

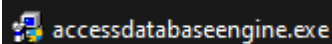
Administración de Base de Datos

Enunciado 01:

De acuerdo con la **base de datos** implementada (mínimo 100 registros), utilice los DBMS **Microsoft SQL Server/MySQL**, o un servidor de la nube como Microsoft Azure o Google FireBase. Explique qué problema soluciona su base de datos y responda las siguientes preguntas:

Desarrollo

Para exportar Excel de descargo un archivo en 32 en x32 por mas que tengas lo demás instalado en x64



Que se ejecuto en cmd administrador

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.4317]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jesuk>cd C:\Users\jesuk\Downloads\
C:\Users\jesuk\Downloads>accessdatabaseengine.exe
```

Se exporta a SQL SERVER un archivo Excel de una base de datos LocalesLima con 03 tablas siguientes:

LIMA_DIRECCION\$

Columnas (ID_LV, NOMBRE_LV, TIPO_VIA, NOMBRE_VIA)

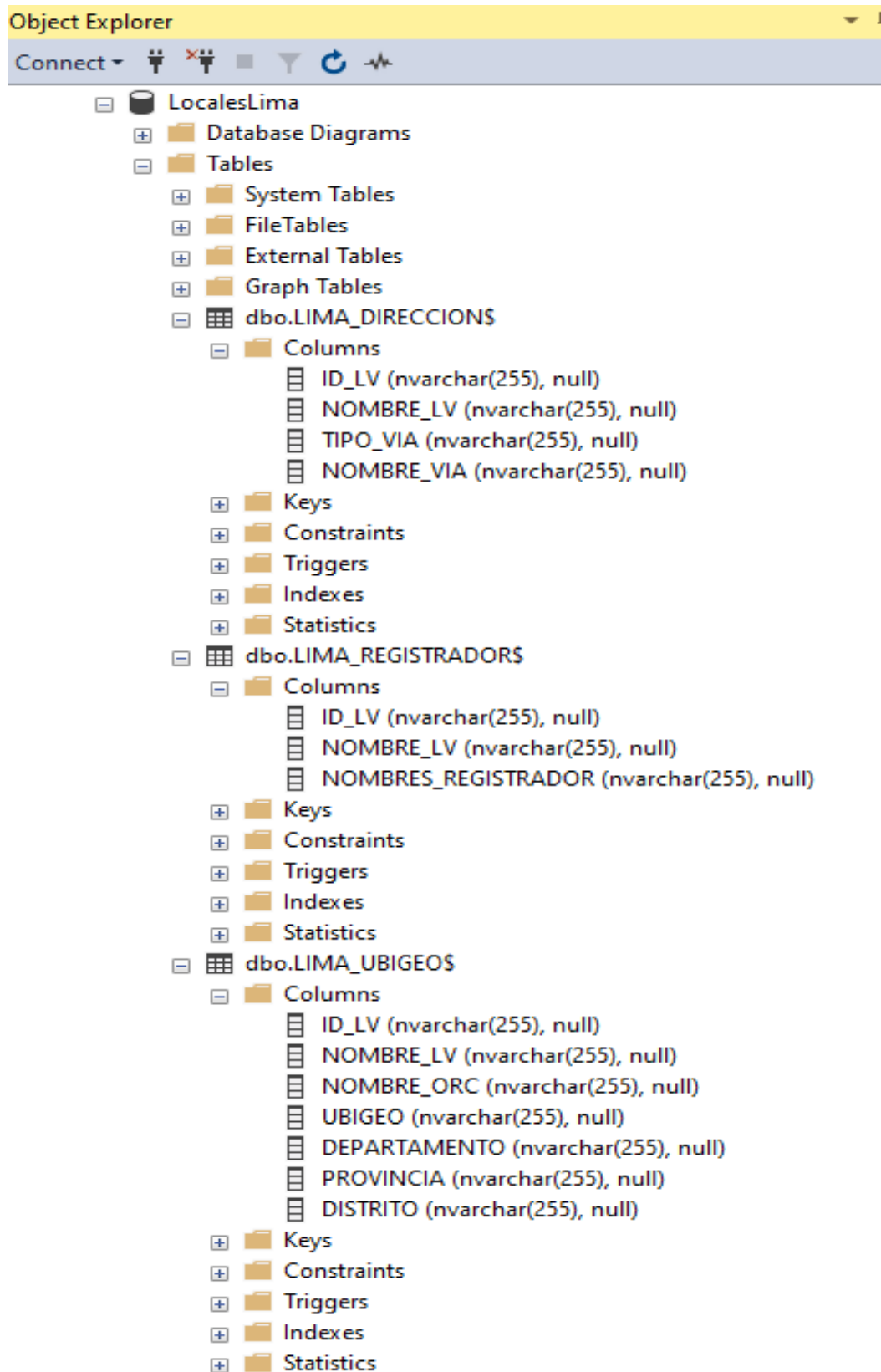
LIMA_REGISTRADOR\$

Columnas (ID_LV, NOMBRE_LV, NOMBRES_REGISTRADOR)

LIMA_UBIGEO\$

Columnas (ID_LV, NOMBRE_LV, NOMBRE_ORC, UBIGEO, DEPARTAMENTO, PROVINCIA, DISTRITO)

Tiene 302 filas



Se conecto con microsoftazure

The screenshot shows the Azure portal interface for an Azure SQL Server. The top navigation bar includes the Azure logo, a search bar, and a Copilot button. The main content area displays the 'ironserver1' SQL Server overview page. The left sidebar contains a navigation menu with options like 'Información general', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Inicio rápido', 'Diagnosticar y solucionar problemas', 'Configuración', 'Administración de datos', 'Seguridad', 'Rendimiento inteligente', 'Supervisión', 'Automation', and 'Ayuda'.

The main content area shows the server's status as 'Deshabilitado' (Disabled). Below this, there are sections for 'Notificaciones (0)' and 'Características (6)'. The 'Características' section includes several configuration cards: 'Administrador de Microsoft Entra' (NO CONFIGURADO), 'Microsoft Defender para SQL' (NO CONFIGURADO), 'Ajuste automático' (CONFIGURADO), 'Auditoría' (NO CONFIGURADO), 'Grupos de conmutación por error' (NO CONFIGURADO), and 'Cifrado de datos transparente' (NO CONFIGURADO).

At the bottom, there is a table titled 'Recursos disponibles' (Available resources) showing a list of databases. The table has columns for 'Nombre' (Name), 'Tipo' (Type), 'Estado' (Status), and 'Plan de tarifa' (Pricing plan).

Nombre	Tipo	Estado	Plan de tarifa
ventas	Base de datos SQL	Pausado	Uso general - Sin servidor: Serie estándar (Gen S), 1 vCore
ironserver1	Base de datos SQL	En línea	Free - Uso general - Sin servidor: Serie estándar (Gen S), 2 Núcleos

- 1) Implemente y explique un Script para crear una **vista** para crear utilizando tres tablas EN SQL SERVER se crea un script que crea una vista combina las tres tablas basándose en la columna común **ID_LV**, mostrando información completa de cada dirección, registrador y ubicación geográfica.

```
.sql - DESKTOP-UF...UFSFOB7\JESU (52))*
CREATE VIEW Vista_Lima AS
SELECT
    ld.ID_LV,
    ld.NOMBRE_LV,
    ld.TIPO_VIA,
    ld.NOMBRE_VIA,
    lr.NOMBRES_REGISTRADOR,
    lu.NOMBRE_ORC,
    lu.UBIGEO,
    lu.DEPARTAMENTO,
    lu.PROVINCIA,
    lu.DISTRITO
FROM
    LIMA_DIRECCIONES$ ld
JOIN
    LIMA_REGISTRADORES$ lr ON ld.ID_LV = lr.ID_LV
JOIN
    LIMA_UBIGEOS$ lu ON ld.ID_LV = lu.ID_LV;
```

Se conecto con microsoftazure

- 2) Implemente y explique un Script para crear un **procedimiento almacenado** para insertar datos a su base de datos.

Este procedimiento permite insertar una nueva dirección en la tabla LIMA_DIRECCION\$ utilizando parámetros para cada columna.

```
2.sql - DESKTOP-UF...UFSFOB7\JESU (51))  X 1.sql - DESKTOP-UF...UFSFOB7\JESU (52))*  
CREATE PROCEDURE InsertarDireccion  
    @ID_LV INT,  
    @NOMBRE_LV NVARCHAR(50),  
    @TIPO_VIA NVARCHAR(50),  
    @NOMBRE_VIA NVARCHAR(50)  
AS  
BEGIN  
    INSERT INTO LIMA_DIRECCION$ (ID_LV, NOMBRE_LV, TIPO_VIA, NOMBRE_VIA)  
    VALUES (@ID_LV, @NOMBRE_LV, @TIPO_VIA, @NOMBRE_VIA);  
END;
```

- 3) Implemente y explique un Script para crear un **procedimiento almacenado** para eliminar datos a su base de datos

Este procedimiento elimina un registro de la tabla LIMA_DIRECCION\$ identificándolo por su ID_LV.

```
3.sql - DESKTOP-UF...UFSFOB7\JESU (78))  X 2.sql - DESK  
CREATE PROCEDURE EliminarDireccion  
    @ID_LV INT  
AS  
BEGIN  
    DELETE FROM LIMA_DIRECCION$  
    WHERE ID_LV = @ID_LV;  
END;
```

- 4) Implemente y explique un Script para crear un **procedimiento almacenado** para actualizar datos a su base de datos

Este procedimiento actualiza el nombre de la vía (NOMBRE_VIA) para un registro específico.

```
4.sql - DESKTOP-UF...UFSFOB7JESU (78)) -> X
CREATE PROCEDURE ActualizarDirecci
    @ID_LV INT,
    @NOMBRE_VIA NVARCHAR(50)
AS
BEGIN
    UPDATE LIMA_DIRECCION$
    SET NOMBRE_VIA = @NOMBRE_VIA
    WHERE ID_LV = @ID_LV;
END;
```

- 5) Implemente y explique un Script para crear un **procedimiento almacenado** para realizar cálculos matemáticos de una columna de su base de datos.
Este procedimiento calcula la longitud de los nombres (NOMBRE_LV) en la tabla LIMA_DIRECCION\$.

```
5.sql - DESKTOP-UF...UFSFOB7JESU (78)) -> X 2.sql - DE
CREATE PROCEDURE CalcularLongitudNombres
AS
BEGIN
    SELECT
        ID_LV,
        LEN(NOMBRE_LV) AS Longitud_Nombre
    FROM LIMA_DIRECCION$;
END;
```

- 6) Implemente y explique un Script para crear un **disparador** para ingresar un registro automáticamente en una tabla de su base de datos.
Este disparador agrega automáticamente un registro en la tabla LIMA_UBIGEO\$ cuando se inserta un registro en LIMA_DIRECCION\$.

```
6.sql - DESKTOP-UF...UFSFOB7JESU (53)) -> X 5.sql - DESKTOP-UF...UFSFOB7JESU (78))
CREATE TRIGGER InsertarUbigeoPorDefecto
ON LIMA_DIRECCION$
AFTER INSERT
AS
BEGIN
    INSERT INTO LIMA_UBIGEO$ (ID_LV, NOMBRE_LV, UBIGEO, DEPARTAMENTO, PROVINCIA, DISTRITO)
    SELECT ID_LV, NOMBRE_LV, '000000', 'LIMA', 'LIMA', 'LIMA'
    FROM INSERTED;
END;
```

- 7) Implemente y explique un Script para crear un **disparador** para elimine un registro automáticamente en una tabla de su base de datos.
Este disparador elimina automáticamente un registro de LIMA_UBIGEO\$ cuando se elimina de LIMA_DIRECCION\$.

```
7.sql - DESKTOP-UF...UFSFOB7\JESU (62)) X 6.sql - DESKTOP
CREATE TRIGGER EliminarUbigeo
ON LIMA_DIRECCION$
AFTER DELETE
AS
BEGIN
    DELETE FROM LIMA_UBIGEO$
    WHERE ID_LV IN (SELECT ID_LV FROM DELETED);
END;
```

- 8) Implemente y explique un Script para crear un **disparador** para actualice un registro automáticamente en una tabla de su base de datos.
Este disparador actualiza el nombre de la localidad (NOMBRE_LV) en LIMA_UBIGEO\$ al actualizarse en LIMA_DIRECCION\$.

```
8.sql - DESKTOP-UF...UFSFOB7\JESU (89)) X 7.sql - DESKTOP-UF...UFSFOB7\JESU (62))
CREATE TRIGGER ActualizarUbigeoNombre
ON LIMA_DIRECCION$
AFTER UPDATE
AS
BEGIN
    UPDATE LIMA_UBIGEO$
    SET NOMBRE_LV = (SELECT NOMBRE_LV FROM INSERTED WHERE LIMA_UBIGEO$.ID_LV = INSERTED.ID_LV)
    WHERE ID_LV IN (SELECT ID_LV FROM INSERTED);
END;
```

- 9) Implemente y explique un Script para crear un **disparador** para verificar el control de datos (Ejemplo: que la nota ingresada este entre 0 y 20)
Este disparador evita que se inserten registros con un código UBIGEO que no tenga exactamente 6 dígitos.

```
9.sql - DESKTOP-UF...UFSFOB7\JESU (89)) X 7.sql - DESKTOP-UF...UFSFOB7\JESU (62))
CREATE TRIGGER VerificarUbigeo
ON LIMA_UBIGEO$
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (SELECT * FROM INSERTED WHERE LEN(UBIGEO) <> 6)
    BEGIN
        RAISERROR ('El código UBIGEO debe tener 6 dígitos.', 16, 1);
    END
    ELSE
    BEGIN
        INSERT INTO LIMA_UBIGEO$
        SELECT * FROM INSERTED;
    END
END;
```

- 10) Utilizando Script Crear 03 usuarios con nombres de sus compañeros y uno suyo
Se crean tres usuarios (Edwin, Luis, Brayan) y se les asigna el rol de lectura (db_datareader) en la base de datos.

```
SQLQuery8.sql - DE...UFSFOB7\JESU (57))* -p X 9.sql - DESKTOP-UF...UFSFO
CREATE LOGIN Edwin WITH PASSWORD = 'Password123';
CREATE LOGIN Luis WITH PASSWORD = 'Password123';
CREATE LOGIN Brayan WITH PASSWORD = 'Password123';

CREATE USER Edwin FOR LOGIN Edwin;
CREATE USER Luis FOR LOGIN Luis;
CREATE USER Brayan FOR LOGIN Brayan;

ALTER ROLE db_datareader ADD MEMBER Edwin;
ALTER ROLE db_datareader ADD MEMBER Luis;
ALTER ROLE db_datareader ADD MEMBER Brayan;
```

- 11) Utilizando un script, copiar la base de datos (creada anteriormente) y compartir en cada uno de los usuarios

```
11.sql - DESKTOP-U...UFSFOB7\JESU (80)) -p X 10.sql - DESKTOP-U...UFSFOB7\JESU (57))
USE master;

-- Crear una nueva base de datos a partir de LocalesLima
BACKUP DATABASE LocalesLima TO DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima.bak';

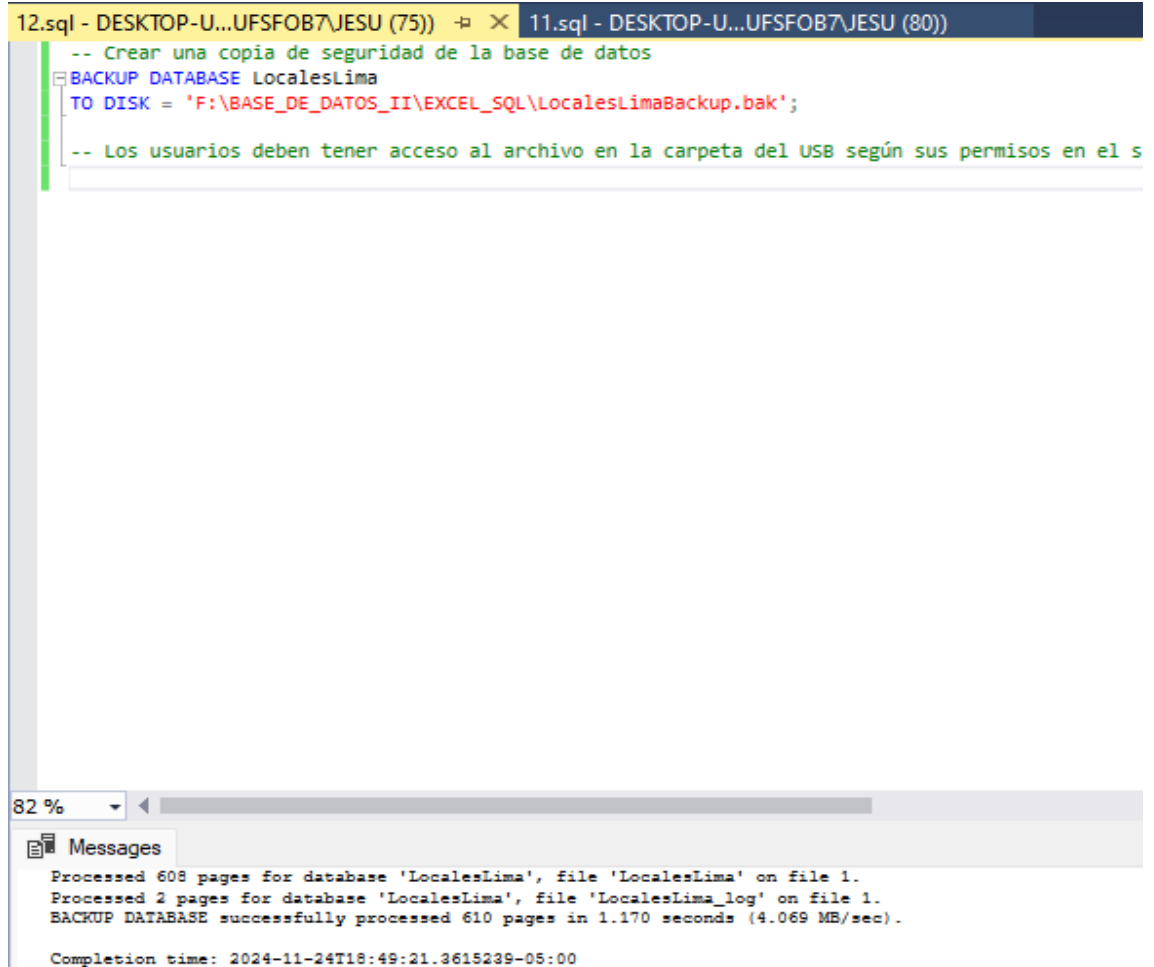
RESTORE DATABASE LocalesLima_Copia
FROM DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima.bak'
WITH MOVE 'LocalesLima' TO 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima_Copia.mdf',
MOVE 'LocalesLima_log' TO 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima_Copia_log.ldf';

-- Compartir con los usuarios
USE LocalesLima_Copia;

CREATE USER Edwin FOR LOGIN Edwin;
CREATE USER Luis FOR LOGIN Luis;
CREATE USER Brayan FOR LOGIN Brayan;

ALTER ROLE db_datareader ADD MEMBER Edwin;
ALTER ROLE db_datareader ADD MEMBER Luis;
ALTER ROLE db_datareader ADD MEMBER Brayan;
```

- 12) Utilizando un script, generar una copia de seguridad de la base de datos y compartir a cada uno de los usuarios



The screenshot shows a SQL Server Enterprise Manager window with two tabs: '12.sql - DESKTOP-U...UFSFOB7\JESU (75))' and '11.sql - DESKTOP-U...UFSFOB7\JESU (80))'. The active tab '12.sql' contains the following SQL script:

```
-- Crear una copia de seguridad de la base de datos
BACKUP DATABASE LocalesLima
TO DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLimaBackup.bak';

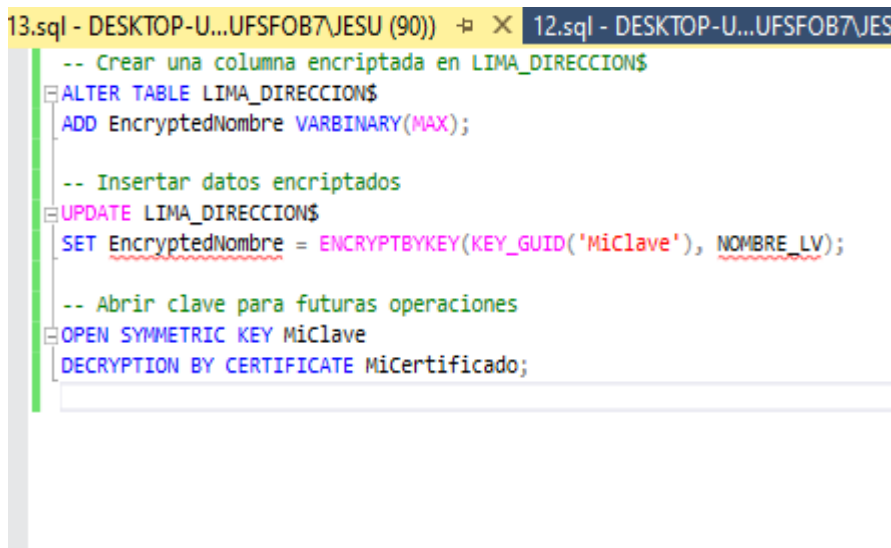
-- Los usuarios deben tener acceso al archivo en la carpeta del USB según sus permisos en el s
```

Below the script, the 'Messages' pane displays the execution results:

```
Processed 608 pages for database 'LocalesLima', file 'LocalesLima' on file 1.
Processed 2 pages for database 'LocalesLima', file 'LocalesLima_log' on file 1.
BACKUP DATABASE successfully processed 610 pages in 1.170 seconds (4.069 MB/sec).

Completion time: 2024-11-24T18:49:21.3615239-05:00
```

- 13) Utilizando un script, encriptar una de las tablas para que no se puedan ver los datos. Este script utiliza una clave simétrica para encriptar la columna NOMBRE_LV. La clave debe estar abierta para que se puedan realizar las operaciones de encriptación y desencriptación.



The screenshot shows a SQL Server Enterprise Manager window with two tabs: '13.sql - DESKTOP-U...UFSFOB7\JESU (90))' and '12.sql - DESKTOP-U...UFSFOB7\JESU (80))'. The active tab '13.sql' contains the following SQL script:

```
-- Crear una columna encriptada en LIMA_DIRECCION$
ALTER TABLE LIMA_DIRECCION$
ADD EncryptedNombre VARBINARY(MAX);

-- Insertar datos encriptados
UPDATE LIMA_DIRECCION$
SET EncryptedNombre = ENCRYPTBYKEY(KEY_GUID('MiClave'), NOMBRE_LV);

-- Abrir clave para futuras operaciones
OPEN SYMMETRIC KEY MiClave
DECRYPTION BY CERTIFICATE MiCertificado;
```

- 14) Utilizando un script, aplique la seguridad a nivel de columna, restringiendo el acceso a la columna que contiene la clave primaria de una de las tablas de su base de datos


```
14.sql - DESKTOP-U...UFSFOB7\JESU (91))  X 13.sql - DESKTOP-U...UFSFOB7\JESU (90))
-- Crear una máscara para la columna clave primaria
CREATE SECURITY POLICY LimitarClavePrimaria
ADD FILTER PREDICATE
    fn_IsUserAuthorized() ON LIMA_DIRECCION$(ID_LV);

-- Implementar la función para verificar permisos
CREATE FUNCTION fn_IsUserAuthorized()
RETURNS TABLE
WITH SCHEMABINDING AS
RETURN SELECT 1 AS Authorized WHERE USER_NAME() NOT IN ('Edwin', 'Luis', 'Brayan');
```

- 15) Utilizando un script, implementé seguridad a nivel de columna restringiendo el acceso a una de las columnas de una tabla.

```
15.sql - DESKTOP-U...UFSFOB7\JESU (74))  X 14.sql - DESKTOP-U...UFSFOB7\JESU (91))
-- Crear una máscara en otra columna
ALTER TABLE LIMA_DIRECCION$
ALTER COLUMN TIPO_VIA ADD MASKED WITH (FUNCTION = 'partial(0,"***",0)');
```

- 16) Utilizando un script, realice el cifrado transparente de datos (TDE) para una las tablas.

```
SQLQuery14.sql - D...UFSFOB7\JESU (87))*  X 15.sql - DESKTOP-U...UFSFOB7\JESU
-- Habilitar TDE en la base de datos
USE master;

-- Crear una clave maestra
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'ContraseñaSegura123!';

-- Crear un certificado para el cifrado
CREATE CERTIFICATE MiCertificadoTDE
WITH SUBJECT = 'Certificado para TDE';

-- Crear una clave de cifrado de la base de datos
USE LocalesLima;

CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE MiCertificadoTDE;

-- Activar TDE
ALTER DATABASE LocalesLima
SET ENCRYPTION ON;
```

- 17) Utilizando un script, configure el usuario con el nombre de su compañero para otorgar permisos de SELECT, INSERT, UPDATE y DELETE en la base de datos.

```
17.sql - DESKTOP-U...UFSFOB7\JESU (92))  X 16.sql - DESKTOP-U...UFSFOB7\JESU
USE LocalesLima;

-- Otorgar permisos al usuario
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::LocalesLima TO Edwin;
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::LocalesLima TO Luis;
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::LocalesLima TO Brayan;
```

- 18) Utilizando un Scripts realice la validación y filtración de entradas del usuario para evitar caracteres maliciosos (Ejemplo: ', --, ;)

```
18.sql - DESKTOP-U...UFSFOB7\JESU (93))  X 17.sql - DESKTOP-U...UFSFOB7\JESU
-- Ejemplo de procedimiento con validación
CREATE PROCEDURE InsertarDireccionValidada
    @ID_LV INT,
    @NOMBRE_LV NVARCHAR(50)
AS
BEGIN
    IF @NOMBRE_LV LIKE '%[;--]%'
    BEGIN
        RAISERROR('Entrada inválida detectada.', 16, 1);
        RETURN;
    END

    INSERT INTO LIMA_DIRECCION$ (ID_LV, NOMBRE_LV)
    VALUES (@ID_LV, @NOMBRE_LV);
END;
```

- 19) Realice un script que verifiquen que los datos ingresados cumplan con formatos esperados (ej.: números en lugar de texto, longitud máxima).

```
19.sql - DESKTOP-U...UFSFOB7\JESU (94))  X 18.sql - DESKTOP-U...UFSFOB7\JESU
CREATE PROCEDURE InsertarDireccionConFormato
    @ID_LV INT,
    @NOMBRE_LV NVARCHAR(50)
AS
BEGIN
    IF NOT ISNUMERIC(@ID_LV) = 1
    BEGIN
        RAISERROR('El ID debe ser numérico.', 16, 1);
        RETURN;
    END

    IF LEN(@NOMBRE_LV) > 50
    BEGIN
        RAISERROR('El nombre supera el máximo permitido.', 16, 1);
        RETURN;
    END

    INSERT INTO LIMA_DIRECCION$ (ID_LV, NOMBRE_LV)
    VALUES (@ID_LV, @NOMBRE_LV);
END;
```

- 20) Utilizando un script, configure la auditoría para el seguimiento y registro de acciones en la base de datos

```
20.sql - DESKTOP-U...UFSFOB7\JESU (54))  X 19.sql - DESKTOP-U...UFSFOB7\JESU (94))
-- Crear auditoría a nivel de servidor
CREATE SERVER AUDIT AuditoriaGeneral
TO FILE (FILEPATH = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\;', MAXSIZE = 10 MB);

-- Activar auditoría
ALTER SERVER AUDIT AuditoriaGeneral
WITH (STATE = ON);

-- Crear una especificación de auditoría para una base de datos
USE LocalesLima;

CREATE DATABASE AUDIT SPECIFICATION AuditoriaDB
FOR SERVER AUDIT AuditoriaGeneral
ADD (SELECT ON SCHEMA::dbo BY [Edwin]);

-- Activar especificación de auditoría
ALTER DATABASE AUDIT SPECIFICATION AuditoriaDB
WITH (STATE = ON);
```

- 21) Utilizando un script, configure de la memoria y el disco duro
Configuración mínima y máxima de la memoria

```
21.sql - DESKTOP-U...UFSFOB7\JESU (65))  X 20.sql - DESKTOP-U...UFSFOB
-- Configurar memoria mínima y máxima en MB
EXEC sp_configure 'min server memory', 4096; -- 4 GB mínimo
EXEC sp_configure 'max server memory', 16384; -- 16 GB máximo
RECONFIGURE;
```

El manejo del disco se realiza configurando las bases de datos y sus archivos de datos (archivo .mdf) y archivos de log (.ldf).

```
SQLQuery20.sql - D...UFSFOB7\JESU (99))*  X 21.sql - DESKTOP-U...UFSFOB7\JESU (65))
-- Agregar un nuevo archivo de datos a la base de datos
ALTER DATABASE LocalesLima
ADD FILE (NAME = 'LocalesLima_Data2', FILENAME = 'D:\SQLData\LocalesLima2.mdf', SIZE = 5GB);

-- Agregar un archivo de log
ALTER DATABASE LocalesLima
ADD LOG FILE (NAME = 'LocalesLima_Log2', FILENAME = 'D:\SQLLogs\LocalesLima2_log.ldf', SIZE = 2GB);
```

- 22) Utilizando un script, genere una copia de seguridad de la base de datos

The screenshot shows a SQL Server Enterprise Manager window with two tabs: '22.sql - DESKTOP-U...FSFOB7\JESU (100))' and '21_1.sql - DESKTOP...UFSFOB7\JESU (99))'. The active tab contains the following SQL script:

```
BACKUP DATABASE LocalesLima
TO DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima.bak'
WITH FORMAT, INIT;
```

Below the script, the 'Messages' pane shows the execution results:

```
Processed 616 pages for database 'LocalesLima', file 'LocalesLima' on file 1.
Processed 2 pages for database 'LocalesLima', file 'LocalesLima_log' on file 1.
BACKUP DATABASE successfully processed 618 pages in 0.581 seconds (8.303 MB/sec).
Completion time: 2024-11-24T19:01:40.4852963-05:00
```

- 23) Realice un script para programar backups automatizados de su base de datos
La copia de seguridad se programa utilizando SQL Server Agent y un job. Aquí te muestro cómo crear un trabajo que haga un respaldo diario:

- Abre SQL Server Management Studio (SSMS).
- Ve a SQL Server Agent -> Jobs -> New Job.

En el paso del trabajo, ingresa el siguiente código:

- The screenshot shows a SQL Server Enterprise Manager window with two tabs: '23.sql - DESKTOP-U...FSFOB7\JESU (100))' and '21_1.sql - DESKTOP...UF'. The active tab contains the following SQL script:

```
BACKUP DATABASE LocalesLima
TO DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima.bak'
WITH FORMAT, INIT;
```

- 24) Utilizando un script, genere la restauración de la base de datos
El comando WITH REPLACE sobrescribirá cualquier base de datos existente con el mismo nombre.

```
24.sql - DESKTOP-U...UFSFOB7JESU (68))  X 23.sql - DESKTOP-U...UFSFOB7JESU (68))
RESTORE DATABASE LocalesLima
FROM DISK = 'F:\BASE_DE_DATOS_II\EXCEL_SQL\LocalesLima.bak'
WITH REPLACE;
```

- 25) Utilizando un script, cree un espejo de la base de datos
El espejo de base de datos en SQL Server se puede crear con Database Mirroring. Para ello, necesitas dos servidores (principal y espejo).

Comando para configurar el espejo:

En el servidor principal:

```
25.sql - DESKTOP-U...UFSFOB7JESU (106))  X 24.sql - DESKTOP-U...UFSFOB7JESU (106))
ALTER DATABASE LocalesLima SET PARTNER = 'TCP://ServidorEspejo:5022';
```

Ambos servidores deben estar en modo de Full Recovery.

- 26) Utilizando un script, para enviar datos a la base de datos espejo creada
Envío de datos: En el servidor principal, cuando insertemos datos en la base de datos principal (por ejemplo, la tabla LIMA_DIRECCION\$), esos datos se replicarán automáticamente en la base de datos espejo debido a la configuración de Database Mirroring.

```
26.sql - DESKTOP-U...UFSFOB7JESU (85))  X 28.sql - DESKTOP-U...UFSFOB7JESU (85))
INSERT INTO LIMA_DIRECCION$ (ID_LV, NOMBRE_LV, TIPO_VIA, NOMBRE_VIA)
VALUES (1, 'Avenida Siempre Viva', 'Calle', '123');
```

300	UNIVERSIDAD NACIONAL DE BARRANCA	AVENIDA	AV TORIBIO LUZURIAGA 376
301	VENTURA CCALAMAQUI	JIRÓN	SAENZ PEÑA SUR
302	VIRGEN DEL CARMEN	AVENIDA	PAMPA DE LARA SN
303	Avenida Siempre Viva	Calle	123

- 27) Utilizando un script, de permiso a un usuario por un determinado tiempo
Puedes otorgar permisos temporales a un usuario utilizando SQL Server Agent para programar un trabajo. Para hacerlo manualmente, puedes establecer un Tiempo de Expiración para el permiso:

```
27.sql - DESKTOP-U...UFSFOB7JESU (71))  X 25.sql - DESKTOP-U...UFSFOB7JESU (71))
GRANT SELECT ON dbo.LIMA_DIRECCION$ TO Usuario;

WAITFOR DELAY '24:00:00';
REVOKE SELECT ON dbo.LIMA_DIRECCION$ FROM Usuario;
```

- 28) Utilizando un script, realice la replicación de bases de datos
Para configurar la replicación de bases de datos en SQL Server, sigue estos pasos:

Configura el servidor de publicación.

Configura el servidor de suscripción.

Usa el siguiente comando para inicializar la replicación:

```
28.sql - DESKTOP-U...UFSFOB7\JESU (55)) -p X 2
-- Configurar la publicación
EXEC sp_replicationdboption
@dbname = 'LocalesLima',
@optname = 'publish',
@value = 'true';

-- Crear un artículo para replicar
EXEC sp_addarticle
@publication = 'MiPublicacion',
@article = 'LIMA_DIRECCION$',
@source_owner = 'dbo',
@source_object = 'LIMA_DIRECCION$';
```

29) Explique que es Always On Availability Groups

Always On Availability Groups es una solución de alta disponibilidad y recuperación ante desastres en SQL Server. Permite tener múltiples réplicas de una base de datos en diferentes servidores, donde una réplica actúa como primaria y las otras como secundarias. Estas réplicas pueden ser de solo lectura (si se desea), y los cambios en la base de datos primaria se sincronizan automáticamente con las secundarias. Esto asegura alta disponibilidad y recuperación rápida en caso de fallos.

30) Explique que es Log Shipping

Log Shipping es una técnica de recuperación ante desastres en SQL Server que involucra la copia de los archivos de registro de transacciones desde una base de datos primaria a una secundaria, generalmente en otro servidor. La base de datos primaria realiza una copia de seguridad de sus registros de transacciones, los copia a un servidor secundario, y luego los restaura en la base de datos secundaria. Esto proporciona una forma de recuperación ante desastres, aunque no es tan rápida como los grupos de disponibilidad Always On, ya que depende de la frecuencia de los backups y restauraciones de los registros.
