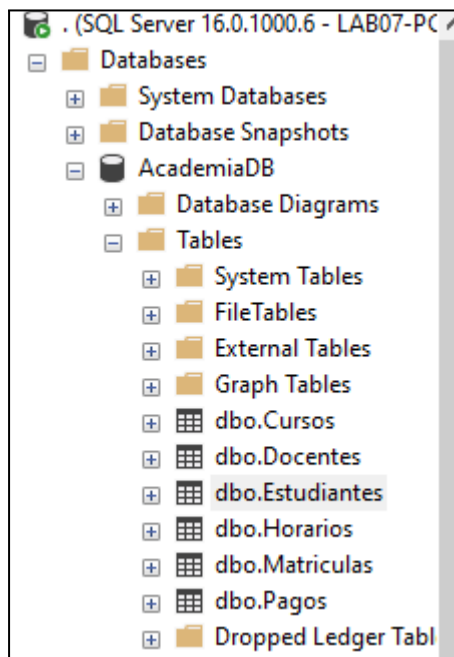


## BASE DE DATOS UTILIZADA



### Funciones de agregación (SUM, COUNT, AVG, MIN, MAX)

1. Calcular el total de pagos realizados por todos los estudiantes.
2. Obtener el monto promedio pagado por matrícula.
3. Mostrar el monto mínimo y máximo pagado por los estudiantes.
4. Contar cuántas matrículas se han registrado en total.
5. Mostrar cuántos cursos tienen una duración mayor al promedio de duración.

1.

```
SQLQuery3.sql - (L...0\LAB07-PC01 (166))* -> X SQLQuery1.sql
SELECT SUM(Monto) AS TotalPagosRealizados
FROM Pagos;
```

100 %

Results Messages

	TotalPagosRealizados
1	21000.00

2.

```
SELECT AVG(Monto) AS MontoPromedioPorMatricula
FROM Pagos;
```

%

Results Messages

	MontoPromedioPorMatricula
	1050.000000

3.

```
SELECT MIN(Monto) AS MontoMinimoPagado, MAX(Monto) AS MontoMaximoPagado
FROM Pagos;
```

	MontoMinimoPagado	MontoMaximoPagado
1	100.00	2000.00

4.

```
SELECT COUNT(MatriculaID) AS TotalMatriculasRegistradas
FROM Matriculas;
```

	TotalMatriculasRegistradas
1	20

5.

```
SELECT COUNT(CursoID) AS CursosConDuracionMayorAlPromedio
FROM Cursos
WHERE DuracionSemanas > (SELECT AVG(DuracionSemanas) FROM Cursos);
```

	CursosConDuracionMayorAlPromedio
1	6

## Agrupación de datos (GROUP BY)

6. Listar el total de pagos realizados por cada estudiante.
7. Mostrar cuántos estudiantes se han matriculado en cada curso.
8. Obtener el promedio de pagos por curso.
9. Listar los cursos con más de 1 estudiante matriculado.
10. Mostrar el total de pagos por método de pago.

6.

```
SELECT
    E.Nombres,
    E.Apellidos,
    SUM(P.Monto) AS TotalPagosPorEstudiante
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 E.Nombres, E.Apellidos;
```

	Nombres	Apellidos	TotalPagosPorEstudiante
1	Ana	Perez	100.00
2	Andrea	Ramirez	1000.00
3	Carlos	Lopez	300.00
4	Carmen	Sanchez	1400.00
5	Daniel	Gonzalez	900.00
6	David	Hernandez	700.00
7	Elena	Ruiz	1600.00
8	Fernando	Morales	1700.00
9	Isabel	Vega	2000.00
10	Jorge	Castro	1900.00
11	Jose	Martinez	500.00
12	Juan	Diaz	1300.00
13	Laura	Garcia	600.00
14	Lucia	Torres	1200.00
15	Luis	Gomez	200.00
16	Maria	Rodriguez	400.00

7.

```

SELECT
    C.NombreCurso,
    COUNT(M.EstudianteID) AS CantidadEstudiantesMatriculados
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
GROUP BY C.CursoID, C.NombreCurso
ORDER BY C.NombreCurso;

```

	NombreCurso	CantidadEstudiantesMatriculados
1	Arte Clásico	1
2	Arte Moderno	1
3	Biología Celular	1
4	Biología Molecular	1
5	Economía Avanzada	1
6	Economía Básica	1
7	Estadística Aplicada	1
8	Física Aplicada	1
9	Física Cuántica	1
10	Geografía Física	1
11	Geografía Mundial	1
12	Historia del Arte	1
13	Historia Universal	1
14	Literatura Española	1
15	Literatura Latinoamericana	1
16	Matemáticas Básicas	1

8.

```

SELECT
    C.NombreCurso,
    AVG(P.Monto) AS PromedioPagosPorCurso
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 C.NombreCurso;

```

	NombreCurso	PromedioPagosPorCurso
1	Arte Clásico	1800.000000
2	Arte Moderno	800.000000
3	Biología Celular	1700.000000
4	Biología Molecular	700.000000
5	Economía Avanzada	1900.000000
6	Economía Básica	900.000000
7	Estadística Aplicada	1100.000000
8	Física Aplicada	500.000000
9	Física Cuántica	1500.000000
10	Geografía Física	2000.000000
11	Geografía Mundial	1000.000000
12	Historia del Arte	1300.000000
13	Historia Universal	300.000000
14	Literatura Española	600.000000
15	Literatura Latinoamericana	1600.000000
16	Matemáticas Básicas	100.000000

9.

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

10.

```
SELECT
    MetodoPago,
    SUM(Monto) AS TotalPagos
FROM Pagos
GROUP BY MetodoPago
ORDER BY MetodoPago;
```

100 %

Results Messages

	MetodoPago	TotalPagos
1	Tarjeta de crédito	21000.00

### Subconsultas

11. Listar los estudiantes que han pagado más que el promedio general de pagos.
12. Mostrar los cursos en los que se ha matriculado el estudiante con el mayor número de
13. Listar los docentes que dictan cursos con más de 2 horarios asignados.
14. Mostrar los estudiantes que no han realizado ningún pago.
15. Listar los cursos cuyo total de pagos supera el total de pagos del curso 'Matemáticas Básicas'.

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)
ORDER BY E.Nombres, E.Apellidos;
```

100 %

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.

```

SELECT DISTINCT
    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
);

```

Results Messages

	NombreCurso
1	Programación en Python

13.

```

SELECT DISTINCT
    D.Nombres,
    D.Apellidos
FROM Docentes AS D
JOIN Cursos AS C ON D.DocenteID = C.DocenteID
JOIN Horarios AS H ON C.CursoID = H.CursoID
GROUP BY D.DocenteID, D.Nombres, D.Apellidos
HAVING COUNT(H.HorarioID) > 2
ORDER BY D.Nombres, D.Apellidos;

```

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
ORDER BY E.Nombres, E.Apellidos;

```

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'
)
ORDER BY TotalPagosCurso DESC;

```

Results Messages

	NombreCurso	TotalPagosCurso
1	Geografía Física	2000.00
2	Economía Avanzada	1900.00
3	Arte Clásico	1800.00
4	Biología Celular	1700.00
5	Literatura Latinoamericana	1600.00
6	Física Cuántica	1500.00
7	Química Orgánica	1400.00
8	Historia del Arte	1300.00
9	Programación en Java	1200.00
10	Estadística Aplicada	1100.00
11	Geografía Mundial	1000.00
12	Economía Básica	900.00

### Ejercicios combinados (agregación + subconsultas + agrupación)

16. Mostrar los 5 estudiantes que más han pagado en total.
17. Listar los cursos con el mayor monto total recaudado por pagos.
18. Mostrar el promedio de pagos por estudiante, junto con su nombre.
19. Listar los cursos con duración mayor al promedio y que tengan más de 2 estudiantes matriculados.
20. Mostrar el nombre del estudiante, curso y monto pagado para las 5 matrículas más recientes.

16.

```
SELECT TOP 5
    E.Nombres,
    E.Apellidos,
    SUM(P.Monto) AS MontoTotalPagado
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 MontoTotalPagado DESC;
```

	Nombres	Apellidos	MontoTotalPagado
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.

```
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;
```

	NombreCurso	MontoTotalRecaudado
1	Geografía Física	2000.00

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
ORDER BY E.Nombres, E.Apellidos;

```

	Nombres	Apellidos	PromedioPagosPorEstudiante
1	Ana	Perez	100.000000
2	Andrea	Ramirez	1000.000000
3	Carlos	Lopez	300.000000
4	Carmen	Sanchez	1400.000000
5	Daniel	Gonzalez	900.000000
6	David	Hernandez	700.000000
7	Elena	Ruiz	1600.000000
8	Fernando	Morales	1700.000000
9	Isabel	Vega	2000.000000
10	Jorge	Castro	1900.000000
11	Jose	Martinez	500.000000
12	Juan	Diaz	1300.000000
13	Laura	Garcia	600.000000
14	Lucia	Torres	1200.000000
15	Luis	Gomez	200.000000
16	Maria	Rodriguez	400.000000

19.

```

SELECT
    C.NombreCurso,
    C.DuracionSemanas,
    COUNT(M.EstudianteID) AS CantidadEstudiantes
FROM Cursos AS C
JOIN Matriculas AS M ON C.CursoID = M.CursoID
WHERE C.DuracionSemanas > (SELECT AVG(DuracionSemanas) FROM Cursos)
GROUP BY C.CursoID, C.NombreCurso, C.DuracionSemanas
HAVING COUNT(M.EstudianteID) > 2
ORDER BY C.NombreCurso;

```

20.

```

SELECT TOP 5
    E.Nombres AS NombreEstudiante,
    E.Apellidos AS ApellidoEstudiante,
    C.NombreCurso,
    P.Monto AS MontoPagado,
    M.FechaMatricula
FROM Matriculas AS M
JOIN Estudiantes AS E ON M.EstudianteID = E.EstudianteID
JOIN Cursos AS C ON M.CursoID = C.CursoID
LEFT JOIN Pagos AS P ON M.MatriculaID = P.MatriculaID
ORDER BY M.FechaMatricula DESC;

```

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