

Creación de la base datos e inserción de datos de prueba

The screenshot displays the SQL Server Enterprise Manager interface. The left sidebar shows the server tree with 'HT1' expanded, showing 'Databases', 'System Databases', 'BD2', 'Security', and 'Server Objects'. The central query editor shows the following SQL code:

```
17 -- Creación de la tabla Estudiante
18 CREATE TABLE Estudiante (
19     IdEstudiante INT IDENTITY(1,1) PRIMARY KEY,
20     Nombre VARCHAR(100) NOT NULL,
21     Carnet VARCHAR(50) NOT NULL,
22     Creditos INT NOT NULL
23 );
24
25 -- Creación de la tabla Curso
26 CREATE TABLE Curso (
27     IdCurso INT IDENTITY(1,1) PRIMARY KEY,
28     Nombre VARCHAR(100) NOT NULL,
29     CreditosMinimos INT NOT NULL
30 );
31
32 -- Creación de la tabla Asignacion
```

The bottom section of the interface shows the 'QUERY RESULTS (PREVIEW)' tab. It contains two tables of results:

	IdEstudia...	Nombre	Carnet	Creditos
1	1	Juan Pérez	C001	30
2	2	María López	C002	15
3	3	Carlos Ruiz	C003	10

	IdCurso	Nombre	CreditosM...
1	1	Base de Datos ...	20
2	2	Programación A...	25
3	3	Redes	15

The screenshot shows the 'Messages' tab in the SQL Server interface. It displays the execution of three queries, each completed successfully at 11:25:59 PM. The messages indicate that each query affected 3 rows. The total execution time was 00:00:00.095.

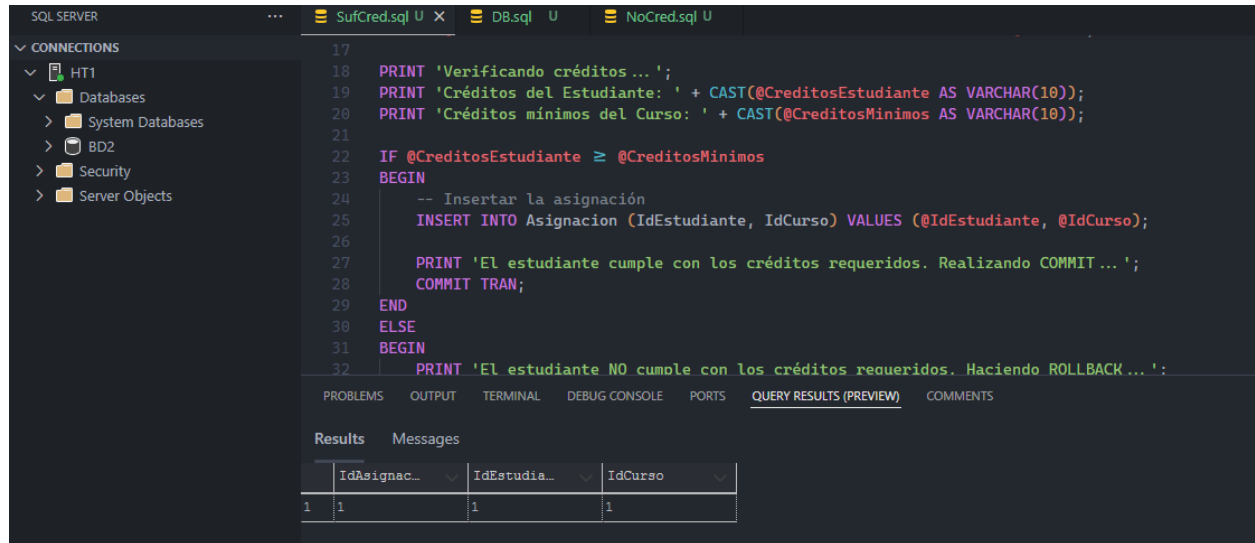
Results Messages

11:25:59 PM Started executing query at [Line 1](#)
Commands completed successfully.

11:25:59 PM Started executing query at [Line 5](#)
Commands completed successfully.

11:25:59 PM Started executing query at [Line 8](#)
(3 rows affected)
(3 rows affected)
(3 rows affected)
(3 rows affected)
Total execution time: 00:00:00.095

Suficientes créditos COMMIT

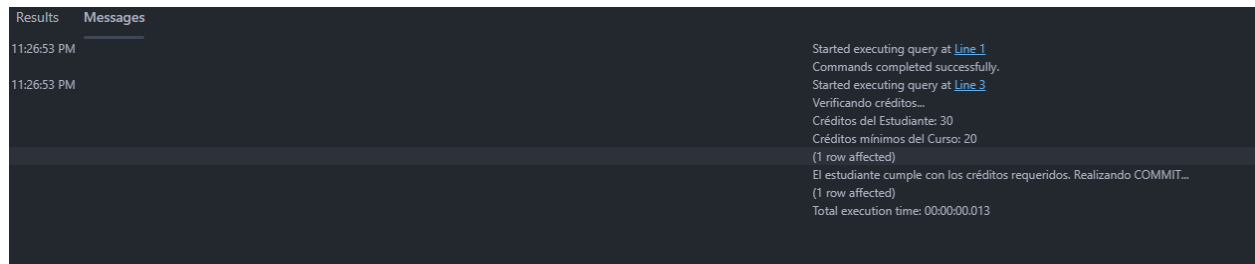


The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'CONNECTIONS' tree with 'HT1' selected. The main pane shows a T-SQL script with the following code:

```
17
18 PRINT 'Verificando créditos ...';
19 PRINT 'Créditos del Estudiante: ' + CAST(@CreditosEstudiante AS VARCHAR(10));
20 PRINT 'Créditos mínimos del Curso: ' + CAST(@CreditosMinimos AS VARCHAR(10));
21
22 IF @CreditosEstudiante >= @CreditosMinimos
23 BEGIN
24     -- Insertar la asignación
25     INSERT INTO Asignacion (IdEstudiante, IdCurso) VALUES (@IdEstudiante, @IdCurso);
26
27     PRINT 'El estudiante cumple con los créditos requeridos. Realizando COMMIT ...';
28     COMMIT TRAN;
29 END
30 ELSE
31 BEGIN
32     PRINT 'El estudiante NO cumple con los créditos requeridos. Haciendo ROLLBACK ...':
```

Below the script, the 'QUERY RESULTS (PREVIEW)' tab is active, showing a table with the following data:

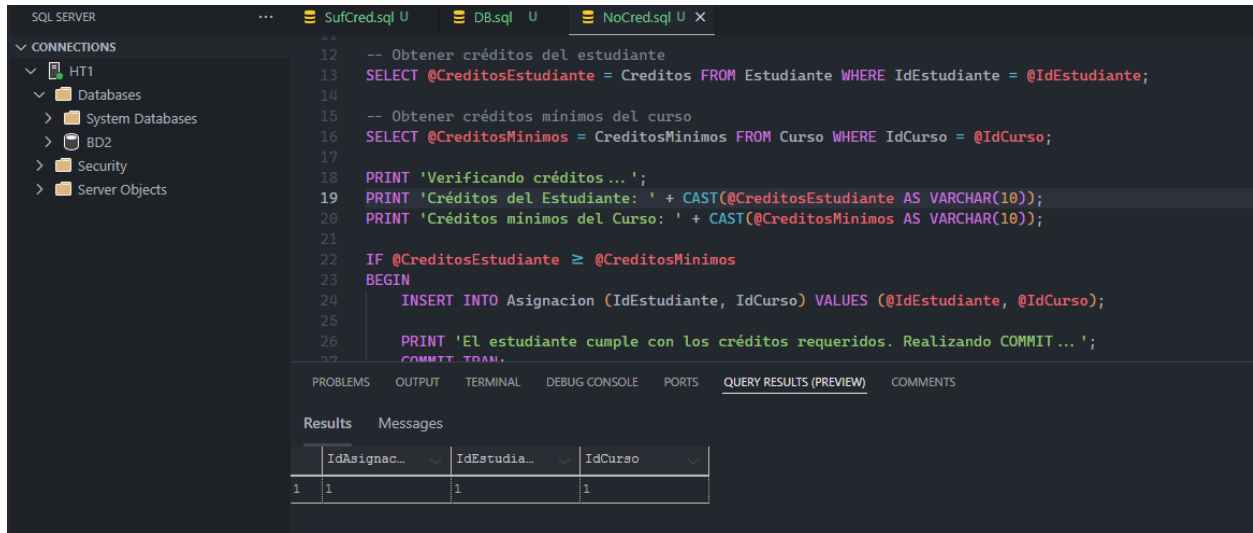
	IdAsignac...	IdEstudia...	IdCurso
1	1	1	1



The screenshot shows the 'Messages' window in SQL Server Enterprise Manager. It displays the execution of the T-SQL script, including the following messages:

```
11:26:53 PM Started executing query at Line 1
11:26:53 PM Commands completed successfully.
11:26:53 PM Started executing query at Line 3
11:26:53 PM Verificando créditos...
11:26:53 PM Créditos del Estudiante: 30
11:26:53 PM Créditos mínimos del Curso: 20
11:26:53 PM (1 row affected)
11:26:53 PM El estudiante cumple con los créditos requeridos. Realizando COMMIT...
11:26:53 PM (1 row affected)
11:26:53 PM Total execution time: 00:00:00.013
```

Insuficientes Créditos y Rollback



```
--
12 -- Obtener créditos del estudiante
13 SELECT @CreditosEstudiante = Creditos FROM Estudiante WHERE IdEstudiante = @IdEstudiante;
14
15 -- Obtener créditos mínimos del curso
16 SELECT @CreditosMinimos = CreditosMinimos FROM Curso WHERE IdCurso = @IdCurso;
17
18 PRINT 'Verificando créditos...';
19 PRINT 'Créditos del Estudiante: ' + CAST(@CreditosEstudiante AS VARCHAR(10));
20 PRINT 'Créditos mínimos del Curso: ' + CAST(@CreditosMinimos AS VARCHAR(10));
21
22 IF @CreditosEstudiante < @CreditosMinimos
23 BEGIN
24     INSERT INTO Asignacion (IdEstudiante, IdCurso) VALUES (@IdEstudiante, @IdCurso);
25
26     PRINT 'El estudiante cumple con los créditos requeridos. Realizando COMMIT...';
27     COMMIT TRANSACTION;
28 END
```

IdAsignac...	IdEstudia...	IdCurso
1	1	1



Results Messages

11:30:36 PM

11:30:36 PM

Started executing query at [Line 1](#)
Commands completed successfully.
Started executing query at [Line 3](#)
Verificando créditos...
Créditos del Estudiante: 10
Créditos mínimos del Curso: 20
El estudiante NO cumple con los créditos requeridos. Haciendo ROLLBACK...
(1 row affected)
Total execution time: 00:00:00.011

SCRIPT

Creación de la Base de datos:

```
-- Crear la base de datos
IF DB_ID('BD2') IS NULL
    CREATE DATABASE BD2;
GO

USE BD2;
GO

-- Eliminación de tablas previas
IF OBJECT_ID('dbo.Asignacion', 'U') IS NOT NULL
    DROP TABLE dbo.Asignacion;
IF OBJECT_ID('dbo.Curso', 'U') IS NOT NULL
    DROP TABLE dbo.Curso;
IF OBJECT_ID('dbo.Estudiante', 'U') IS NOT NULL
    DROP TABLE dbo.Estudiante;
```

```

-- Creación de la tabla Estudiante
CREATE TABLE Estudiante (
    IdEstudiante INT IDENTITY(1,1) PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL,
    Carnet VARCHAR(50) NOT NULL,
    Creditos INT NOT NULL
);

-- Creación de la tabla Curso
CREATE TABLE Curso (
    IdCurso INT IDENTITY(1,1) PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL,
    CreditosMinimos INT NOT NULL
);

-- Creación de la tabla Asignacion
CREATE TABLE Asignacion (
    IdAsignacion INT IDENTITY(1,1) PRIMARY KEY,
    IdEstudiante INT NOT NULL,
    IdCurso INT NOT NULL,
    FOREIGN KEY (IdEstudiante) REFERENCES Estudiante(IdEstudiante),
    FOREIGN KEY (IdCurso) REFERENCES Curso(IdCurso)
);

-- Insertar datos de prueba en Estudiante
INSERT INTO Estudiante (Nombre, Carnet, Creditos) VALUES
('Juan Pérez', 'C001', 30),
('María López', 'C002', 15),
('Carlos Ruiz', 'C003', 10);

-- Insertar datos de prueba en Curso
-- Supongamos que el curso requiere al menos 20 créditos para poder
inscribirse
INSERT INTO Curso (Nombre, CreditosMinimos) VALUES
('Base de Datos II', 20),
('Programación Avanzada', 25),
('Redes', 15);

SELECT * FROM Estudiante;
SELECT * FROM Curso;

```

Commit

```

USE BD2;
GO

-- Definir variables
DECLARE @IdEstudiante INT = 1; -- Estudiante Juan Pérez con 30 créditos
DECLARE @IdCurso INT = 1;      -- Curso "Base de Datos II" con mínimo 20
                                -- créditos
DECLARE @CreditosEstudiante INT;
DECLARE @CreditosMinimos INT;

BEGIN TRAN -- Inicia la transacción

-- Obtener los créditos del estudiante
SELECT @CreditosEstudiante = Creditos FROM Estudiante WHERE IdEstudiante =
@IdEstudiante;

-- Obtener los créditos mínimos del curso
SELECT @CreditosMinimos = CreditosMinimos FROM Curso WHERE IdCurso =
@IdCurso;

PRINT 'Verificando créditos...';
PRINT 'Créditos del Estudiante: ' + CAST(@CreditosEstudiante AS
VARCHAR(10));
PRINT 'Créditos mínimos del Curso: ' + CAST(@CreditosMinimos AS
VARCHAR(10));

IF @CreditosEstudiante >= @CreditosMinimos
BEGIN
    -- Insertar la asignación
    INSERT INTO Asignacion (IdEstudiante, IdCurso) VALUES (@IdEstudiante,
@IdCurso);

    PRINT 'El estudiante cumple con los créditos requeridos. Realizando
COMMIT...';
    COMMIT TRAN;
END
ELSE
BEGIN
    PRINT 'El estudiante NO cumple con los créditos requeridos. Haciendo
ROLLBACK...';
    ROLLBACK TRAN;
END

-- Verificar resultados
SELECT * FROM Asignacion;

```

Rollback

```
USE BD2;
GO

-- Definir variables
DECLARE @IdEstudiante INT = 3; -- Estudiante Carlos Ruiz con 10 créditos
DECLARE @IdCurso INT = 1;      -- Curso "Base de Datos II" con mínimo 20
                                -- créditos
DECLARE @CreditosEstudiante INT;
DECLARE @CreditosMinimos INT;

BEGIN TRAN -- Inicia la transacción

-- Obtener créditos del estudiante
SELECT @CreditosEstudiante = Creditos FROM Estudiante WHERE IdEstudiante =
@IdEstudiante;

-- Obtener créditos mínimos del curso
SELECT @CreditosMinimos = CreditosMinimos FROM Curso WHERE IdCurso =
@IdCurso;

PRINT 'Verificando créditos...';
PRINT 'Créditos del Estudiante: ' + CAST(@CreditosEstudiante AS
VARCHAR(10));
PRINT 'Créditos mínimos del Curso: ' + CAST(@CreditosMinimos AS
VARCHAR(10));

IF @CreditosEstudiante >= @CreditosMinimos
BEGIN
    INSERT INTO Asignacion (IdEstudiante, IdCurso) VALUES (@IdEstudiante,
@IdCurso);

    PRINT 'El estudiante cumple con los créditos requeridos. Realizando
COMMIT...';
    COMMIT TRAN;
END
ELSE
BEGIN
    PRINT 'El estudiante NO cumple con los créditos requeridos. Haciendo
ROLLBACK...';
    ROLLBACK TRAN;
```

```
END
```

```
-- Verificar resultados (no debe haber cambios en esta ocasión)
```

```
SELECT * FROM Asignacion;
```