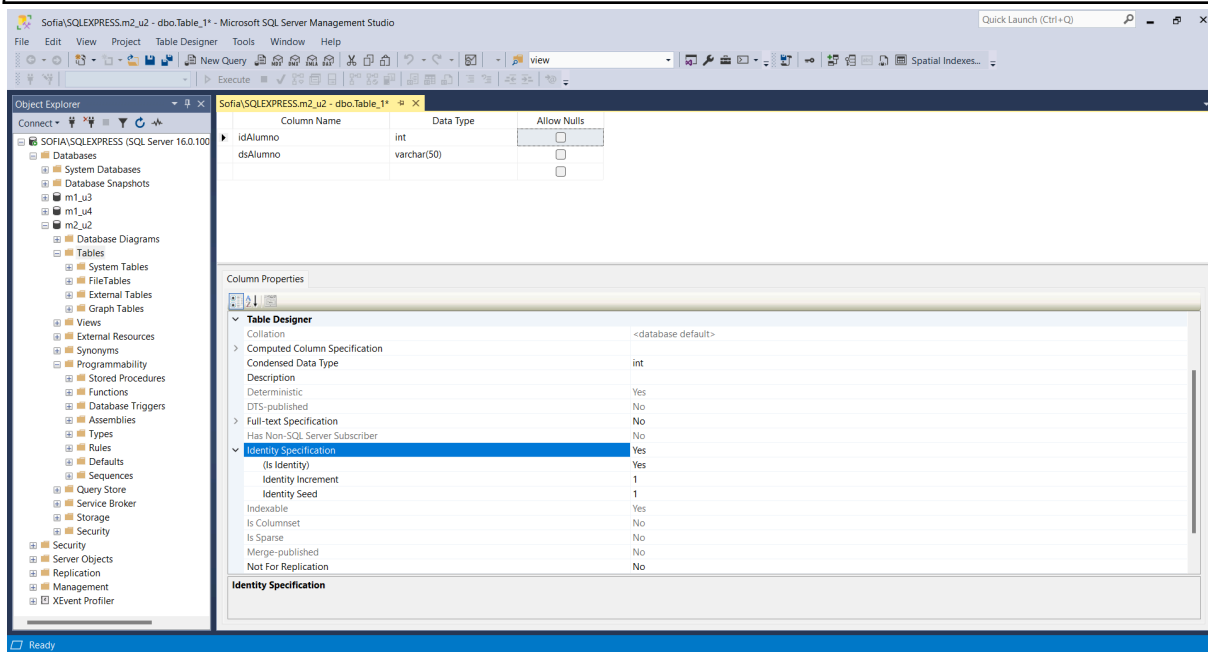


Ejercicios Prácticos

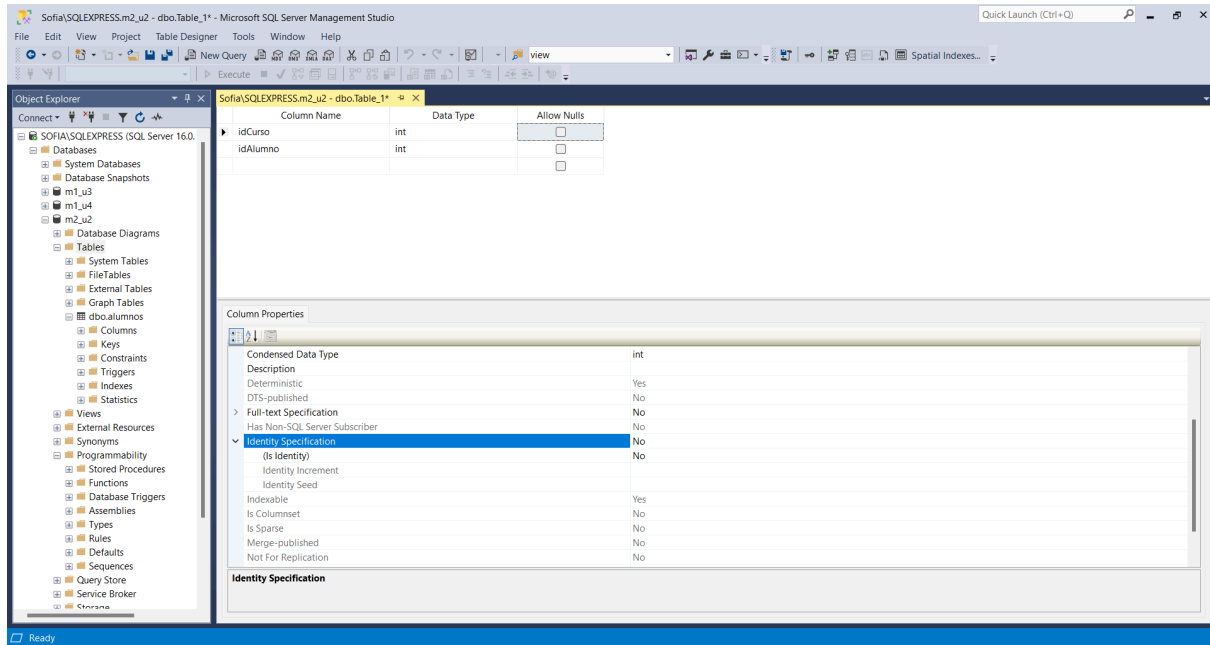
Ejercicio Práctico 6.1

Cree una tabla Alumnos con un campo IdAlumno que sea autonumérico y otro campo DsAlumno con su nombre y apellido



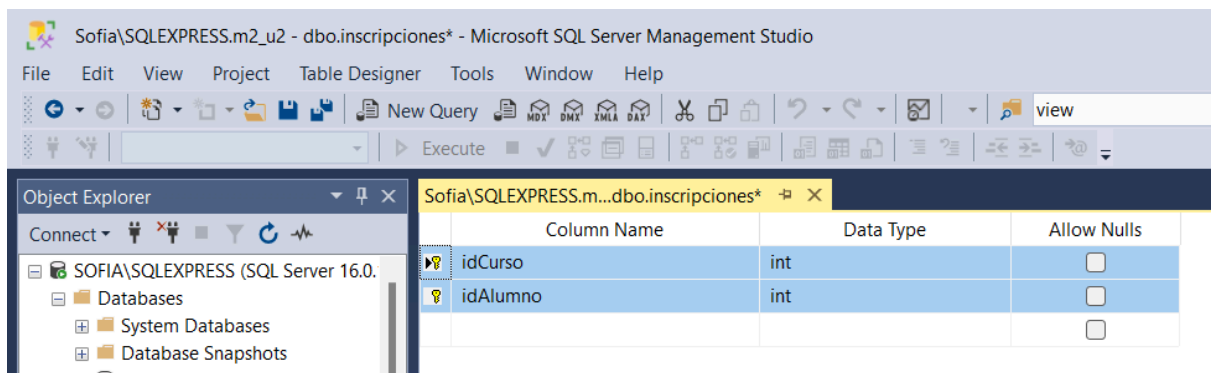
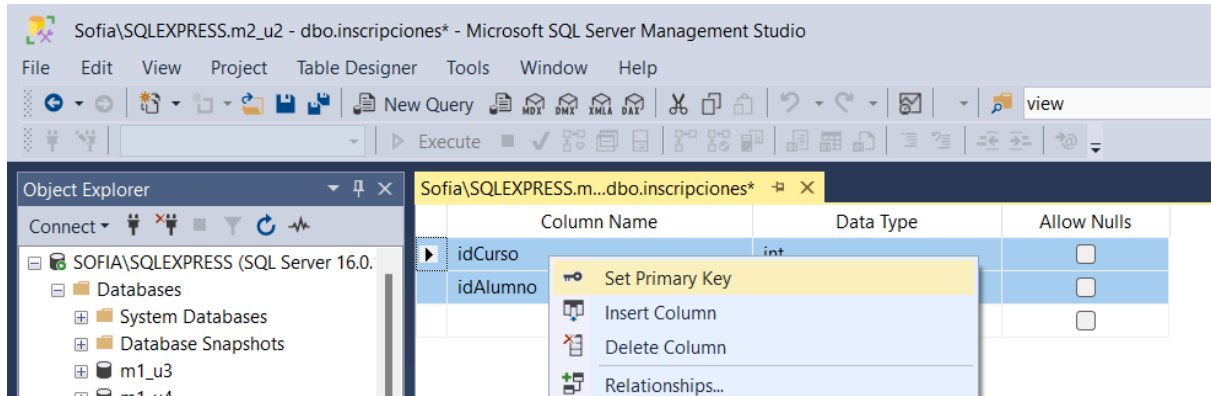
Ejercicio Práctico 6.2

Cree una tabla que se llame Inscripciones en la cual tengamos el IdCurso y el IdAlumno



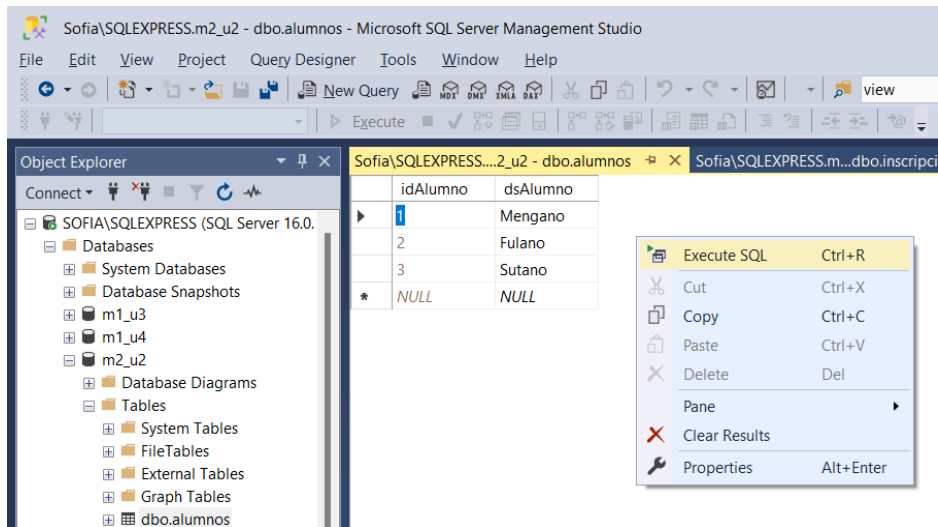
Ejercicio Práctico 6.3

Cree una clave primaria compuesta para la tabla Inscripciones que incluya el campo IdAlumno y el campo IdCurso



Ejercicio Práctico 6.4

Cargar tres alumnos que se llamen Mengano, Fulano y Sutano



Ejercicio Práctico 6.5

Escriba una consulta para obtener todos los alumnos y pruébela.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'SQLQuery1.sql' connected to 'SOFIA\SQLEXPRESS.m2_u2 (SOFIA\pablo (52))*'. The menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains icons for opening files, saving, and executing queries. The 'Object Explorer' on the left shows the database structure for 'SOFIA\SQLEXPRESS (SQL Server 16.0)'. It lists 'Databases' (System, Snapshots, m1_u3, m1_u4, m2_u2), 'Database Diagrams', 'Tables' (System, File, External, Graph, dbo.alumnos, dbo.cursos, dbo.inscripciones), 'Views', 'External Resources', 'Synonyms', 'Programmability' (Stored Procedures, Functions, Triggers, Assemblies), and 'Assemblies'. The 'dbo.alumnos' table is selected. The main query editor shows the following SQL code:

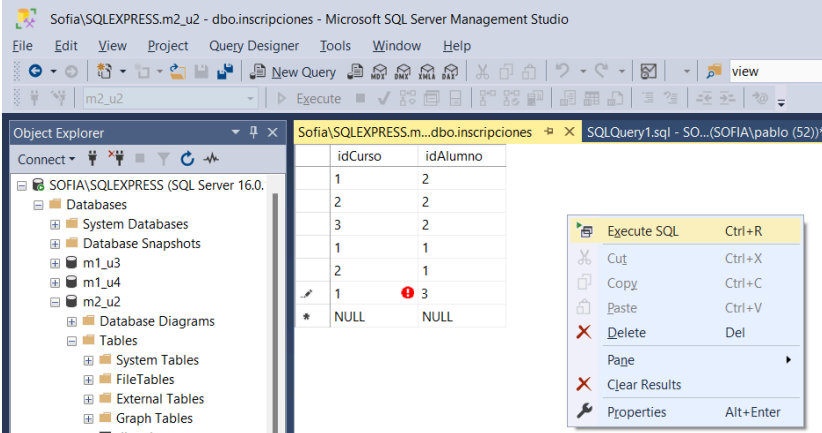
```
-- 6.5
SELECT * FROM alumnos;
```

The 'Results' pane at the bottom displays the output of the query as a table with two columns: 'idAlumno' and 'dsAlumno'. The table contains three rows of data:

	idAlumno	dsAlumno
1	1	Mengano
2	2	Fulano
3	3	Sutano

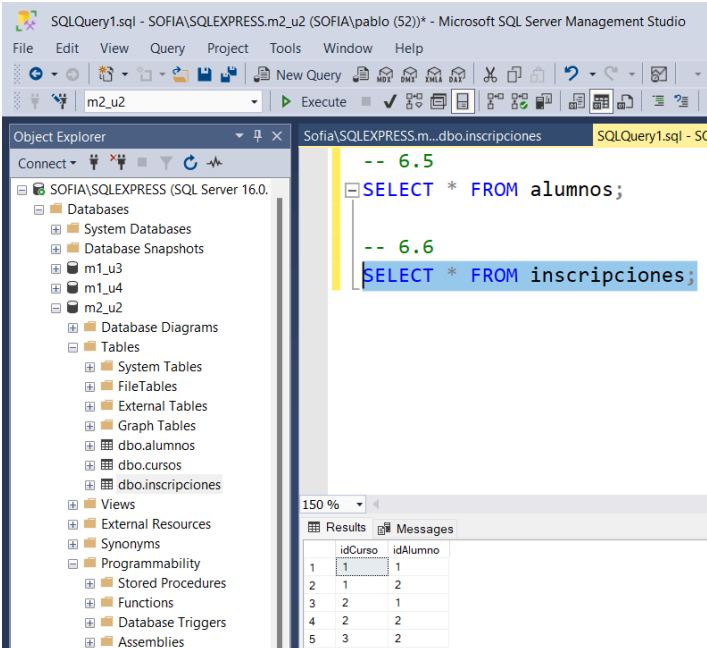
Ejercicio Práctico 6.6

Cargue seis registros en la tabla Inscripciones para que Fulano este tomando los tres cursos, Mengano sólo los dos primeros y Sutano sólo el primero de todos.



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the database structure for 'SOFIA\SQLEXPRESS (SQL Server 16.0.)'. The 'inscripciones' table is selected, and its data is displayed in the 'SQLQuery1.sql' window. The table has two columns: 'idCurso' and 'idAlumno'. There are 6 records in total. A context menu is open over the table, showing options like 'Execute SQL', 'Cut', 'Copy', 'Paste', 'Delete', 'Page', 'Clear Results', and 'Properties'.

idCurso	idAlumno
1	2
2	2
3	2
1	1
2	1
1	3
NULL	NULL

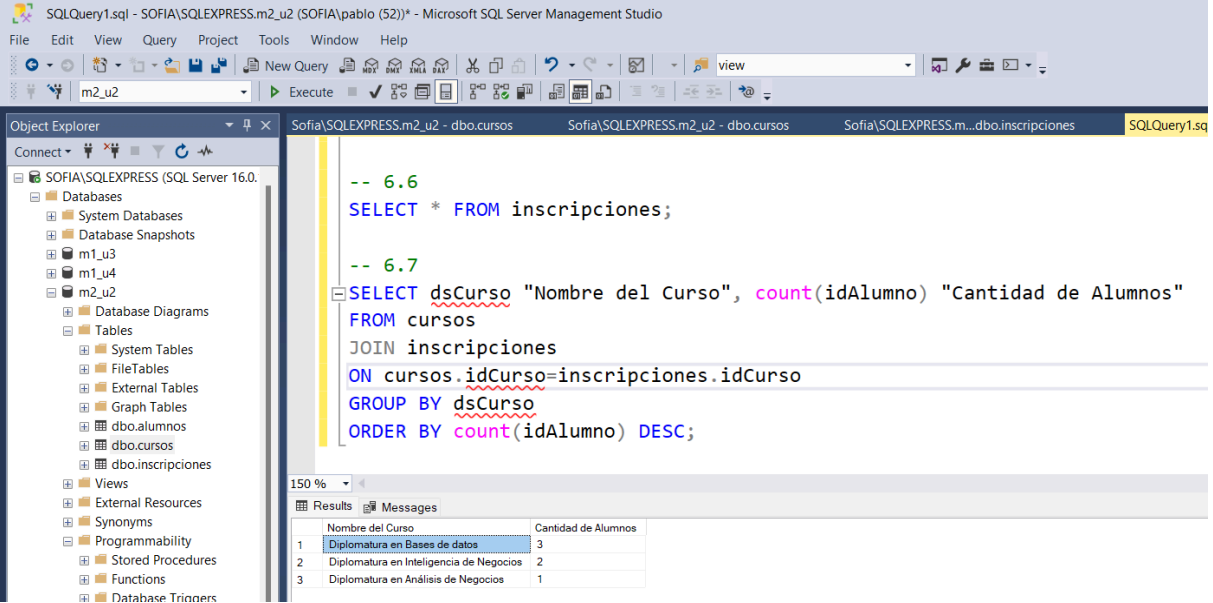


The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the database structure for 'SOFIA\SQLEXPRESS (SQL Server 16.0.)'. The 'inscripciones' table is selected, and its data is displayed in the 'SQLQuery1.sql' window. The table has two columns: 'idCurso' and 'idAlumno'. There are 6 records in total. The 'Results' window is open, showing the first 5 records.

idCurso	idAlumno
1	1
2	2
3	1
2	2
3	2

Ejercicio Práctico 6.7

Cree y ejecute una consulta que indique cuantos alumnos hay en cada curso (va a tener que combinar GROUP BY y JOIN)



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'SOFIA\SQLEXPRESS (SQL Server 16.0.)', including databases, tables, and views. The main query window shows the following SQL code:

```
-- 6.6
SELECT * FROM inscripciones;

-- 6.7
SELECT dsCurso "Nombre del Curso", count(idAlumno) "Cantidad de Alumnos"
FROM cursos
JOIN inscripciones
ON cursos.idCurso=inscripciones.idCurso
GROUP BY dsCurso
ORDER BY count(idAlumno) DESC;
```

The Results pane at the bottom displays the output of the query, showing a table with two columns: 'Nombre del Curso' and 'Cantidad de Alumnos'. The results are as follows:

	Nombre del Curso	Cantidad de Alumnos
1	Diplomatura en Bases de datos	3
2	Diplomatura en Inteligencia de Negocios	2
3	Diplomatura en Análisis de Negocios	1

Ejercicio Práctico 6.8

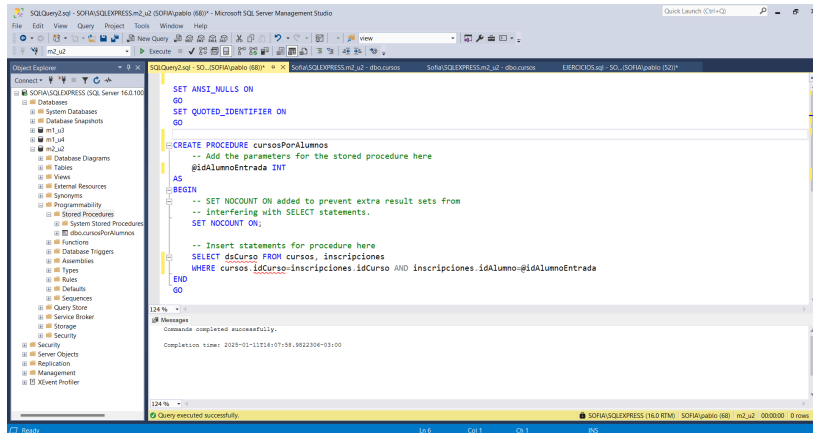
Cargue valores para la tabla de inscripciones de manera que el primer alumno tenga asignados 3 cursos, el segundo 2 y el tercero 1. Si no ha cargado aún los alumnos hágalo.

Realizado en el ejercicio 6.6

Ejercicio Práctico 6.9

Vuelva a ejecutar el procedimiento almacenado para verificar que lista los cursos adecuadamente.

Primero se crea el procedimiento almacenado

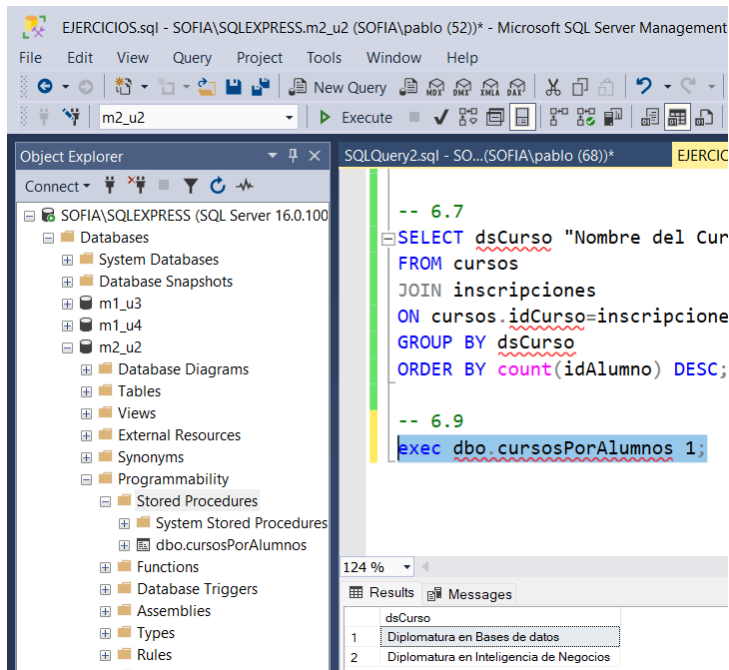


```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE cursosPorAlumnos
    @idAlumnoEntrada INT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT dsCurso FROM cursos, inscripciones
    WHERE cursos.idCurso=inscripciones.idCurso AND inscripciones.idAlumno=@idAlumnoEntrada
END
GO
```

124 %
Messages
Create completed successfully.
Completion time: 2023-01-11 15:41:07.58, 3922306-03:00

124 %
Query executed successfully.



```
-- 6.7
SELECT dsCurso "Nombre del Cur
FROM cursos
JOIN inscripciones
ON cursos.idCurso=inscripciones
GROUP BY dsCurso
ORDER BY count(idAlumno) DESC;

-- 6.9
exec dbo.cursosPorAlumnos 1;
```

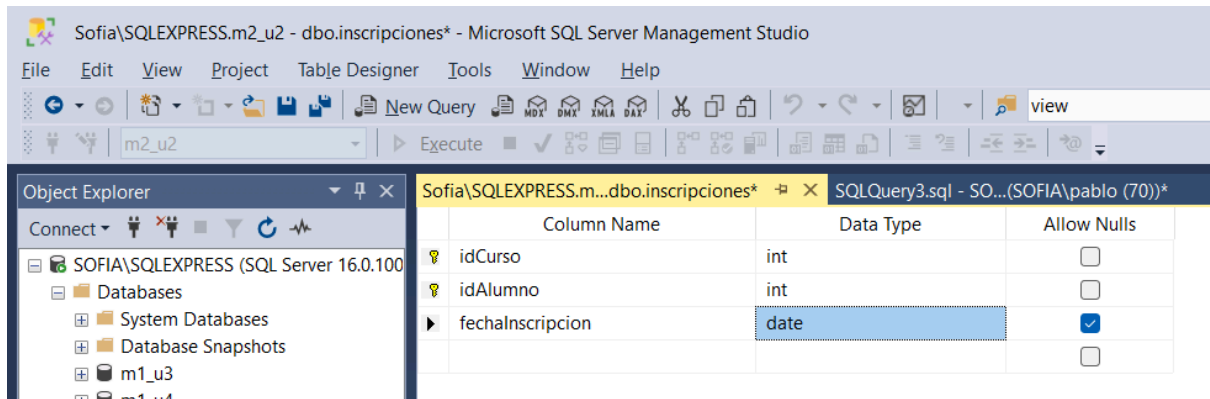
124 %
Results
Messages

	dsCurso
1	Diplomatura en Bases de datos
2	Diplomatura en Inteligencia de Negocios

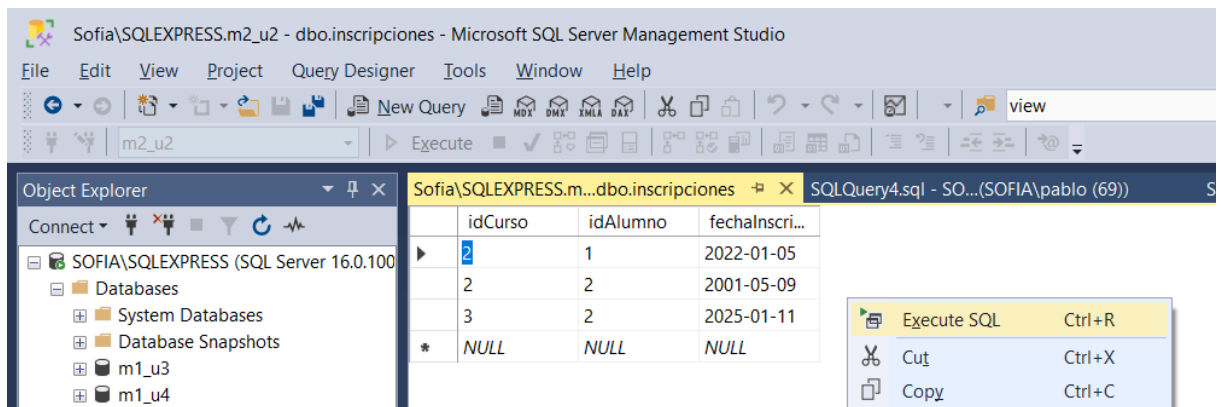
Ejercicio Práctico 6.10

Cargue las fechas de inscripción y vuelva a ejecutar la función para verificar que devuelva el resultado esperado.

Se agrega la columna fechaInscripcion a la tabla inscripcion



Se agregan registros en la nueva columna



Se genera la función

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [dbo].[ultimaInscripcionCurso]
(
    -- Add the parameters for the function here
    @entrada int
)
RETURNS date
AS
BEGIN
    -- Declare the return variable here
    DECLARE @fecha date

    -- Add the T-SQL statements to compute the return value here
    SET @fecha=(SELECT MAX(fechaInscripcion) FROM inscripciones WHERE idCurso=@entrada)

    -- Return the result of the function
    RETURN @fecha
END
GO
```

113 %
Messages
Commands completed successfully.
Completion time: 2022-01-11T16:24:24.7351638-03:00

Se ejecuta la función

```
FROM cursos
JOIN inscripciones
ON cursos.idCurso=inscripciones.idCurso
GROUP BY dsCurso
ORDER BY count(idAlumno) DESC;

-- 6.9
exec dbo.cursosPorAlumnos 1;

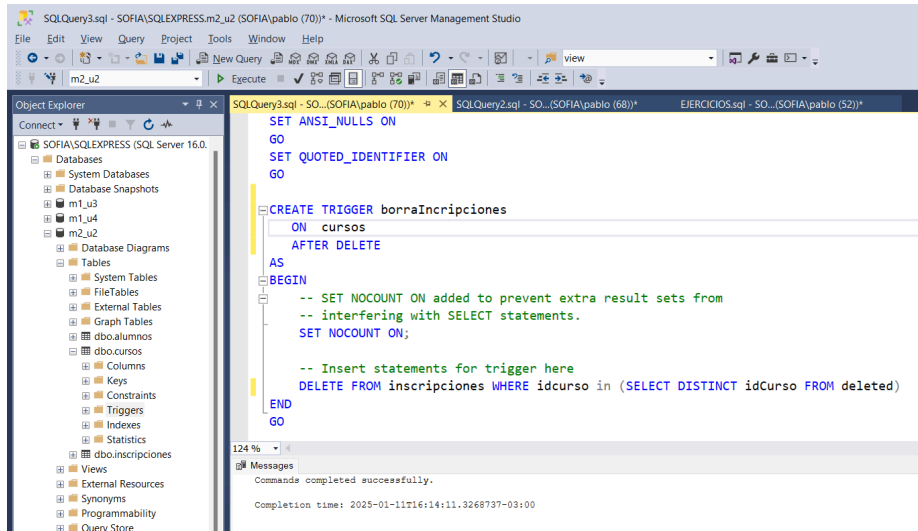
-- 6.10
SELECT dbo.ultimaInscripcionCurso (2);

-- EXTRA
SELECT * FROM cursos;
SELECT * FROM inscripciones;
DELETE FROM cursos WHERE idCurso=1;
```

113 %
Results
(No column name)
1 2022-01-05

EXTRA

Se genera el TRIGGER

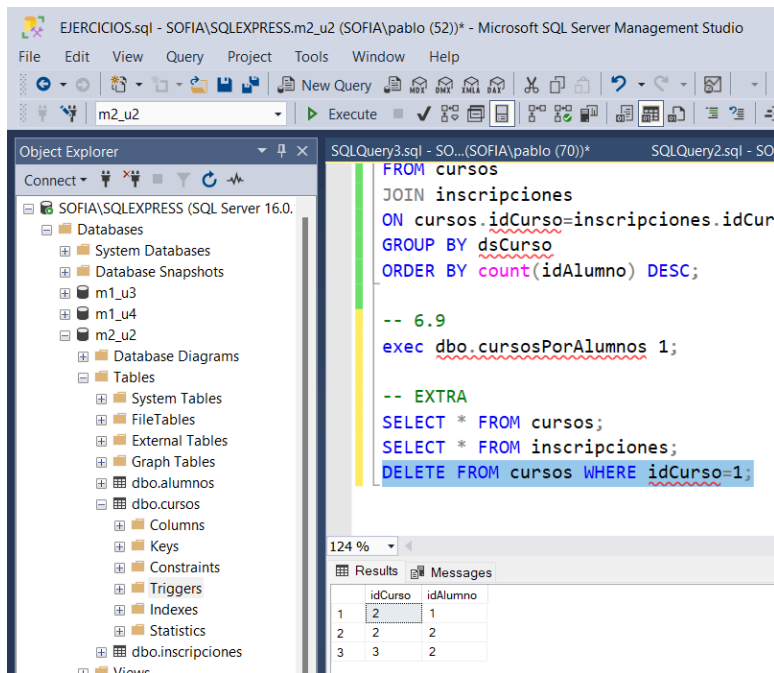


```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER borraInscripciones
ON cursos
AFTER DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    DELETE FROM inscripciones WHERE idCurso IN (SELECT DISTINCT idCurso FROM deleted)
END
GO
```

124 %
Messages
Commands completed successfully.
Completion time: 2025-01-11T16:14:11.3268737-03:00



```
FROM cursos
JOIN inscripciones
ON cursos.idCurso=inscripciones.idCurso
GROUP BY idCurso
ORDER BY count(idAlumno) DESC;

-- 6.9
exec dbo.cursosPorAlumnos 1;

-- EXTRA
SELECT * FROM cursos;
SELECT * FROM inscripciones;
DELETE FROM cursos WHERE idCurso=1;
```

124 %
Results
Messages

	idCurso	idAlumno
1	2	1
2	2	2
3	3	2

Luego de ejecutar el borrado de un curso, se consulta en la tabla inscripciones y también se eliminó lo relacionado al curso idCurso=1.