

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.

The screenshot shows the SQL Server Management Studio (SSMS) interface with three tabs open: 'SQLQuery4.sql - LA...PC24\USER 17 (53)*', 'SQLQuery3.sql - LA...PC24\USER 17 (75)*', and 'SQLQueryINSERCCIO...C24\USER 17 (82)'. The central pane displays the following T-SQL script:

```
--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_Principal',
    FILENAME = N'C:\SQLData\QhatuPeru_Principal.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
```

The status bar at the bottom indicates 'Commands completed successfully.' and the completion time: '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 Server Management Studio window with two tabs: 'SQLQuery4.sql - LA...PC24\USER 17 (53)*' and 'SQLQuery3.sql - LA...PC24\USER 17 (75)*'. The code in the top tab is:

```
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
```

The results tab displays the following table:

| | NombreArchivo | RutaFisica | TipoArchivo |
|---|----------------------|-------------------------------------|-------------|
| 1 | QhatuPeru_Primario | C:\SQLData\QhatuPeru_Primario.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_Primario',
    FILENAME = N'C:\SQLData\QhatuPeru_Primario.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 shows three distinct sections of SQL code in a query editor:

- Section 1:** A query to check the current properties of the database "QhatuPeru".

```
-- 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
```
- Section 2:** A series of ALTER DATABASE statements to change the collation to "Modern_Spanish_CI_AS" and to modify the primary file to have automatic growth of 10MB.

```
-- 2@ 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

-- 3@ Configurar crecimiento automático del archivo de datos principal (10MB inicialmente)
ALTER DATABASE QhatuPeru
MODIFY FILE
(
    NAME = N'QhatuPeru_Principal',
    FILEGROWTH = 10MB
);
GO
```
- Section 3:** A query to verify the current configuration of the files for the database "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
```

Below each section is a "Results" grid showing the output of the query. The first section shows the database properties. The second section shows the database has been modified. The third section shows the current file configuration.

| BaseDatos | NombreArchivo | TipoArchivo | RutaFisica | TamañoMB | CrecimientoMB |
|-------------|----------------------|-------------|-------------------------------------|----------|---------------|
| 1 QhatuPeru | QhatuPeru_Principal | ROWS | C:\SQLData\QhatuPeru_Principal.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.

-- 5@ Cambiar crecimiento automático del archivo primario a 20MB

```
ALTER DATABASE QhatuPeru
MODIFY FILE
(
    NAME = N'QhatuPeru_Primario',
    FILEGROWTH = 20MB
);
GO

-- 6@ 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_Primario';
GO
```

100 %

| | BaseDatos | NombreArchivo | CrecimientoMB |
|---|-----------|--------------------|---------------|
| 1 | QhatuPeru | QhatuPeru_Primario | 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_Primario',
    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_Primario',  
    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_Primario';  
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 Management Studio (SSMS) window with two panes. The top pane displays T-SQL code for changing the recovery model of the 'QhatuPeru' database. The bottom pane shows the results of the executed queries.

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

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

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

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

Results pane output:

| | 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 the SQL Server Management Studio interface with two tabs open: 'SQLQuery6.sql' and 'SQLQuery5.sql'. The 'SQLQuery6.sql' tab contains the following T-SQL script:

```
-- 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' tab displays two sets of results from the 'sys.databases' system catalog. The first set shows the current recovery model for the 'QhatuPeru' database:

| | NombreBaseDatos | ModeloRecuperacion |
|---|-----------------|--------------------|
| 1 | QhatuPeru | FULL |

The second set shows the recovery model after the 'ALTER DATABASE' command was executed:

| | NombreBaseDatos | ModeloRecuperacion |
|---|-----------------|--------------------|
| 1 | QhatuPeru | BULK_LOGGED |

The third set shows the recovery model after the 'BACKUP' command was executed:

| | NombreBaseDatos | ModeloRecuperacion |
|---|-----------------|--------------------|
| 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)* + 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 asínignallo solo al rol db_datareader.

```
SQLQuery1.sql - L...H2M65\Megam (72)* + 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 auditible.

- **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)*

```

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.