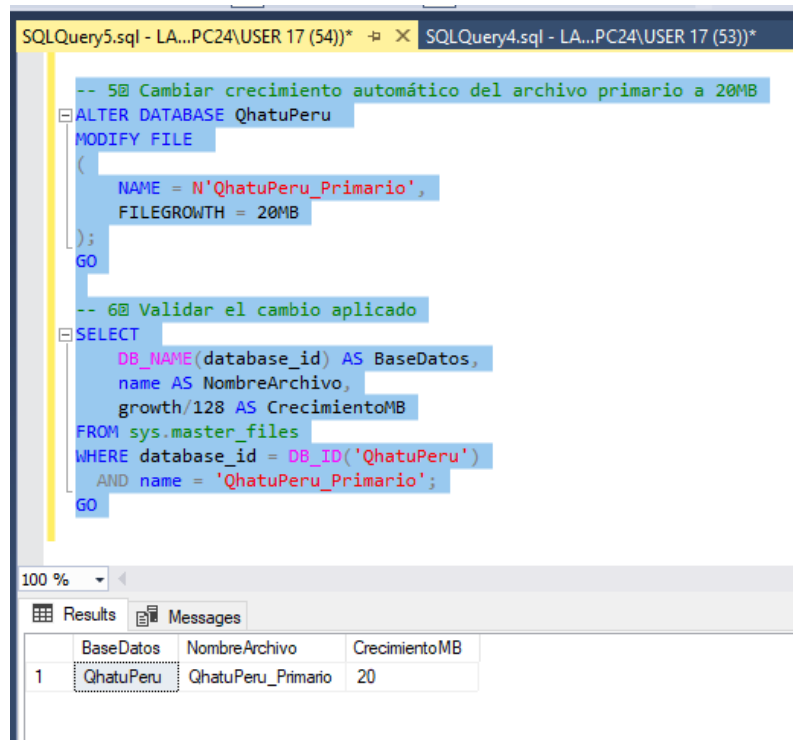
Query 3: Consultar configuración actual de los archivos de QhatuPeru

```
-- 4@ Consultar configuración actual de los archivos de QhatuPeru
USE master;
GO
SELECT
    DB_NAME(database_id) AS BaseDatos,
    name AS NombreArchivo,
    type_desc AS TipoArchivo,
    physical_name AS RutaFisica,
    size/128 AS TamañoMB,
    growth/128 AS CrecimientoMB
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru');
GO
```

Results:

	BaseDatos	NombreArchivo	TipoArchivo	RutaFisica	TamañoMB	CrecimientoMB
1	QhatuPeru	QhatuPeru_Primary	ROWS	C:\SQLData\QhatuPeru_Primary.mdf	10	10
2	QhatuPeru	QhatuPeru_Log	LOG	C:\SQLData\QhatuPeru_Log.ldf	10	5
3	QhatuPeru	QhatuPeru_Secundario	ROWS	C:\SQLData\QhatuPeru_Secundario.ndf	10	5

b) Modifica el crecimiento automático del archivo primario de datos a 20 MB.



```
-- 50 Cambiar crecimiento automático del archivo primario a 20MB
ALTER DATABASE QhatuPeru
MODIFY FILE
(
    NAME = N'QhatuPeru_Primary',
    FILEGROWTH = 20MB
);
GO

-- 60 Validar el cambio aplicado
SELECT
    DB_NAME(database_id) AS BaseDatos,
    name AS NombreArchivo,
    growth/128 AS CrecimientoMB
FROM sys.master_files
WHERE database_id = DB_ID('QhatuPeru')
AND name = 'QhatuPeru_Primary';
GO
```

	BaseDatos	NombreArchivo	CrecimientoMB
1	QhatuPeru	QhatuPeru_Primary	20

2. Script de la solución en T-SQL

-- Consultar las propiedades actuales de la base de datos

SELECT

name AS NombreBaseDatos,
 collation_name AS ColacionActual

FROM sys.databases

WHERE name = 'QhatuPeru';

GO

-- Cambiar la colación para soportar tildes y eñes

ALTER DATABASE QhatuPeru COLLATE Modern_Spanish_CI_AS; -- Español
moderno, insensible a mayúsculas, sensible a acentos GO

-- Configurar crecimiento automático del archivo de datos principal (10MB
inicialmente)

ALTER DATABASE QhatuPeru

MODIFY FILE

(

NAME = N'QhatuPeru_Primary',
 FILEGROWTH = 10MB

);

GO

-- Consultar configuración actual de los archivos de QhatuPeru

USE master;

GO

SELECT DB

NAME(database_id) AS BaseDatos,

name AS NombreArchivo,

type_desc AS TipoArchivo,

physical_name AS RutaFisica,

size/128 AS TamañoMB,

growth/128 AS CrecimientoMB

FROM sys.master_files

WHERE database_id = DB_ID('QhatuPeru');

GO

-- Cambiar crecimiento automático del archivo primario a 20MB

ALTER DATABASE QhatuPeru

MODIFY FILE

(

NAME = N'QhatuPeru_Primary',

FILEGROWTH = 20MB

);

GO

-- Validar el cambio aplicado

SELECT

DB_NAME(database_id) AS BaseDatos,

name AS NombreArchivo,

growth/128 AS CrecimientoMB

FROM sys.master_files

WHERE database_id = DB_ID('QhatuPeru')

AND name = 'QhatuPeru_Primary';

GO

3. Justificación técnica de la solución aplicada

- **Cambio de colación:**

La colación define cómo SQL Server compara y ordena los caracteres.

Se usa Modern_Spanish_CI_AS porque:

- Soporta correctamente los caracteres del idioma español

- No distingue entre mayúsculas/minúsculas
 - Distingue entre letras con y sin tilde
- **Configuración del crecimiento automático:**

El parámetro FILEGROWTH define cuánto crece el archivo de datos cuando se llena, ajustarlo a 10 MB (y luego 20 MB) permite un crecimiento controlado que evita:

 - Saturación inesperada del almacenamiento.
 - Fragmentación de archivos.
 - Sobrecarga del sistema por expansiones muy frecuentes.
- **Verificación mediante vistas del sistema:**

Las consultas a `sys.databases` y `sys.master_files` permiten validar que los cambios se aplicaron correctamente, mostrando colación, tamaño y crecimiento de cada archivo.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- **Verificación antes del cambio:**

Antes de modificar configuraciones, se consultan los valores actuales (`sys.databases`), asegurando control y trazabilidad.
- **Selección adecuada de colación:**

Se utiliza una colación acorde al idioma de los datos (español moderno), lo que evita errores de codificación, búsqueda y ordenamiento.
- **Crecimiento automático planificado:**

Definir un crecimiento moderado (10–20 MB) mejora el rendimiento y la administración del espacio en disco, evitando fragmentación.

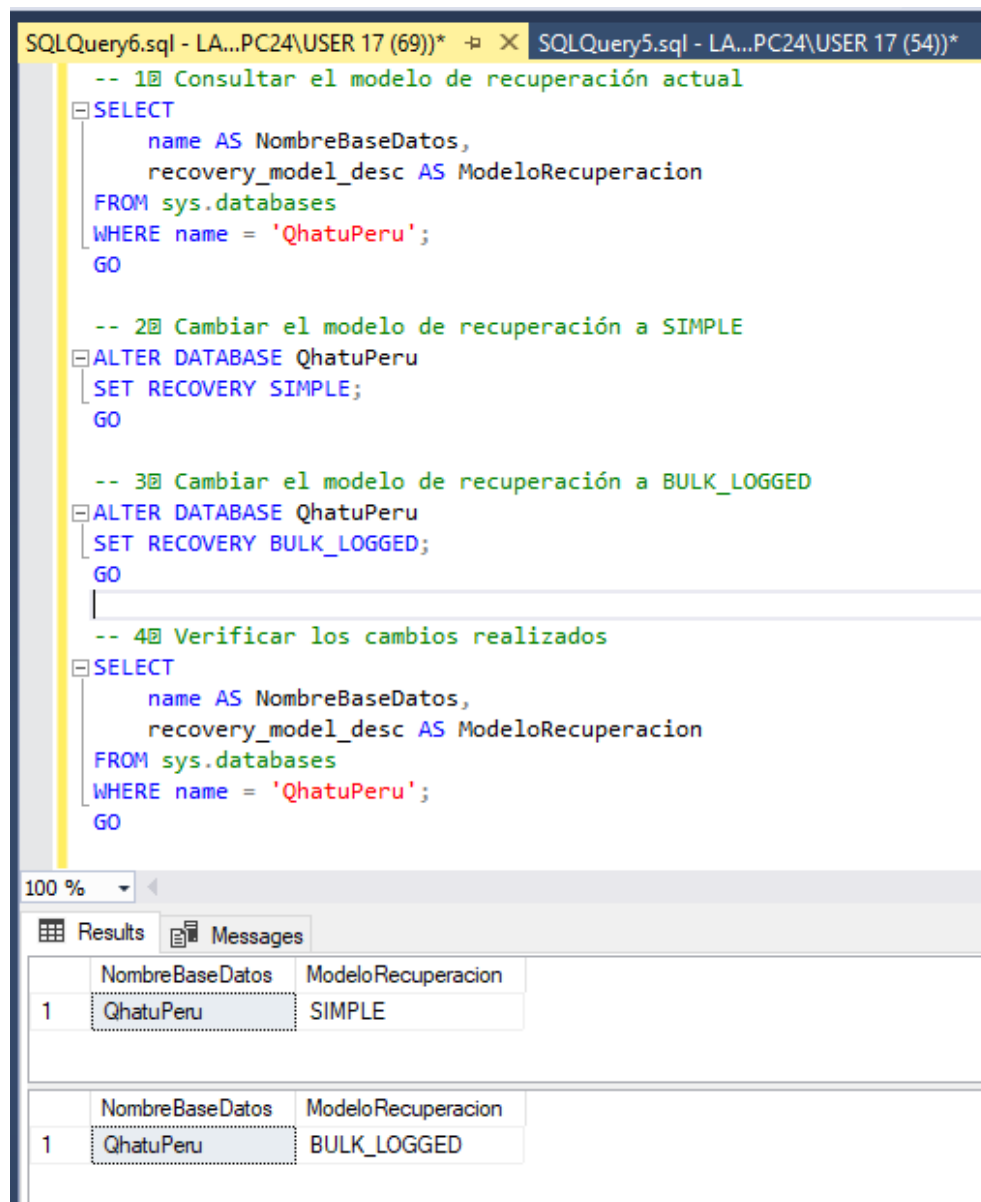
Proyecto 3: Definición de modelo de recuperación y respaldo para QhatuPeru

1. Enunciado del Proyecto

"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:

a) Cambia el modelo de recuperación de QhatuPeru a Simple y luego a Bulk-Logged, explicando la diferencia práctica entre ambos modelos.



The screenshot shows a SQL Server Enterprise Manager window with a script titled "SQLQuery6.sql - LA...PC24\USER 17 (69))*" and "SQLQuery5.sql - LA...PC24\USER 17 (54))*". The script contains the following T-SQL commands:

```
-- 1@ Consultar el modelo de recuperación actual
SELECT
    name AS NombreBaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

-- 2@ Cambiar el modelo de recuperación a SIMPLE
ALTER DATABASE QhatuPeru
SET RECOVERY SIMPLE;
GO

-- 3@ Cambiar el modelo de recuperación a BULK_LOGGED
ALTER DATABASE QhatuPeru
SET RECOVERY BULK_LOGGED;
GO

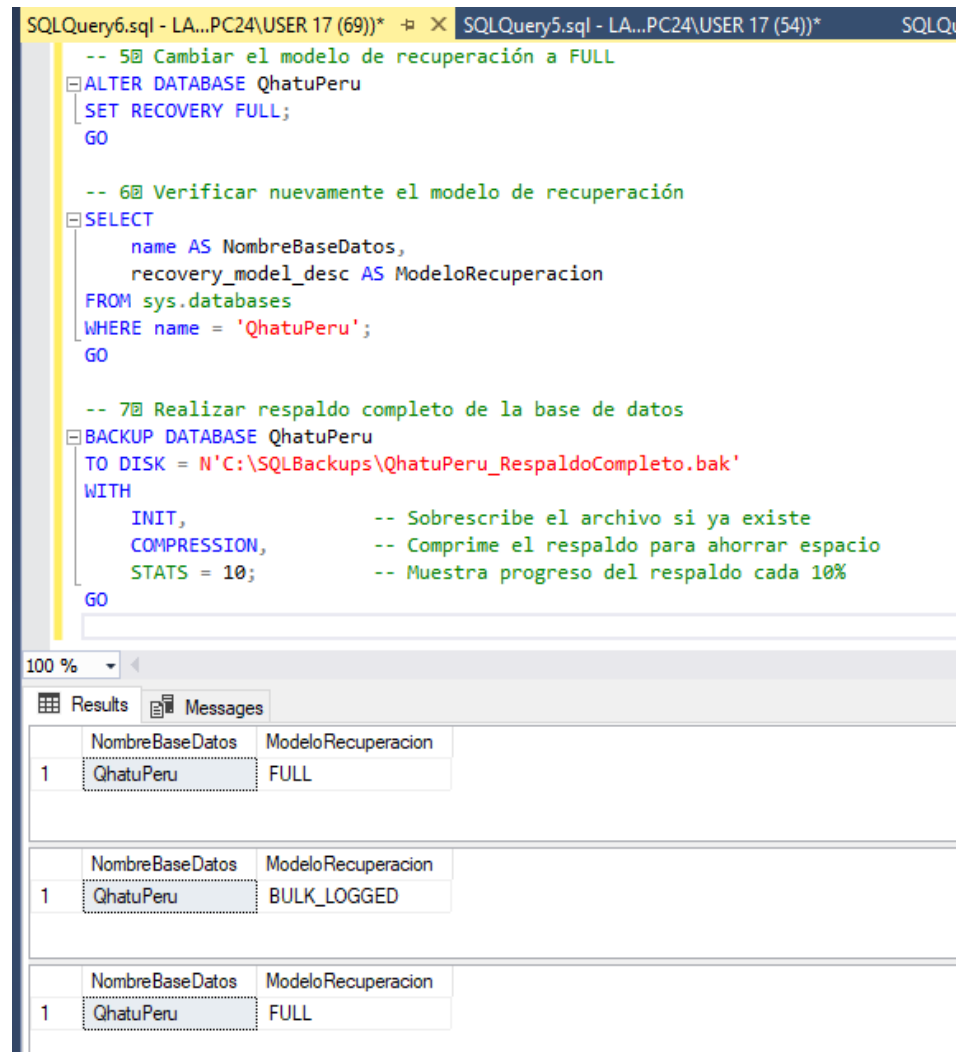
-- 4@ Verificar los cambios realizados
SELECT
    name AS NombreBaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO
```

The bottom of the window shows the "Results" tab with two tables of data. The first table shows the current recovery model as SIMPLE, and the second table shows it as BULK_LOGGED.

	NombreBaseDatos	ModeloRecuperacion
1	QhatuPeru	SIMPLE

	NombreBaseDatos	ModeloRecuperacion
1	QhatuPeru	BULK_LOGGED

b) Realiza un respaldo completo después de cambiar al modelo FULL



The screenshot shows a SQL script in a query window and its execution results in the Results pane. The script performs three tasks: changing the recovery model to FULL, verifying the model, and performing a full backup.

```
-- 5@ Cambiar el modelo de recuperación a FULL
ALTER DATABASE QhatuPeru
SET RECOVERY FULL;
GO

-- 6@ Verificar nuevamente el modelo de recuperación
SELECT
    name AS NombreBaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

-- 7@ Realizar respaldo completo de la base de datos
BACKUP DATABASE QhatuPeru
TO DISK = N'C:\SQLBackups\QhatuPeru_RespaldoCompleto.bak'
WITH
    INIT, -- Sobrescribe el archivo si ya existe
    COMPRESSION, -- Comprime el respaldo para ahorrar espacio
    STATS = 10; -- Muestra progreso del respaldo cada 10%
GO
```

The Results pane shows three rows of data, indicating the recovery model was successfully changed to FULL and the backup was completed.

	NombreBaseDatos	ModeloRecuperacion
1	QhatuPeru	FULL
1	QhatuPeru	BULK_LOGGED
1	QhatuPeru	FULL

2. Script de la solución en T-SQL

-- Consultar el modelo de recuperación actual

```
SELECT name AS NombreBaseDatos, recovery_model_desc AS ModeloRecuperacion
FROM sys.databases WHERE name = 'QhatuPeru'; GO
```

-- Cambiar el modelo de recuperación a SIMPLE

```
ALTER DATABASE QhatuPeru SET RECOVERY SIMPLE; GO
```

-- Cambiar el modelo de recuperación a BULK_LOGGED

```
ALTER DATABASE QhatuPeru SET RECOVERY BULK_LOGGED; GO
```

-- Verificar los cambios realizados

```
SELECT name AS NombreBaseDatos, recovery_model_desc AS ModeloRecuperacion  
FROM sys.databases WHERE name = 'QhatuPeru'; GO
```

-- Cambiar el modelo de recuperación a FULL

```
ALTER DATABASE QhatuPeru SET RECOVERY FULL; GO
```

-- Verificar nuevamente el modelo de recuperación

```
SELECT name AS NombreBaseDatos, recovery_model_desc AS  
ModeloRecuperacion FROM sys.databases WHERE name = 'QhatuPeru'; GO
```

-- Realizar respaldo completo de la base de datos BACKUP

```
DATABASE QhatuPeru TO DISK =  
N'C:\SQLBackups\QhatuPeru_RespaldoCompleto.bak' WITH INIT, -- Sobrescribe el  
archivo si ya existe COMPRESSION, -- Comprime el respaldo para ahorrar espacio  
STATS = 10; -- Muestra progreso del respaldo cada 10% GO
```

3. Justificación técnica de la solución aplicada

- **Cambio de modelo a FULL antes del respaldo:**
Se requiere el modelo FULL para realizar respaldos **completos y diferenciales** que permitan una recuperación detallada.
En este modo, el archivo de log conserva todas las transacciones hasta que se respalde.
- **Respaldo completo (BACKUP DATABASE):**
Crea una copia exacta de todos los datos y estructuras de la base, garantizando una restauración total en caso de pérdida.
- **Opción WITH COMPRESSION:**
Permite que el respaldo ocupe menos espacio físico en disco, optimizando recursos.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- **Verificación previa y posterior:**
Consultar el modelo de recuperación antes y después de cada cambio asegura que las configuraciones se aplicaron correctamente.

- **Uso de rutas seguras:**
Los respaldos se almacenan en una carpeta externa (C:\SQLBackups) para protegerlos de fallos en los archivos de datos.
- **Modelos de recuperación adecuados al contexto:**
Se utiliza **SIMPLE** para entornos sin necesidad de restauración detallada, **BULK_LOGGED** para operaciones masivas, y **FULL** para máxima protección de datos.
- **Respaldo con compresión y control de progreso:**
Mejora el rendimiento y reduce espacio de almacenamiento, además de mostrar el avance del proceso.
- **Automatización posible:**
Estos comandos pueden incluirse en tareas programadas de mantenimiento (SQL Agent Jobs) para garantizar respaldos periódicos sin intervención manual.

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

1. Enunciado del Proyecto

"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:

a) Crea el usuario VendedorQhatu y asignalo al rol db_datawriter para registrar ventas.

```
SQLQuery1.sql - L...H2M65\Megam (72))* -P X
-- a) Crear usuario VendedorQhatu y asignarlo al rol db_datawriter
USE QhatuPeru;
GO

CREATE LOGIN VendedorQhatu WITH PASSWORD = 'Vendedor2025*';
GO

CREATE USER VendedorQhatu FOR LOGIN VendedorQhatu;
GO

EXEC sp_addrolemember 'db_datawriter', 'VendedorQhatu';
GO

-- b) Crear usuario ConsultaCliente y asignarlo solo al rol db_datareader
CREATE LOGIN ConsultaCliente WITH PASSWORD = 'Cliente2025*';
GO

CREATE USER ConsultaCliente FOR LOGIN ConsultaCliente;
GO

EXEC sp_addrolemember 'db_datareader', 'ConsultaCliente';
GO
```

b) Crea el usuario ConsultaCliente y asígnalo solo al rol db_datareader.

```
SQLQuery1.sql - L...H2M65\Megam (72))* -P X
-- a) Crear usuario VendedorQhatu y asignarlo al rol db_datawriter
USE QhatuPeru;
GO

CREATE LOGIN VendedorQhatu WITH PASSWORD = 'Vendedor2025*';
GO

CREATE USER VendedorQhatu FOR LOGIN VendedorQhatu;
GO

EXEC sp_addrolemember 'db_datawriter', 'VendedorQhatu';
GO

-- b) Crear usuario ConsultaCliente y asignarlo solo al rol db_datareader
CREATE LOGIN ConsultaCliente WITH PASSWORD = 'Cliente2025*';
GO

CREATE USER ConsultaCliente FOR LOGIN ConsultaCliente;
GO

EXEC sp_addrolemember 'db_datareader', 'ConsultaCliente';
GO
```

2. Script de la solución en T-SQL

-- a) Crear usuario VendedorQhatu y asignarlo al rol db_datawriter

USE QhatuPeru; GO

CREATE LOGIN VendedorQhatu WITH PASSWORD = 'Vendedor2025*'; GO

CREATE USER VendedorQhatu FOR LOGIN VendedorQhatu; GO

EXEC sp_addrolemember 'db_datawriter', 'VendedorQhatu'; GO

-- b) Crear usuario ConsultaCliente y asignarlo solo al rol db_datareader

CREATE LOGIN ConsultaCliente WITH PASSWORD = 'Cliente2025*'; GO

```
CREATE USER ConsultaCliente FOR LOGIN ConsultaCliente; GO
```

```
EXEC sp_addrolemember 'db_datareader', 'ConsultaCliente'; GO
```

3. Justificación técnica de la solución aplicada

- **Separación entre LOGIN y USER:**
Se crean *logins* a nivel de servidor (para autenticación) y *usuarios* a nivel de base de datos (para autorización). Esto cumple el modelo de seguridad de SQL Server.
- **Uso de roles predefinidos:**
- db_datawriter permite insertar, actualizar y eliminar datos. Es adecuado para un vendedor que registra ventas.
- db_datareader permite consultar información sin modificarla. Es adecuado para un usuario de consultas de clientes.
- **Principio de privilegios mínimos:**
Cada usuario recibe únicamente los permisos necesarios. Vendedor no tiene permisos para leer todo y el usuario de consulta no puede modificar datos.
- **Contraseñas fuertes:**
Se definen contraseñas con complejidad mínima requerida por SQL Server.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- **Reutilización de roles predefinidos:**
Evita configuraciones manuales erróneas y mantiene la administración simple.
- **Uso de scripts parametrizados y claros:**
Cada bloque crea login, usuario y asigna roles de forma ordenada y auditable.
- **Separación clara de ambientes:**
Los usuarios se crean en la base de datos QhatuPeru, no en todas las bases del servidor, evitando accesos innecesarios.

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

1. Enunciado del Proyecto

"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:


```
SQLQuery1.sql - L...H2M65\Megam (72))* X
CREATE DATABASE Qperu;
GO
USE Qperu;
GO

-----
-- 1. Eliminar usuarios si ya existen (evita errores)
-----
IF EXISTS (SELECT 1 FROM sys.database_principals WHERE name = 'GerenteQhatu')
    DROP USER GerenteQhatu;

IF EXISTS (SELECT 1 FROM sys.database_principals WHERE name = 'CajeroQhatu')
    DROP USER CajeroQhatu;
GO

-----
-- 2. Crear usuarios a partir de los logins existentes
-----
CREATE USER GerenteQhatu FOR LOGIN GerenteQhatu;
CREATE USER CajeroQhatu FOR LOGIN CajeroQhatu;
GO

100 %
Messages
Commands completed successfully.
Completion time: 2025-11-06T21:04:23.0355118-05:00
```

a) Otorga a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes

```
-- a) Otorgar a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes
GRANT SELECT ON dbo.Reportes TO GerenteQhatu;
GO

-- b) Revocar a CajeroQhatu el permiso UPDATE sobre la tabla Ventas
REVOKE UPDATE ON dbo.Ventas FROM CajeroQhatu;
GO
```

b) Revoca a CajeroQhatu el permiso UPDATE sobre la tabla Ventas.

```
-- a) Otorgar a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes
GRANT SELECT ON dbo.Reportes TO GerenteQhatu;
GO

-- b) Revocar a CajeroQhatu el permiso UPDATE sobre la tabla Ventas
REVOKE UPDATE ON dbo.Ventas FROM CajeroQhatu;
GO
```

2. Script de la solución en T-SQL

-- 1. Eliminar usuarios si ya existen (evita errores)

IF EXISTS (SELECT 1 FROM sys.database_principals WHERE name = 'GerenteQhatu') DROP USER GerenteQhatu;

IF EXISTS (SELECT 1 FROM sys.database_principals WHERE name = 'CajeroQhatu') DROP USER CajeroQhatu; GO

-- 2. Crear usuarios a partir de los logins existentes

CREATE USER GerenteQhatu FOR LOGIN GerenteQhatu; CREATE USER CajeroQhatu FOR LOGIN CajeroQhatu; GO

-- 3. Asignación de permisos (Ejercicio solicitado)

-- a) Otorgar a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes

GRANT SELECT ON dbo.Reportes TO GerenteQhatu; GO

-- b) Revocar a CajeroQhatu el permiso UPDATE sobre la tabla Ventas

REVOKE UPDATE ON dbo.Ventas FROM CajeroQhatu; GO

3. Justificación técnica de la solución aplicada

- **Uso de GRANT para permisos explícitos:**
GRANT SELECT otorga solo lectura sobre la tabla Reportes, cumpliendo la necesidad del gerente de revisar información sin modificarla.
- **Aplicación de REVOKE para retirar privilegios:**
REVOKE UPDATE elimina el permiso de modificación sobre la tabla Ventas en caso el usuario lo haya heredado de un rol o asignación previa.
- **Control granular de objetos:**
Los permisos se aplican directamente a nivel de tabla, asegurando que el usuario tenga únicamente las acciones requeridas.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- **Permisos específicos por tabla:**
Evita otorgar permisos globales innecesarios. Se controla exactamente qué puede hacer cada usuario sobre cada objeto.
- **Revocación ante cambios de funciones:**
Adaptar permisos al rol real del usuario disminuye riesgos y mantiene la base actualizada con las funciones de cada puesto.
- **Separación de funciones:**
Si un usuario solo debe visualizar o solo registrar ventas, se ajustan permisos para reforzar responsabilidades sin superposición.
- **Auditabilidad y control:**
Scripts claros permiten mantener una trazabilidad precisa de la configuración de seguridad.

Proyecto 6: Identificación y solución de procesos lentos en QhatuPeru

1. Enunciado del ejercicio

"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:

2. Script de la solución en T-SQL

a) Usa Activity Monitor para identificar tres procesos que consuman más CPU durante una operación masiva.

```
USE QhatuPERU;  
GO
```

```
-- Consulta pesada 1 (SCAN completo sin índice)  
SELECT *  
FROM ARTICULO  
CROSS JOIN PROVEEDOR;  
GO
```

```
-- Consulta pesada 2  
SELECT *  
FROM ORDEN_DETALLE od  
JOIN ARTICULO a ON od.CodArticulo = a.CodArticulo  
CROSS JOIN LINEA;  
GO
```

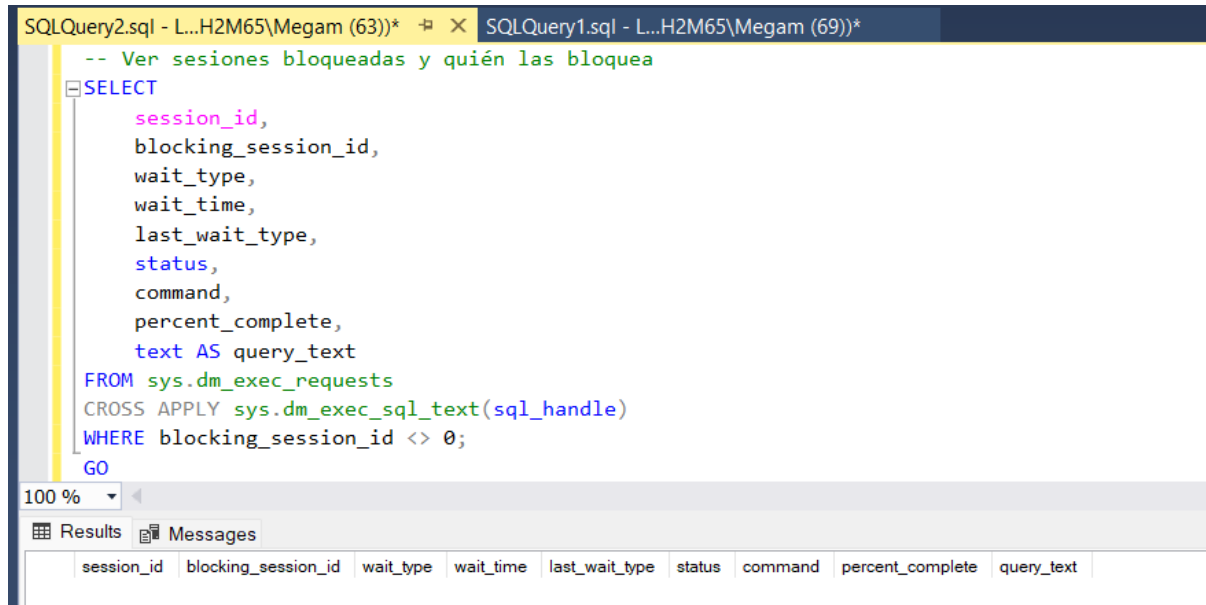
```
-- Consulta pesada 3  
SELECT COUNT(*), CodLinea  
FROM ARTICULO  
GROUP BY CodLinea  
HAVING COUNT(*) > 0;  
GO
```



```

SELECT session_id, blocking_session_id, wait_type, wait_time, last_wait_type,
status, command, percent_complete, text AS query_text
FROM sys.dm_exec_requests
CROSS APPLY sys.dm_exec_sql_text(sql_handle)
WHERE blocking_session_id <> 0;
GO

```



```

SQLQuery2.sql - L...H2M65\Megam (63))*  SQLQuery1.sql - L...H2M65\Megam (69))*
-- Ver sesiones bloqueadas y quién las bloquea
SELECT
    session_id,
    blocking_session_id,
    wait_type,
    wait_time,
    last_wait_type,
    status,
    command,
    percent_complete,
    text AS query_text
FROM sys.dm_exec_requests
CROSS APPLY sys.dm_exec_sql_text(sql_handle)
WHERE blocking_session_id <> 0;
GO

```

session_id	blocking_session_id	wait_type	wait_time	last_wait_type	status	command	percent_complete	query_text
------------	---------------------	-----------	-----------	----------------	--------	---------	------------------	------------

-- Ver todos los bloqueos existentes

```

SELECT tl.resource_type, tl.resource_database_id,
DB_NAME(tl.resource_database_id) AS DatabaseName,
tl.resource_associated_entity_id, tl.request_mode, tl.request_status,
tl.request_session_id, wt.blocking_session_id, wt.wait_type, wt.wait_duration_ms
FROM sys.dm_tran_locks tl
LEFT JOIN sys.dm_os_waiting_tasks wt
ON tl.lock_owner_address = wt.resource_address
WHERE DB_NAME(tl.resource_database_id) = 'QhatuPERU'; GO

```

SQLQuery2.sql - L...H2M65\Megam (63))* SQLQuery1.sql - L...H2M65\Megam (69))*

```
-- Ver todos los bloqueos existentes
SELECT
    tl.resource_type,
    tl.resource_database_id,
    DB_NAME(tl.resource_database_id) AS DatabaseName,
    tl.resource_associated_entity_id,
    tl.request_mode,
    tl.request_status,
    tl.request_session_id,
    wt.blocking_session_id,
    wt.wait_type,
    wt.wait_duration_ms
FROM sys.dm_tran_locks tl
LEFT JOIN sys.dm_os_waiting_tasks wt
    ON tl.lock_owner_address = wt.resource_address
WHERE DB_NAME(tl.resource_database_id) = 'QhātuPERU';
GO
```

100 %

Results Messages

	resource_type	resource_database_id	DatabaseName	resource_associated_entity_id	request_mode	request_status	request_session_id	blocking_session_id	wait_type	wait_duration_ms
1	DATABASE	5	QhātuPERU	0	S	GRANT	65	NULL	NULL	NULL
2	DATABASE	5	QhātuPERU	0	S	GRANT	64	NULL	NULL	NULL
3	DATABASE	5	QhātuPERU	0	S	GRANT	63	NULL	NULL	NULL
4	DATABASE	5	QhātuPERU	0	S	GRANT	69	NULL	NULL	NULL

-- Ver árbol de bloqueos (muy útil)

EXEC sp_who2;

GO

SQLQuery2.sql - L...H2M65\Megam (63))* SQLQuery1.sql - L...H2M65\Megam (69))*

```
-- Ver árbol de bloqueos (muy útil)
EXEC sp_who2;
GO
```

100 %

Results Messages

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID
1	1	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:30		1	0
2	2	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:50:24		2	0
3	3	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:30		3	0
4	4	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:30		4	0
5	5	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:30		5	0
6	6	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:50:24		6	0
7	7	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:35		7	0
8	8	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:35		8	0
9	9	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:35		9	0
10	10	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:35		10	0
11	11	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:35		11	0
12	12	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:55:50		12	0
13	13	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		13	0
14	14	BACKGROUND	sa	.	.	NULL	PARALLEL REDO TASK	0	0	11/16 14:37:02		14	0
15	15	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		15	0
16	16	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		16	0
17	17	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		17	0
18	18	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		18	0
19	19	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		19	0
20	20	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		20	0
21	21	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		21	0
22	22	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		22	0
23	23	BACKGROUND	sa	.	.	NULL	PARALLEL REDO TASK	0	0	11/16 14:37:02		23	0
24	24	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		24	0
25	25	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		25	0
26	26	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		26	0
27	27	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		27	0
28	28	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:51:19		28	0
29	30	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:52:19		30	0
30	31	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/18 21:56:20		31	0
31	32	BACKGROUND	sa	.	.	master	BRKR TASK	0	0	11/18 21:56:43		32	0

3. Justificación técnica de la solución aplicada

Por qué se usa Activity Monitor

Activity Monitor permite detectar visualmente:

- Procesos que consumen **más CPU**
- Consultas que generan **lecturas excesivas (I/O)**
- Sesiones bloqueadas
- Esperas SQL (Waits)

Por qué se ejecutan consultas pesadas

- Aumento del % de CPU
- Lecturas altas (I/O elevado)
- Posibles bloqueos entre sesiones

Por qué se usa `sys.dm_exec_requests`

Permite ver en tiempo real:

- Sesiones activas
- Sesiones bloqueadas
- Quién está bloqueando a quién
- Tipo de espera (wait_type)
- Consulta exacta que se está ejecutando

Por qué se usa `sys.dm_tran_locks`

- Qué objetos están bloqueados
- Tipo de bloqueo (S, X, IX)
- A qué sesión pertenece cada lock
- Conflictos entre transacciones

4. Explicación de las buenas prácticas utilizadas en el proyecto

Se usan DMV (Dynamic Management Views) que permiten monitoreo sin afectar operaciones.

- Activity Monitor → análisis visual rápido
- T-SQL DMV → análisis técnico detallado

Ambos se complementan.

- Las consultas CROSS JOIN no modifican datos y solo generan carga temporal.
- Las tablas ARTICULO, LINEA, ORDEN, etc., mantienen coherencia semántica.
- Esto permite identificar **qué consulta exacta provoca la lentitud**, que es la base de todo tuning.

Proyecto 7: Automatización de respaldos y limpieza del sistema QhatuPeru

1. Enunciado del ejercicio

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

Ejercicio práctico:

2. Script de la solución en T-SQL

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:)

-- Crear el Job

```
EXEC msdb.dbo.sp_add_job @job_name = N'Backup_Diario_QhatuPeru';
```

-- Agregar step que hace el backup

```
EXEC msdb.dbo.sp_add_jobstep @job_name =  
N'Backup_Diario_QhatuPeru', @step_name = N'RespaldoDiario',  
@subsystem = N'TSQL', @command = N'BACKUP DATABASE QhatuPERU TO  
DISK = "C:\Backups\QhatuPERU_diario.bak" WITH FORMAT, INIT, NAME =  
"Respaldo Diario QhatuPERU", SKIP, NOUNLOAD, STATS = 10;',  
@retry_attempts = 3, @retry_interval = 5;
```

-- Programar ejecución diaria

```
EXEC msdb.dbo.sp_add_schedule @schedule_name =  
N'HorarioBackupDiario', @freq_type = 4, -- Diario @freq_interval = 1, -- Cada  
1 día @active_start_time = 020000, -- Hora: 02:00 AM (formato hhmmss)  
@schedule_id = @schedule_id OUTPUT;
```

-- Asociar Job con el horario

```
EXEC msdb.dbo.sp_attach_schedule @job_name =  
N'Backup_Diario_QhatuPeru', @schedule_name = N'HorarioBackupDiario';
```

-- Habilitar para que se ejecute

```
EXEC msdb.dbo.sp_add_jobserver @job_name =  
N'Backup_Diario_QhatuPeru';
```

b) Diseña y programa un Job que elimine registros de la tabla Sesiones con más de 15 días de antigüedad cada semana.

-- Crear el Job

```
EXEC msdb.dbo.sp_add_job @job_name = N'Limpieza_Sesiones_Antiguas';
```

-- Agregar step de eliminación

```
EXEC msdb.dbo.sp_add_jobstep @job_name =  
N'Limpieza_Sesiones_Antiguas', @step_name = N'EliminarSesionesViejas',  
@subsystem = N'TSQL', @command = N'DELETE FROM Sesiones WHERE  
FechaSesion < DATEADD(DAY, -15, GETDATE());;
```

-- Crear horario semanal

```
EXEC msdb.dbo.sp_add_schedule @schedule_name =  
N'HorarioLimpiezaSemanal', @freq_type = 8, -- Semanal @freq_interval = 1, -  
- Domingo @active_start_time = 030000; -- 3:00 AM
```

-- Asociarlo al Job

```
EXEC msdb.dbo.sp_attach_schedule @job_name =  
N'Limpieza_Sesiones_Antiguas', @schedule_name =  
N'HorarioLimpiezaSemanal';
```

-- Habilitar Job

```
EXEC msdb.dbo.sp_add_jobserver @job_name =  
N'Limpieza_Sesiones_Antiguas';
```

3. Justificación técnica de la solución aplicada

El uso de SQL Server Agent garantiza:

- **Automatización completa:** sin intervención humana.
- **Confiabilidad:** los Jobs pueden reintentarse si fallan.
- **Monitoreo:** permite revisar historial de ejecución.
- **Buenas prácticas profesionales:** toda empresa con base de datos usa tareas programadas.

4. Explicación de las buenas prácticas utilizadas en el proyecto

Se creó un **Job del SQL Server Agent** con un *step* que ejecuta:

```
BACKUP DATABASE QhatuPERU TO DISK = ...
```

El Job se programó para ejecutar **todos los días a las 2:00 AM**, horario de baja carga, lo cual:

- Reduce impacto en rendimiento.
- Asegura consistencia de la copia.
- Mantiene un historial de respaldos organizado.

Proyecto 8: Registro de estados en pedidos en QhatuPeru

1. Enunciado del ejercicio

"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:

2. Script de la solución en T-SQL

The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery3.sql' and 'SQLQuery2.sql'. The 'SQLQuery3.sql' tab is active, displaying the following SQL code:

```
USE QhatuPERU;
GO
--AGREGAR LA COLUMNA PRIORIDAD
ALTER TABLE GUIA_ENVIO
ADD Prioridad INT;
GO

--AGREGAR LA CULUNANA ESTADOENVIO
ALTER TABLE GUIA_ENVIO
ADD EstadoEnvio VARCHAR(20);
GO

SELECT name
FROM sys.columns
WHERE object_id = OBJECT_ID('GUIA_ENVIO');
GO
```

Below the code editor, the 'Results' tab is selected, showing a table with 6 rows and 1 column named 'name'. The rows contain the following values:

	name
1	NumGuia
2	CodTienda
3	FechaSalida
4	CodTransportista
5	Prioridad
6	EstadoEnvio

a) Agrega la columna "Prioridad" tipo INT a la tabla GUIA_ENVIO.

ALTER TABLE GUIA_ENVIO

ADD Prioridad INT;

GO

b) Elimina la columna "EstadoEnvio" de la tabla GUIA_ENVIO.

--AGREGAR LA CULUNANA ESTADOENVIO

ALTER TABLE GUIA_ENVIO

ADD EstadoEnvio VARCHAR(20);

GO

--ELIMANR LA COLUMNA ESTADOENVIO

ALTER TABLE GUIA_ENVIO

DROP COLUMN EstadoEnvio;

GO

```
--AGREGAR LA CULUNANA ESTADOENVIO
ALTER TABLE GUIA_ENVIO
ADD EstadoEnvio VARCHAR(20);
GO

--ELIMANR LA COLUMNA ESTADOENVIO
ALTER TABLE GUIA_ENVIO
DROP COLUMN EstadoEnvio;
GO

SELECT name
FROM sys.columns
WHERE object_id = OBJECT_ID('GUIA_ENVIO');
GO
```

100 %

Results Messages

	name
1	NumGuia
2	CodTienda
3	FechaSalida
4	CodTransportista
5	Prioridad

3. Justificación técnica de la solución aplicada

Agregar Prioridad

Se agrega para:

- Identificar qué envíos deben procesarse primero.
- Mejorar la planificación logística.
- Permitir ranking y filtros rápidos.

INT es ideal porque:

- Es ligero.
- Permite comparar prioridades numéricas.
- Optimiza búsquedas y ordenamientos.

Eliminar EstadoEnvio

Eliminar columnas obsoletas es bueno porque:

- Reduce errores
- Evita duplicidad de información
- Mantiene limpio el modelo de datos
- Permite migrar el estado a tablas más completas en el futuro

4. Explicación de las buenas prácticas utilizadas en el proyecto

- Uso correcto de ALTER TABLE
- Mantener integridad y claridad del modelo
- Nombres de columnas claros
- Estructura simple y escalable

Proyecto 9: Implementación de registros automáticos de modificaciones en QhatuPeru

1. Enunciado del ejercicio

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

Ejercicio práctico:

2. Script de la solución en T-SQL

a) Crea una tabla AuditoriaClientes para registrar cambios en Clientes.

1) Crear tabla Clientes

USE QhatuPERU;

GO

CREATE TABLE CLIENTES (

CodCliente INT IDENTITY(1,1) PRIMARY KEY,

NomCliente VARCHAR(50) NOT NULL,

Direccion VARCHAR(60),

Telefono VARCHAR(20),

FechaRegistro DATETIME DEFAULT GETDATE()

);

GO

2) Crear tabla AuditoriaClientes

CREATE TABLE AuditoriaClientes (

IdAuditoria INT IDENTITY(1,1) PRIMARY KEY,

CodCliente INT,

NomCliente VARCHAR(50),

Direccion VARCHAR(60),

Telefono VARCHAR(20),

FechaRegistro DATETIME,

FechaEliminacion DATETIME,

UsuarioSistema VARCHAR(50)

);

GO

b) Desarrolla un trigger que registre en AuditoriaClientes cada eliminación de registros en Clientes.

1) Crear trigger para registrar eliminaciones

CREATE TRIGGER TRG_Auditoria_Clientes_Delete

ON CLIENTES

FOR DELETE AS

BEGIN

```

INSERT INTO AuditoriaClientes ( CodCliente, NomCliente, Direccion,
Telefono, FechaRegistro, FechaEliminacion, UsuarioSistema )
SELECT d.CodCliente, d.NomCliente, d.Direccion, d.Telefono,
d.FechaRegistro, GETDATE(), SUSER_SNAME()
FROM deleted d; END; GO

```

2) Prueba del trigger

```

-- Insertar datos de ejemplo INSERT INTO CLIENTES (NomCliente,
Direccion, Telefono) VALUES ('Juan Torres', 'Av Perú 123', '987654321');
-- Eliminar un cliente DELETE FROM CLIENTES WHERE CodCliente = 1;
-- Ver auditoría SELECT * FROM AuditoriaClientes; GO

```

```

-- Insertar datos de ejemplo
INSERT INTO CLIENTES (NomCliente, Direccion, Telefono)
VALUES ('Juan Torres', 'Av Perú 123', '987654321');

-- Eliminar un cliente
DELETE FROM CLIENTES WHERE CodCliente = 1;

-- Ver auditoría
SELECT * FROM AuditoriaClientes;
GO

```

100 %

Results Messages

	IdAuditoria	CodCliente	NomCliente	Direccion	Telefono	FechaRegistro	FechaEliminacion	UsuarioSistema
1	1	1	Juan Torres	Av Perú 123	987654321	2025-11-18 23:00:31.903	2025-11-18 23:00:31.910	LAPTOP-5OMH2M65IMegam

3. Justificación técnica de la solución aplicada

- **La tabla AuditoriaClientes** permite guardar un histórico de las eliminaciones realizadas, lo que garantiza trazabilidad y cumplimiento de políticas de seguridad.
- **El trigger AFTER DELETE** captura automáticamente los datos eliminados de la tabla Clientes utilizando la tabla lógica deleted.
Se registra:
quién eliminó el registro (SUSER_SNAME ())
cuándo lo eliminó (GETDATE ())
los datos antes de ser eliminados
- La implementación asegura que **ningún borrado pase desapercibido**, incluso si proviene de aplicaciones, scripts o procedimientos almacenados.

4. Explicación de las buenas prácticas utilizadas en el proyecto

- Uso de tabla de auditoría dedicada
- Se registran datos críticos del registro eliminado
- Uso de triggers con acciones precisas (DELETE)

- Registro del usuario del sistema (SUSER_SNAME())
- Evitar auditorías excesivas (solo DELETE)
- Estructura clara y mantenible
-

Proyecto 10: Simulación de restauración tras un incidente en QhatuPeru

1. Enunciado del ejercicio

"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:

2. Script de la solución en T-SQL

a) Simula la eliminación de registros de Clientes y la posterior restauración desde backup.

Primero, insertamos datos de ejemplo:

```
USE QhatuPERU;
```

```
GO
```

```
INSERT INTO CLIENTES (NomCliente, Direccion, Telefono)
```

```
VALUES
```

```
('Carlos Gutiérrez', 'Av. Arequipa 1500', '999111222'),
```

```
('Ana Morales', 'Jr. Cuzco 345', '988776655'),
```

```
('Luis Salazar', 'Av. Brasil 900', '977665544');
```

```
GO
```

Script de backup (lo que se debió ejecutar antes del incidente)

```
BACKUP DATABASE QhatuPERU
```

```
TO DISK = 'C:\Respaldo\QhatuPERU.bak'
```

```
WITH INIT;
```

```
GO
```

Ahora simulamos el error:

```
-- Eliminación accidental
```

```
DELETE FROM CLIENTES;
```

```
GO
```

Verificamos que los datos desaparecieron:

```
SELECT * FROM CLIENTES;
```

```
GO
```

```
USE QhatuPERU;
GO

INSERT INTO CLIENTES (NomCliente, Direccion, Telefono)
VALUES
('Carlos Gutiérrez', 'Av. Arequipa 1500', '999111222'),
('Ana Morales', 'Jr. Cuzco 345', '988776655'),
('Luis Salazar', 'Av. Brasil 900', '977665544');
GO

BACKUP DATABASE QhatuPERU
TO DISK = 'C:\Respaldo\QhatuPERU.bak'
WITH INIT;
GO

-- Eliminación accidental
DELETE FROM CLIENTES;
GO

SELECT * FROM CLIENTES;
GO
```

100 %

Results Messages

CodCliente	NomCliente	Direccion	Telefono	FechaRegistro
------------	------------	-----------	----------	---------------

b) Verifica la existencia y correcta restauración de los datos con una consulta.

Restauración después del incidente:

USE master;

GO

**ALTER DATABASE QhatuPERU SET SINGLE_USER WITH ROLLBACK
IMMEDIATE;**

GO

**RESTORE DATABASE QhatuPERU
FROM DISK = 'C:\Respaldo\QhatuPERU.bak'
WITH REPLACE;**

GO

ALTER DATABASE QhatuPERU SET MULTI_USER;

GO

Verificación de la restauración

USE QhatuPERU;

GO

SELECT * FROM CLIENTES;

GO


```
SQLQuery7.sql - L...H2M65\Megam (67))* X SQLQuery6.sql - L...H2M65\Megam (74))* S
USE master;
GO

ALTER DATABASE QhatuPERU SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO

RESTORE DATABASE QhatuPERU
FROM DISK = 'C:\Respaldo\QhatuPERU.bak'
WITH REPLACE;
GO

ALTER DATABASE QhatuPERU SET MULTI_USER;
GO

SELECT * FROM CLIENTES;
GO
```

100 %

Results Messages

	CodCliente	NomCliente	Direccion	Telefono	FechaRegistro
1	5	Carlos Gutiérrez	Av. Arequipa 1500	999111222	2025-11-18 23:20:53.330
2	6	Ana Morales	Jr. Cuzco 345	988776655	2025-11-18 23:20:53.330
3	7	Luis Salazar	Av. Brasil 900	977665544	2025-11-18 23:20:53.330

3. Justificación técnica de la solución aplicada

La restauración desde un backup es la forma profesional de recuperar datos perdidos por:

- errores humanos
- fallas del sistema
- corrupción de datos

SQL Server permite restaurar la base completa usando:

- **BACKUP DATABASE** → genera el archivo de respaldo
- **RESTORE DATABASE** → devuelve los datos al punto exacto del backup

Además:

- Se usa **SINGLE_USER** para evitar conexiones que bloqueen la restauración.
- Se usa **WITH REPLACE** para sobrescribir la base dañada.
- La verificación final asegura que los datos se restauraron correctamente.

4. Explicación de las buenas prácticas utilizadas en el proyecto

Realizar backups periódicos

Permite recuperar información ante fallos o errores humanos.

En empresas se hacen:

- diarios
- semanales

- backups diferenciales

Gardar backups en una ruta externa

Nunca deben guardarse en el mismo disco de la base para evitar pérdidas totales.

Probar la restauración

No solo basta hacer backups:

Se debe validar que se pueden recuperar correctamente.

Uso de SINGLE_USER y MULTI_USER

Evita errores de bloqueo durante la restauración.

Documentar el proceso

Permite que cualquier miembro del equipo pueda ejecutar el procedimiento durante un incidente.