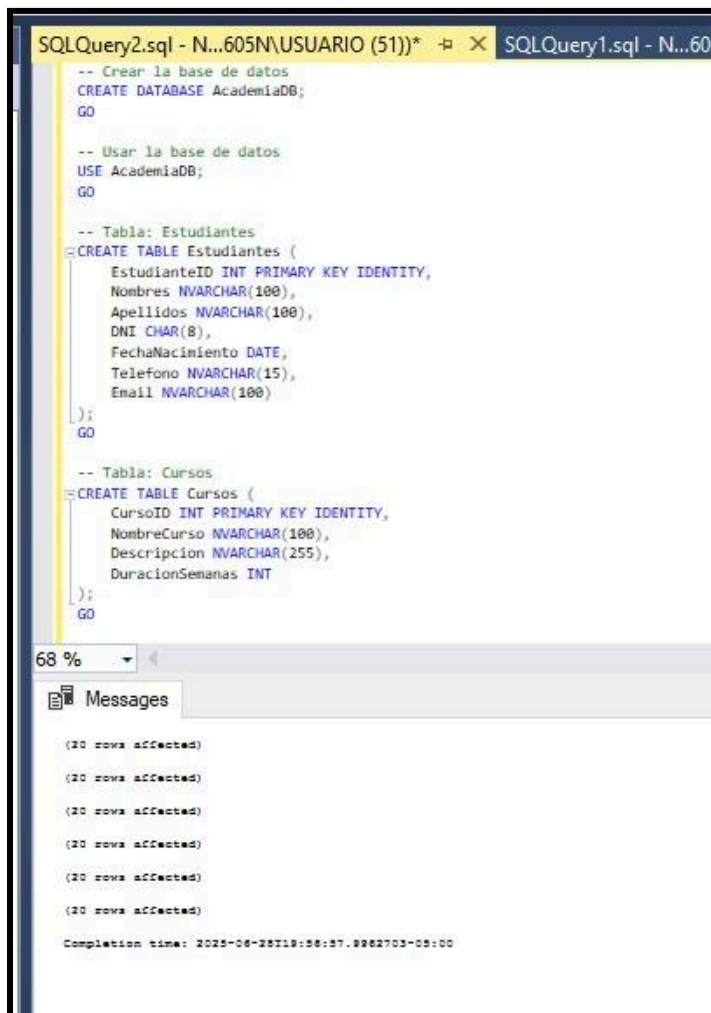


CONSULTAS AVANZADAS EN SQL SERVER - RICSE POMALAZA SOFÍA

La creación de la base de datos



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery2.sql - N...605N\USUARIO (51))' and 'SQLQuery1.sql - N...60...'. The active tab contains the following SQL script:

```
-- Crear la base de datos
CREATE DATABASE AcademiaDB;
GO

-- Usar la base de datos
USE AcademiaDB;
GO

-- Tabla: Estudiantes
CREATE TABLE Estudiantes (
    EstudianteID INT PRIMARY KEY IDENTITY,
    Nombres NVARCHAR(100),
    Apellidos NVARCHAR(100),
    DNI CHAR(8),
    FechaNacimiento DATE,
    Telefono NVARCHAR(15),
    Email NVARCHAR(100)
);
GO

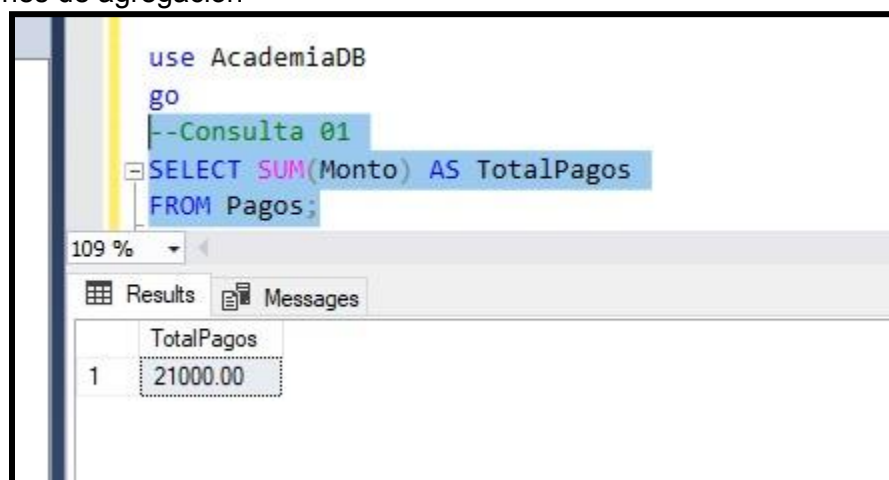
-- Tabla: Cursos
CREATE TABLE Cursos (
    CursoID INT PRIMARY KEY IDENTITY,
    NombreCurso NVARCHAR(100),
    Descripcion NVARCHAR(255),
    DuracionSemanas INT
);
GO
```

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

```
(20 rows affected)
(20 rows affected)
(20 rows affected)
(20 rows affected)
(20 rows affected)
(20 rows affected)
Completion time: 2023-06-28T19:58:37.9962703-05:00
```

- Funciones de agregación

1.



The screenshot shows a SQL Server Enterprise Manager window with a single tab: 'SQLQuery1.sql - N...60...'. The active tab contains the following SQL script:

```
use AcademiaDB
go
--Consulta 01
SELECT SUM(Monto) AS TotalPagos
FROM Pagos;
```

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

	TotalPagos
1	21000.00

2.

```
--Consulta 02
SELECT AVG(Monto) AS PromedioMontoPorMatricula
FROM Pagos;
```

109 %

Results Messages

	PromedioMontoPorMatricula
1	1050.000000

3.

```
--Consulta 03
SELECT MIN(Monto) AS MontoMinimo, MAX(Monto) AS MontoMaximo
FROM Pagos;
```

109 %

Results Messages

	PromedioMontoPorMatricula
1	1050.000000

4.

SQLQuery2.sql - N...605N\USUARIO (51))

```
--Consulta 04
SELECT COUNT(MatriculaID) AS TotalMatriculas
FROM Matriculas;
```

--Consulta 05

109 %

Results Messages

	TotalMatriculas
1	20

SQLQuery1.sql - N...605N\USUARIO (62))

5.

```
--Consulta 05
SELECT COUNT(CursoID) AS CursosMayorDuracionPromedio
FROM Cursos
WHERE DuracionSemanas > (SELECT AVG(DuracionSemanas) FROM Cursos);
```

2 %

Results Messages

	CursosMayorDuracionPromedio
1	6

6.

```
--Consulta 06
SELECT E.Nombres, E.Apellidos, SUM(P.Monto) AS TotalPagado
FROM Estudiantes AS E
JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY E.EstudianteID, E.Nombres, E.Apellidos;
```

109 %

Results Messages

	Nombres	Apellidos	TotalPagado
1	Ana	Perez	100.00
2	Luis	Gomez	200.00
3	Carlos	Lopez	300.00
4	Maria	Rodriguez	400.00
5	Jose	Martinez	500.00
6	Laura	Garcia	600.00
7	David	Hernandez	700.00
8	Sofia	Lopez	800.00
9	Daniel	Gonzalez	900.00
10	Andrea	Ramirez	1000.00
11	Miguel	Fernandez	1100.00
12	Lucia	Torres	1200.00

7.

```
--Consulta 07
SELECT C.NombreCurso, COUNT(M.EstudianteID) AS NumeroEstudiantes
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
GROUP BY C.CursoID, C.NombreCurso;
```

109 %

Results Messages

	NombreCurso	NumeroEstudiantes
1	Matemáticas Básicas	1
2	Programación en Python	1
3	Historia Universal	1
4	Química General	1
5	Física Aplicada	1
6	Literatura Española	1
7	Biología Molecular	1
8	Arte Moderno	1
9	Economía Básica	1
10	Geografía Mundial	1
11	Estadística Aplicada	1

8.

```
--Consulta 08
SELECT C.NombreCurso, AVG(P.Monto) AS PromedioPagos
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY C.CursoID, C.NombreCurso;
```

109 %

Results Messages

	NombreCurso	PromedioPagos
1	Matemáticas Básicas	100.000000
2	Programación en Python	200.000000
3	Historia Universal	300.000000
4	Química General	400.000000
5	Física Aplicada	500.000000
6	Literatura Española	600.000000
7	Biología Molecular	700.000000
8	Arte Moderno	800.000000
9	Economía Básica	900.000000
10	Geografía Mundial	1000.000000

9.

```
--Consulta 09
SELECT C.NombreCurso, COUNT(M.EstudianteID) AS NumeroEstudiantes
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
GROUP BY C.CursoID, C.NombreCurso
HAVING COUNT(M.EstudianteID) > 1;
```

109 %

Results Messages

	NombreCurso	NumeroEstudiantes
--	-------------	-------------------

10.

```
--Consulta 10
SELECT MetodoPago, SUM(Monto) AS TotalPagado
FROM Pagos
GROUP BY MetodoPago;
```

109 %

Results Messages

	MetodoPago	TotalPagado
1	Tarjeta de crédito	21000.00

11.

```
--Consulta 11
SELECT DISTINCT E.Nombres, E.Apellidos
FROM Estudiantes AS E
JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
WHERE P.Monto > (SELECT AVG(Monto) FROM Pagos);
```

109 %

Results Messages

	Nombres	Apellidos
1	Carmen	Sanchez
2	Elena	Ruiz
3	Fernando	Morales
4	Isabel	Vega
5	Jorge	Castro
6	Juan	Diaz
7	Lucia	Torres
8	Miguel	Fernandez
9	Patricia	Ortiz
10	Pedro	Jimenez

12.

```
--Consulta 12
SELECT C.NombreCurso
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
WHERE M.EstudianteID = (
    SELECT TOP 1 EstudianteID
    FROM Matriculas
    GROUP BY EstudianteID
    ORDER BY COUNT(MatriculaID) DESC
);
```

109 %

Results Messages

	NombreCurso
1	Programación en Python

13.

```
--Consulta 13
create table CursoDocente (CursoID INT, DocenteID INT, PRIMARY KEY (CursoID, DocenteID),
FOREIGN KEY (CursoID) REFERENCES Cursos(CursoID), FOREIGN KEY (DocenteID) REFERENCES Docentes(DocenteID));

SELECT D.Nombres, D.Apellidos
FROM Docentes AS D
JOIN CursoDocente AS CD ON D.DocenteID = CD.DocenteID
JOIN Horarios AS H ON CD.CursoID = H.CursoID
GROUP BY D.DocenteID, D.Nombres, D.Apellidos
HAVING COUNT(H.HorarioID) > 2;
```

99 %

Results Messages

Nombres	Apellidos
---------	-----------

14.

```
--Consulta 14
SELECT E.Nombres, E.Apellidos
FROM Estudiantes AS E
LEFT JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
LEFT JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
WHERE P.PagoID IS NULL;

--Consulta 15
```

109 %

Results Messages

Nombres	Apellidos
---------	-----------

15.

```
--Consulta 15
SELECT C.NombreCurso, SUM(P.Monto) AS TotalPagosCurso
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY C.CursoID, C.NombreCurso
HAVING SUM(P.Monto) > (
    SELECT SUM(P2.Monto)
    FROM Cursos AS C2
    JOIN Matriculas AS M2 ON C2.CursoID = M2.CursoID
    JOIN Pagos AS P2 ON M2.MatriculaID = P2.MatriculaID
    WHERE C2.NombreCurso = 'Matemáticas Básicas'
);
```

90 %

Results Messages

	NombreCurso	TotalPagosCurso
1	Programación en Python	200.00
2	Historia Universal	300.00
3	Química General	400.00
4	Física Aplicada	500.00
5	Literatura Española	600.00
6	Biología Molecular	700.00
7	Arte Moderno	800.00
8	Economía Básica	900.00
9	Geografía Mundial	1000.00
10	Estadística Aplicada	1100.00
11	Programación en Java	1200.00
12	Historia del Arte	1300.00

16.

```
--Consulta 16
SELECT TOP 5 E.Nombres, E.Apellidos, SUM(P.Monto) AS TotalPagado
FROM Estudiantes AS E
JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY E.EstudianteID, E.Nombres, E.Apellidos
ORDER BY TotalPagado DESC;

--Consulta 17
SELECT TOP 1 C.NombreCurso, SUM(P.Monto) AS MontoTotalPagado
```

90 %

Results Messages

	Nombres	Apellidos	TotalPagado
1	Isabel	Vega	2000.00
2	Jorge	Castro	1900.00
3	Patricia	Ortiz	1800.00
4	Fernando	Morales	1700.00
5	Elena	Ruiz	1600.00

17.

```
--Consulta 17
SELECT TOP 1 C.NombreCurso, SUM(P.Monto) AS MontoTotalRecaudado
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY C.CursoID, C.NombreCurso
ORDER BY MontoTotalRecaudado DESC;
```

90 %

Results Messages

	NombreCurso	MontoTotalRecaudado
1	Geografía Física	2000.00

18.

```
--Consulta 18
SELECT E.Nombres, E.Apellidos, AVG(P.Monto) AS PromedioPagosPorEstudiante
FROM Estudiantes AS E
JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
GROUP BY E.EstudianteID, E.Nombres, E.Apellidos;
```

90 %

Results Messages

	Nombres	Apellidos	PromedioPagosPorEstudiante
1	Ana	Perez	100.000000
2	Luis	Gomez	200.000000
3	Carlos	Lopez	300.000000
4	Maria	Rodriguez	400.000000
5	Jose	Martinez	500.000000
6	Laura	Garcia	600.000000
7	David	Hernandez	700.000000

19.

```
--Consulta 19
SELECT C.NombreCurso, C.DuracionSemanas, COUNT(M.EstudianteID) AS NumeroEstudiantes
FROM Cursos AS C
JOIN Matriculas AS M ON CCursoID = M.CursoID
WHERE C.DuracionSemanas > (SELECT AVG(DuracionSemanas) FROM Cursos)
GROUP BY C.CursoID, C.NombreCurso, C.DuracionSemanas
HAVING COUNT(M.EstudianteID) > 2;
```

90 %

Results Messages

	NombreCurso	DuracionSemanas	NumeroEstudiantes
--	-------------	-----------------	-------------------

20.

```
--Consulta 20
SELECT TOP 5 E.Nombres AS NombreEstudiante, C.NombreCurso, P.Monto, M.FechaMatricula
FROM Estudiantes AS E
JOIN Matriculas AS M ON E.EstudianteID = M.EstudianteID
JOIN Cursos AS C ON M.CursoID = C.CursoID
JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
ORDER BY M.FechaMatricula DESC, P.FechaPago DESC;
```

100 %

Results Messages

	NombreEstudiante	NombreCurso	Monto	FechaMatricula
1	Isabel	Geografía Física	2000.00	2025-01-20
2	Jorge	Economía Avanzada	1900.00	2025-01-19
3	Patricia	Arte Clásico	1800.00	2025-01-18
4	Fernando	Biología Celular	1700.00	2025-01-17
5	Elena	Literatura Latinoamericana	1600.00	2025-01-16