 DOMINGO SAVIO UNIVERSIDAD PRIVADA	SEDE LA PAZ		LABORATORIO N° 3
	ACTIVIDAD - LABORATORIOS		
	FACULTAD:INGENIERIA		NOTA
	CARRERA: INGENIERIA DE SISTEMAS		
	ASIGNATURA: TECNOLOGIA DE BASE DE DATOS		
	DOCENTE: ING. MARIANELA L. SALAS FLORES		
	ESTUDIANTE:LUIS CARLOS FLORES DIAZ		Folio :
	FECHA: 26/08/2025	FIRMA ESTUDIANTE:	

MIGRACION DE BASE DE DATOS

Tomando en cuenta el material proporcionado en la plataforma realizar:

a) Empleando Encriptación de copias realizar

□ Crea una Master Key y el certificado llamado

CertificadoSupermercado. de la base de datos Supermercado para habilitar la encriptación

```
/* ① CREAR Y VERIFICAR DMK + CERTIFICADO EN
SUPERMERCADO */
USE Supermercado;
GO
```

```
DECLARE @DMKPassword NVARCHAR(128) =
N'P@ssw0rd_DMK_Supermercado_2025!';
```

-- 1.1) Crear la DMK si no existe (con SQL dinámico porque no admite variable directa)

```
IF NOT EXISTS (SELECT 1 FROM sys.symmetric_keys WHERE name =
'##MS_DatabaseMasterKey##')
```

```
BEGIN
```

```
PRINT 'Creando MASTER KEY en Supermercado...';
```

```
DECLARE @sql_create_dmk NVARCHAR(MAX) =
```

```
N'CREATE MASTER KEY ENCRYPTION BY PASSWORD = N'''
```

```
+ REPLACE(@DMKPassword, '"', '""') + N'''; -- escapar comillas
```

```
EXEC (@sql_create_dmk);
```

```
END
```

```
ELSE
```

```
PRINT 'La MASTER KEY ya existe en Supermercado.';
```

-- 1.2) Crear el certificado si no existe

```
IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name =
N'CertificadoSupermercado')
```

```
BEGIN
```

```
PRINT 'Creando certificado CertificadoSupermercado en Supermercado...';
```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```

CREATE CERTIFICATE CertificadoSupermercado
WITH SUBJECT = N'Certificado para cifrado (demo) en BD
Supermercado';
END
ELSE
PRINT 'El certificado CertificadoSupermercado ya existe en
Supermercado.';

```

```

-- 1.3) VERIFICACIÓN (listados)
SELECT 'DMK Supermercado' AS Objeto, name, create_date
FROM sys.symmetric_keys
WHERE name = '##MS_DatabaseMasterKey##';

```

```

SELECT 'Certificado' AS Objeto, name, subject,
pvt_key_encryption_type_desc, expiry_date
FROM sys.certificates
WHERE name = N'CertificadoSupermercado';
GO

```

Objeto	name	create_date
1 DMK Supermercado	##MS_DatabaseMasterKey##	2025-08-26 19:47:25.270

Objeto	name	subject	pvt_key_encryption_type_desc	expiry_date
1 Certificado	CertificadoSupermercado	Certificado para cifrado (demo) en BD Supermercado	ENCRYPTED_BY_MASTER_KEY	2026-08-27 00:06:35.000

- Desplegar la forma de cómo abrir la clave para usarla y cerrarla después.

```

/* ② ABRIR Y CERRAR LA MASTER KEY (DEMO + VERIFICACIÓN) */
USE Supermercado;
GO

```

```

DECLARE @DMKPassword NVARCHAR(128) =
N'P@ssw0rd_DMK_Supermercado_2025!';

```

```

BEGIN TRY
-- Abrir la DMK (SQL dinámico para permitir variable)
DECLARE @sql_open NVARCHAR(MAX) =
N'OPEN MASTER KEY DECRYPTION BY PASSWORD = N'
+ REPLACE(@DMKPassword, '"', '""') + N'";
EXEC (@sql_open);
PRINT 'DMK de Supermercado ABIERTA.';

```

```

/* (Opcional) Habilitar apertura automática si no lo está:
cifra además la DMK con la Service Master Key (SMK) del servidor */
IF NOT EXISTS (

```

ESTUDIANTE: LUIS CARLOS
FLORES DIAZ

FIRMA ESTUDIANTE:

FOLIO

```

SELECT 1
FROM sys.key_encryptions AS ke
JOIN sys.symmetric_keys AS sk
    ON ke.key_id = sk.symmetric_key_id -- <== join correcto
WHERE sk.name = '##MS_DatabaseMasterKey##'
    AND ke.crypt_type_desc = 'SERVICE MASTER KEY'
)
BEGIN
    ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY;
    PRINT 'DMK ahora también protegida por la Service Master Key.';
END

-- Cerrar la DMK
CLOSE MASTER KEY;
PRINT 'DMK de Supermercado CERRADA.';
END TRY
BEGIN CATCH
    PRINT 'No se pudo abrir/cerrar la DMK con la contraseña indicada.';
    PRINT ERROR_MESSAGE();
END CATCH;
GO

/* 2.1) VERIFICACIÓN: ver cómo está protegida la DMK */
USE Supermercado;
GO
SELECT
    sk.name AS DMK,
    ke.crypt_type_desc AS Proteccion -- Debes ver filas con 'PASSWORD' y, si aplicaste,
    también 'SERVICE MASTER KEY'
FROM sys.symmetric_keys AS sk
JOIN sys.key_encryptions AS ke
    ON ke.key_id = sk.symmetric_key_id
WHERE sk.name = '##MS_DatabaseMasterKey##';

```

	DMK	Proteccion
1	##MS_DatabaseMasterKey##	ENCRYPTION BY MASTER KEY
2	##MS_DatabaseMasterKey##	ENCRYPTION BY PASSWORD V2

b) Crear backup de la base de datos de su proyecto:

- Modo Completo
- Modo Diferencial
- Y realizar un Procedimiento denominado **“Respaldo”**

Una vez creada el backup de la base de datos de su proyecto, eliminarla y realizar:

La restauración completa de su BD

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

BACKUP COMPLETO

/* QUERY 1 — BACKUP COMPLETO (FULL) de YorchDB */

```
DECLARE @DbName SYSNAME = N'YorchDB';
DECLARE @Ruta NVARCHAR(4000) = N'C:\Backups';
DECLARE @ts NVARCHAR(20) =
REPLACE(REPLACE(CONVERT(NVARCHAR(19),GETDATE(),120),'-',''),':','');

DECLARE @FullFile NVARCHAR(4000) = @Ruta + N'\ ' + @DbName + N'_FULL_' + @ts
+ N'.bak';

-- Si existe el certificado en master, agrega cifrado AES_256
DECLARE @enc NVARCHAR(200) =
CASE WHEN EXISTS (SELECT 1 FROM master.sys.certificates WHERE name =
N'CertificadoSupermercado')
THEN N', ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE =
CertificadoSupermercado)'
ELSE N'' END;

DECLARE @sqlFull NVARCHAR(MAX) =
N'BACKUP DATABASE ' + QUOTENAME(@DbName) +
N' TO DISK = N''' + @FullFile + N''' WITH INIT, COMPRESSION, CHECKSUM,
STATS=5' + @enc + N'';

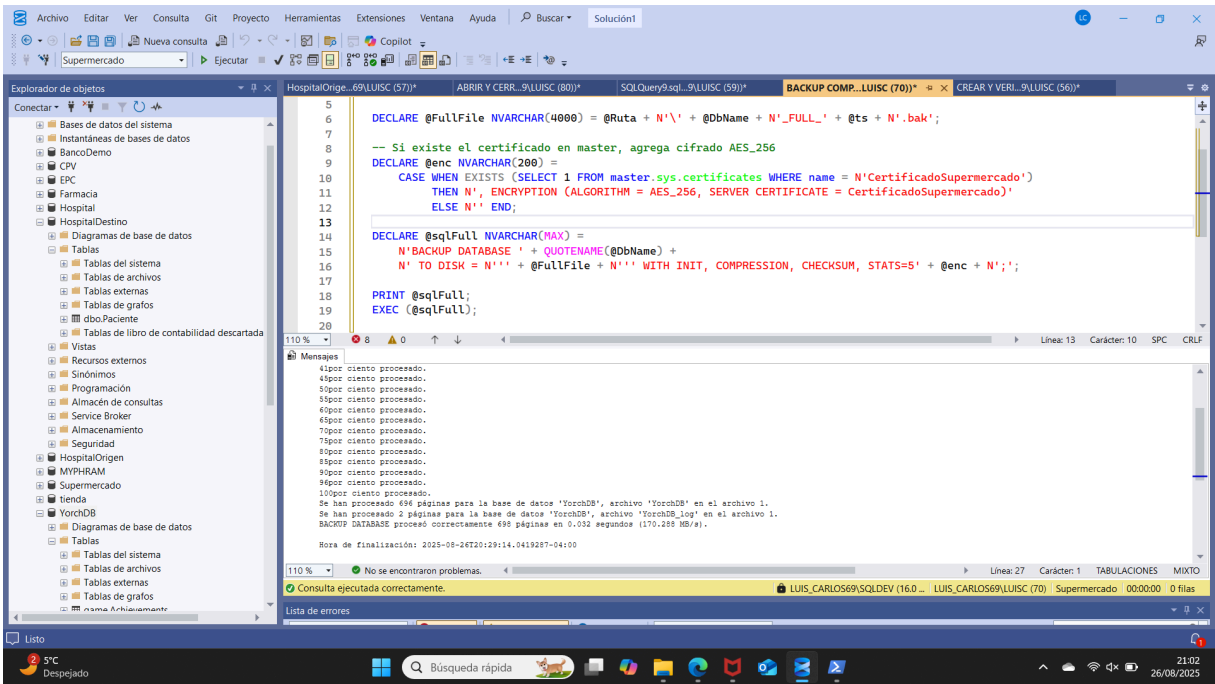
PRINT @sqlFull;
EXEC (@sqlFull);
```

-- Si existe el certificado en master, agrega cifrado AES_256

```
DECLARE @enc NVARCHAR(200) =
CASE WHEN EXISTS (SELECT 1 FROM master.sys.certificates WHERE name =
N'CertificadoSupermercado')
THEN N', ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE =
CertificadoSupermercado)'
ELSE N'' END;
```

```
DECLARE @sqlFull NVARCHAR(MAX) =
N'BACKUP DATABASE ' + QUOTENAME(@DbName) +
N' TO DISK = N''' + @FullFile + N''' WITH INIT, COMPRESSION, CHECKSUM,
STATS=5' + @enc + N'';
```

```
PRINT @sqlFull;
EXEC (@sqlFull);
```



BACKUP DIFF

/* QUERY 2 — Crear/actualizar PROC master.dbo.Respaldo y ejecutar un DIFF de YorchDB */

USE master;

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
-------------------------------------	-------------------	-------

```

GO
CREATE OR ALTER PROCEDURE dbo.Respaldo
    @DbName SYSNAME,
    @Tipo VARCHAR(10) = 'FULL',      -- 'FULL' o 'DIFF'
    @Ruta NVARCHAR(4000) = N'C:\Backups'
AS
BEGIN
    SET NOCOUNT ON;

    IF @Tipo NOT IN ('FULL','DIFF')
    BEGIN
        RAISERROR('Tipo debe ser FULL o DIFF', 16, 1);
        RETURN;
    END;

    DECLARE @ts NVARCHAR(20) =
    REPLACE(REPLACE(CONVERT(NVARCHAR(19),GETDATE(),120),'-',''),':','');
    DECLARE @file NVARCHAR(4000) = @Ruta + N'\ ' + @DbName + N'_ ' +
        CASE WHEN @Tipo='FULL' THEN 'FULL' ELSE 'DIFF' END +
        N'_ ' + @ts + N'.bak';

    DECLARE @enc NVARCHAR(200) =
        CASE WHEN EXISTS (SELECT 1 FROM master.sys.certificates WHERE name =
        N'CertificadoSupermercado')
            THEN N', ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE =
        CertificadoSupermercado)'
            ELSE N'' END;

    DECLARE @sql NVARCHAR(MAX) =
        N'BACKUP DATABASE ' + QUOTENAME(@DbName) +
        N' TO DISK = N''' + @file + N''' WITH ' +
        CASE WHEN @Tipo='FULL' THEN N'INIT, ' ELSE N'DIFFERENTIAL, ' END +
        N'COMPRESSION, CHECKSUM, STATS=5' + @enc + N'';

    PRINT @sql;
    EXEC (@sql);
END;
GO

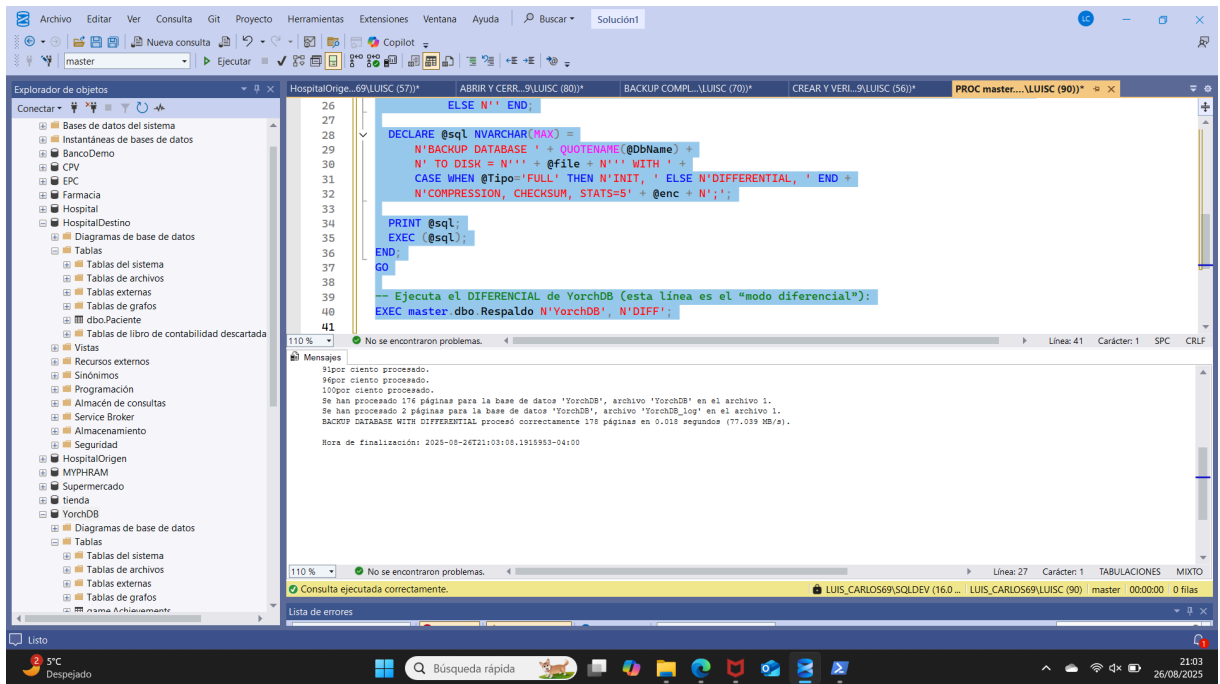
-- Ejecuta el DIFERENCIAL de YorchDB (esta línea es el “modo diferencial”):
EXEC master.dbo.Respaldo N'YorchDB', N'DIFF';

```

ESTUDIANTE: LUIS CARLOS
FLORES DIAZ

FIRMA ESTUDIANTE:

FOLIO



ELIMINAR BDD:

```
/* QUERY 3 — ELIMINAR (DROP) la BD YorchDB de forma segura (CORREGIDO) */
USE master;
GO
DECLARE @DbName SYSNAME = N'YorchDB';
```

```
IF DB_ID(@DbName) IS NOT NULL
BEGIN
```

```
    DECLARE @sql1 NVARCHAR(MAX) = N'ALTER DATABASE ' +
    QUOTENAME(@DbName)
    + N' SET SINGLE_USER WITH ROLLBACK IMMEDIATE;';
    DECLARE @sql2 NVARCHAR(MAX) = N'DROP DATABASE ' +
    QUOTENAME(@DbName) + N';';
```

```
    PRINT @sql1; EXEC (@sql1);
    PRINT @sql2; EXEC (@sql2);
```

```
    PRINT N'Base de datos eliminada: ' + @DbName;
END
```

```
ELSE
BEGIN
```

```
    PRINT N'La base de datos no existe: ' + @DbName;
END;
```

RESTAURAR BDD

```
/* QUERY 4 — RESTAURAR YorchDB desde el último FULL (+ DIFF si existe) */
USE master;
GO
```

```
DECLARE @DbName SYSNAME = N'YorchDB';
```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```

-- 1) Buscar último FULL en msdb
DECLARE @FullBak NVARCHAR(4000), @BaseLSN NUMERIC(25,0);
SELECT TOP (1)
    @FullBak = bmf.physical_device_name,
    @BaseLSN = bs.checkpoint_lsn
FROM msdb.dbo.backupset bs
JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
WHERE bs.database_name = @DbName AND bs.type = 'D' -- D = FULL
ORDER BY bs.backup_finish_date DESC;

IF @FullBak IS NULL
BEGIN
    RAISERROR('No se encontró backup FULL para %s en msdb.', 16, 1, @DbName);
    RETURN;
END;

-- 2) Buscar DIFF compatible con ese FULL (si existe)
DECLARE @DiffBak NVARCHAR(4000) = NULL;
SELECT TOP (1)
    @DiffBak = bmf.physical_device_name
FROM msdb.dbo.backupset bs
JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
WHERE bs.database_name = @DbName
    AND bs.type = 'I' -- I = DIFF
    AND bs.differential_base_lsn = @BaseLSN
ORDER BY bs.backup_finish_date DESC;

-- 3) Restaurar
IF @DiffBak IS NULL
BEGIN
    -- Solo FULL
    DECLARE @sqlRestoreFull NVARCHAR(MAX) =
        N'RESTORE DATABASE ' + QUOTENAME(@DbName) +
        N' FROM DISK = N''' + @FullBak + N''' WITH REPLACE, RECOVERY, STATS=5;';
    PRINT @sqlRestoreFull;
    EXEC (@sqlRestoreFull);
END
ELSE
BEGIN
    -- FULL + DIFF
    DECLARE @sqlRestoreFullNoRec NVARCHAR(MAX) =
        N'RESTORE DATABASE ' + QUOTENAME(@DbName) +
        N' FROM DISK = N''' + @FullBak + N''' WITH REPLACE, NORECOVERY, STATS=5;';
    DECLARE @sqlRestoreDiff NVARCHAR(MAX) =
        N'RESTORE DATABASE ' + QUOTENAME(@DbName) +
        N' FROM DISK = N''' + @DiffBak + N''' WITH RECOVERY, STATS=5;';

    PRINT @sqlRestoreFullNoRec;

```

ESTUDIANTE: LUIS CARLOS
FLORES DIAZ

FIRMA ESTUDIANTE:

FOLIO

```
PRINT @sqlRestoreDiff;
EXEC (@sqlRestoreFullNoRec);
EXEC (@sqlRestoreDiff);
END;

-- 4) Verificaciones rápidas
PRINT '--- Backups registrados en msdb ---';
SELECT TOP (10)
    bs.database_name, bs.backup_start_date,
    CASE bs.type WHEN 'D' THEN 'FULL' WHEN 'T' THEN 'DIFF' ELSE bs.type END AS
    Tipo,
    bmf.physical_device_name
FROM msdb.dbo.backupset bs
JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
WHERE bs.database_name = @DbName
ORDER BY bs.backup_start_date DESC;

PRINT '--- Conteo de tablas tras la restauración ---';
DECLARE @sqlVer NVARCHAR(MAX) =
    N'USE ' + QUOTENAME(@DbName) + N'; SELECT COUNT(*) AS Tablas FROM
    sys.tables;';
EXEC (@sqlVer);
```

Resultados

Mensajes

	database_name	backup_start_date	Tipo	physical_device_name
1	YorchDB	2025-08-26 20:22:23.000	DIFF	C:\Backups\YorchDB_DIFF_20250826 202223.bak
2	YorchDB	2025-08-26 20:21:42.000	FULL	C:\Backups\YorchDB_FULL_20250826 202142.bak
3	YorchDB	2025-08-26 20:17:14.000	FULL	C:\Backups\YorchDB_FULL_20250826 201714.bak
4	YorchDB	2025-08-26 20:17:14.000	DIFF	C:\Backups\YorchDB_DIFF_20250826 201714.bak

	Tablas
1	11

c) Copia la tabla Paciente desde HospitalOrigen hacia HospitalDestino, pero solo la estructura, Inserta todos los pacientes desde HospitalOrigen hacia

```
QUERY

HospitalDestino.

/*
=====
=
```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
-------------------------------------	-------------------	-------

VERIFICACIÓN: HospitalOrigen.dbo.Paciente ->
HospitalDestino.dbo.Paciente

Estructura + Datos + Muestras presentables

=====

DECLARE @Origen SYSNAME = N'HospitalOrigen';

DECLARE @Destino SYSNAME = N'HospitalDestino';

DECLARE @Esquema SYSNAME = N'dbo';

DECLARE @Tabla SYSNAME = N'Paciente';

DECLARE @TopN INT = 10; -- filas a mostrar de ejemplo

-- ----- Limpieza de temporales (re-ejecutable) -----

IF OBJECT_ID('tempdb..#S') IS NOT NULL DROP TABLE #S;

IF OBJECT_ID('tempdb..#D') IS NOT NULL DROP TABLE #D;

IF OBJECT_ID('tempdb..#EstructuraDif') IS NOT NULL DROP TABLE
#EstructuraDif;

IF OBJECT_ID('tempdb..#FaltanEnDestino') IS NOT NULL DROP TABLE
#FaltanEnDestino;

IF OBJECT_ID('tempdb..#SobraEnDestino') IS NOT NULL DROP TABLE
#SobraEnDestino;

-- ----- 0) Validar existencia de la tabla en ambas BD -----

DECLARE @exS INT, @exD INT, @q NVARCHAR(MAX);

SET @q = N'

SELECT @ex = COUNT(*)

FROM ' + QUOTENAME(@Origen) + N'.sys.tables t

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```
JOIN ' + QUOTENAME(@Origen) + N'.sys.schemas s ON  
s.schema_id=t.schema_id
```

```
WHERE t.name=@T AND s.name=@S;;
```

```
EXEC sp_executesql @q, N'@T sysname,@S sysname,@ex int OUTPUT',  
@Tabla, @Esquema, @ex=@exS OUTPUT;
```

```
SET @q = N'
```

```
SELECT @ex = COUNT(*)
```

```
FROM ' + QUOTENAME(@Destino) + N'.sys.tables t
```

```
JOIN ' + QUOTENAME(@Destino) + N'.sys.schemas s ON  
s.schema_id=t.schema_id
```

```
WHERE t.name=@T AND s.name=@S;;
```

```
EXEC sp_executesql @q, N'@T sysname,@S sysname,@ex int OUTPUT',  
@Tabla, @Esquema, @ex=@exD OUTPUT;
```

```
IF ISNULL(@exS,0)=0 OR ISNULL(@exD,0)=0
```

```
BEGIN
```

```
    SELECT 'ERROR' AS Tipo, 'dbo.Paciente no existe en una de las BD' AS  
Detalle,
```

```
        @Origen AS Origen, @Destino AS Destino;
```

```
RETURN;
```

```
END;
```

```
-- ----- 1) Estructura de columnas (origen/destino) -----
```

```
CREATE TABLE #S(  
    column_id INT, name SYSNAME, type_name SYSNAME, max_length INT,  
    precision TINYINT, scale TINYINT,  
    is_nullable BIT, is_identity BIT
```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

);

CREATE TABLE #D(

column_id INT, name SYSNAME, type_name SYSNAME, max_length INT,
precision TINYINT, scale TINYINT,

is_nullable BIT, is_identity BIT

);

SET @q = N'

INSERT INTO

#S(column_id,name,type_name,max_length,precision,scale,is_nullable,is_identity)

SELECT c.column_id, c.name, ty.name, c.max_length, c.precision, c.scale,
c.is_nullable,

IIF(ic.column_id IS NULL,0,1)

FROM ' + QUOTENAME(@Origen) + N'.sys.tables t

JOIN ' + QUOTENAME(@Origen) + N'.sys.schemas s ON
s.schema_id=t.schema_id

JOIN ' + QUOTENAME(@Origen) + N'.sys.columns c ON
c.object_id=t.object_id

JOIN ' + QUOTENAME(@Origen) + N'.sys.types ty ON
ty.user_type_id=c.user_type_id

LEFT JOIN ' + QUOTENAME(@Origen) + N'.sys.identity_columns ic ON
ic.object_id=t.object_id AND ic.column_id=c.column_id

WHERE t.name=@T AND s.name=@S;';

EXEC sp_executesql @q, N'@T sysname,@S sysname', @Tabla, @Esquema;

SET @q = N'

INSERT INTO

#D(column_id,name,type_name,max_length,precision,scale,is_nullable,is_identity)

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```

SELECT c.column_id, c.name, ty.name, c.max_length, c.precision, c.scale,
c.is_nullable,

      IIF(ic.column_id IS NULL,0,1)

FROM ' + QUOTENAME(@Destino) + N'.sys.tables t

JOIN ' + QUOTENAME(@Destino) + N'.sys.schemas s ON
s.schema_id=t.schema_id

JOIN ' + QUOTENAME(@Destino) + N'.sys.columns c ON
c.object_id=t.object_id

JOIN ' + QUOTENAME(@Destino) + N'.sys.types ty ON
ty.user_type_id=c.user_type_id

LEFT JOIN ' + QUOTENAME(@Destino) + N'.sys.identity_columns ic ON
ic.object_id=t.object_id AND ic.column_id=c.column_id

WHERE t.name=@T AND s.name=@S;

EXEC sp_executesql @q, N'@T sysname,@S sysname', @Tabla, @Esquema;

```

-- ----- 2) Comparación de estructura -----

```

SELECT

COALESCE(S.column_id,D.column_id) AS column_id,

COALESCE(S.name,D.name)      AS Columna,

S.type_name AS Tipo_Origen, D.type_name AS Tipo_Destino,

S.max_length AS Len_Origen, D.max_length AS Len_Destino,

S.precision AS Prec_Origen, D.precision AS Prec_Destino,

S.scale AS Scale_Origen, D.scale AS Scale_Destino,

S.is_nullable AS Null_Origen, D.is_nullable AS Null_Destino,

S.is_identity AS Id_Origen, D.is_identity AS Id_Destino,

CASE

WHEN S.name IS NULL OR D.name IS NULL THEN 1

```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```

WHEN S.type_name<>D.type_name OR S.max_length<>D.max_length
OR S.precision<>D.precision OR S.scale<>D.scale
OR S.is_nullable<>D.is_nullable OR S.is_identity<>D.is_identity THEN 1
ELSE 0

END AS Diferente

INTO #EstructuraDif

FROM #S S

FULL JOIN #D D ON S.column_id=D.column_id AND S.name=D.name;

-- ----- 3) Conteos y diferencias de datos por CI (si existe) -----

DECLARE @countO BIGINT, @countD BIGINT, @hasCI BIT;

SELECT @countO = COUNT(*) FROM HospitalOrigen.dbo.Paciente;

SELECT @countD = COUNT(*) FROM HospitalDestino.dbo.Paciente;

SET @hasCI = CASE WHEN EXISTS(SELECT 1 FROM #S WHERE name=N'CI')
THEN 1 ELSE 0 END;

CREATE TABLE #FaltanEnDestino (CI NVARCHAR(200) NULL, Nombres
NVARCHAR(200) NULL, Apellidos NVARCHAR(200) NULL);

CREATE TABLE #SobraEnDestino (CI NVARCHAR(200) NULL, Nombres
NVARCHAR(200) NULL, Apellidos NVARCHAR(200) NULL);

IF @hasCI = 1

BEGIN

INSERT INTO #FaltanEnDestino(CI,Nombres,Apellidos)

SELECT o.CI, o.Nombres, o.Apellidos

FROM HospitalOrigen.dbo.Paciente o

```

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```
LEFT JOIN HospitalDestino.dbo.Paciente d ON d.CI = o.CI  
WHERE d.CI IS NULL;
```

```
INSERT INTO #SobraEnDestino(CI,Nombres,Apellidos)  
SELECT d.CI, d.Nombres, d.Apellidos  
FROM HospitalDestino.dbo.Paciente d  
LEFT JOIN HospitalOrigen.dbo.Paciente o ON o.CI = d.CI  
WHERE o.CI IS NULL;  
END;
```

-- ----- 4) RESUMEN (para tu presentación) -----

```
SELECT  
  
'Estructura columnas (tipo/len/prec/scale/null/identity)' AS Chequeo,  
  
CASE WHEN NOT EXISTS(SELECT 1 FROM #EstructuraDif WHERE  
Diferente=1) THEN 'OK' ELSE 'FAIL' END AS Resultado,  
  
(SELECT COUNT(*) FROM #EstructuraDif WHERE Diferente=1) AS  
Diferencias;
```

```
SELECT  
  
'Filas copiadas' AS Chequeo,  
  
CASE WHEN @countO=@countD THEN 'OK' ELSE 'FAIL' END AS Resultado,  
  
@countO AS Origen_Filas, @countD AS Destino_Filas,  
  
(SELECT COUNT(*) FROM #FaltanEnDestino) AS FaltanEnDestino,  
  
(SELECT COUNT(*) FROM #SobraEnDestino) AS SobraEnDestino;
```

-- ----- 5) DETALLES (solo si hay diferencias) -----

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

SELECT * FROM #EstructuraDif WHERE Diferente=1 ORDER BY column_id;
-- diferencias de estructura

SELECT TOP(50) * FROM #FaltanEnDestino; -- en Origen y no
en Destino (por CI)

SELECT TOP(50) * FROM #SobraEnDestino; -- en Destino y no
en Origen (por CI)

-- ----- 6) MUESTRAS VISIBLES (siempre) -----

-- TOP N de cada base (para que "se vea")

SELECT TOP (@TopN)

IdPaciente, CI, Nombres, Apellidos, FechaNacimiento, Sexo, Direccion,
Telefono, FechaRegistro

FROM HospitalOrigen.dbo.Paciente

ORDER BY IdPaciente;

SELECT TOP (@TopN)

IdPaciente, CI, Nombres, Apellidos, FechaNacimiento, Sexo, Direccion,
Telefono, FechaRegistro

FROM HospitalDestino.dbo.Paciente

ORDER BY IdPaciente;

-- Pareo lado a lado por CI (si existe)

IF @hasCI = 1

BEGIN

SELECT TOP (@TopN)

o.IdPaciente AS Id_Origen,

d.IdPaciente AS Id_Destino,

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
--	-------------------	-------

```
o.CI, o.Nombres, o.Apellidos, o.FechaNacimiento, o.Sexo

FROM HospitalOrigen.dbo.Paciente o

JOIN HospitalDestino.dbo.Paciente d ON d.CI = o.CI

ORDER BY o.IdPaciente;

END

ELSE

BEGIN

SELECT 'No se muestra pareo lado a lado porque la columna CI no existe.'
AS Info;

END;
```

10 %

No se encontraron problemas.

Línea: 166

Carácter: 1

Resultados

Mensajes

	Chequeo	Resultado	Diferencias
1	Estructura columnas (tipo/len/prec/scale/null/ide...	OK	0

	Chequeo	Resultado	Origen_Filas	Destino_Filas	FaltanEnDestino	SobraEnDestino
1	Filas copiadas	OK	100	100	0	0

	column_id	Columna	Tipo_Origen	Tipo_Destino	Len_Origen	Len_Destino	Prec_Origen	Prec_Destino	Scale_Origen	Scale_Destino	Null_Origen	Null_Destino	Id_Origen	Id_Destino	Diferente
	CI	Nombres	Apellidos												
	CI	Nombres	Apellidos												

	IdPaciente	CI	Nombres	Apellidos	FechaNacimiento	Sexo	Direccion	Telefono	FechaRegistro
1	1	12345678	Ana	Gutiérrez	1999-03-12	F	Av. Principal 123	70111111	2025-08-26 19:48:40
2	2	87654321	Luis	Pérez	1995-11-05	M	Calle 10 #45	70222222	2025-08-26 19:48:40
3	3	44556677	Marco	Ramírez	2001-01-20	M	Zona Centro	70333333	2025-08-26 19:48:40
4	4	CI00000...	Carlos	Torres	1964-05-26	F	Calle 115 # 992	708933...	2025-08-26 20:47:27

De los procesos realizar su copia de código y captura de imágenes

ESTUDIANTE: LUIS CARLOS FLORES DIAZ	FIRMA ESTUDIANTE:	FOLIO
-------------------------------------	-------------------	-------