Víctor Manuel Ridao Chaves
17/03/2025
1º DAW ADAITS

Consultas con la base de datos SistemaEducativo

1. **Obtener todos los estudiantes registrados.**

```
SELECT *
FROM estudiantes;
```

estudiantes (20r × 6c)

#	id_estudiante	nombre	apellido	fecha_nacimiento	genero	email
1	1	Juan	Pérez	2005-04-15	M	juan.perez@email.com
2	2	Ana	Gómez	2006-06-20	F	ana.gomez@email.com
3	3	Carlos	Ramírez	2004-11-10	M	carlos.ramirez@email.com
4	4	Laura	Torres	2005-08-25	F	laura.torres@email.com
5	5	Pedro	Martínez	2006-03-12	M	pedro.martinez@email.com
6	6	Sofía	López	2007-07-18	F	sofia.lopez@email.com
7	7	Daniel	González	2005-09-21	M	daniel.gonzalez@email.com
8	8	Mariana	Díaz	2006-05-30	F	mariana.diaz@email.com

2. **Listar los profesores con su especialidad.**

```
SELECT nombre, apellido, especialidad
FROM profesores;
```

profesores (20r × 3c)

#	nombre	apellido	especialidad
1	María	López	Matemáticas
2	Javier	Fernández	Física
3	Elena	Díaz	Historia
4	Roberto	Sánchez	Lengua
5	Sofía	García	Biología

3. **Mostrar todos los cursos con su profesor asignado.**

```
SELECT c.nombre, p.nombre, p.apellido
FROM cursos c
INNER JOIN profesores p ON c.id_profesor = p.id_profesor;
```

cursos (20r × 3c)

#	nombre	nombre	apellido
1	Álgebra	María	López
2	Física Cuántica	Javier	Fernández
3	Historia Universal	Elena	Díaz
4	Literatura	Roberto	Sánchez
5	Biología	Sofía	García
6	Química Orgánica	Fernando	Ruiz

4. **Obtener todas las asignaturas de un curso específico (ej. ID 1).**

```
SELECT a.id_asignatura, a.nombre, c.id_curso
FROM asignaturas a
INNER JOIN cursos c ON a.id_curso = c.id_curso
WHERE c.id_curso = 3;
```

asignaturas (2r × 3c)

#	id_asignatura	nombre	id_curso
1	3	Edad Media	3
2	23	Revolución Industrial	3

5. **Mostrar los estudiantes que se inscribieron después del 1 de enero de 2024.**

```
SELECT e.*, i.id_estudiante, i.fecha_inscripcion
FROM estudiantes e
INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
WHERE i.fecha_inscripcion > 2024-01-01;
```

estudiantes (20r × 8c)

#	id_estudiante	nombre	apellido	fecha_nacimiento	genero	email	id_estudiante	fecha_inscripcion
1	1	Juan	Pérez	2005-04-15	M	juan.perez@email.com	1	2024-02-01
2	2	Ana	Gómez	2006-06-20	F	ana.gomez@email.com	2	2024-02-05
3	3	Carlos	Ramírez	2004-11-10	M	carlos.ramirez@email.com	3	2024-02-10
4	4	Laura	Torres	2005-08-25	F	laura.torres@email.com	4	2024-02-15
5	5	Pedro	Martínez	2006-03-12	M	pedro.martinez@email.com	5	2024-02-20

6. **Listar las calificaciones de un estudiante específico (ej. ID 2).**

```
SELECT e.nombre, e.apellido, c.nota, c.id_estudiante
FROM estudiantes e
INNER JOIN calificaciones c ON c.id_estudiante = e.id_estudiante
WHERE e.id_estudiante = 6;
```

estudiantes (1r × 4c)

#	nombre	apellido	nota	id_estudiante
1	Sofía	López	7,8	6

7. **Mostrar los pagos realizados en el mes de abril de 2024.**

```
SELECT *
FROM pagos;
```

pagos (20r × 4c)

#	id_pago	id_estudiante	monto	fecha_pago
1	1	1	500,0	2024-04-01
2	2	2	600,0	2024-04-02
3	3	3	450,0	2024-04-03
4	4	4	700,0	2024-04-04
5	5	5	550,0	2024-04-05

8. **Obtener los horarios de clases de un aula específica (ej. ID 3).**

```
SELECT h.*, a.id_aula
FROM horarios h
INNER JOIN aulas a ON h.id_aula = a.id_aula
WHERE a.id_aula = 3;
```

horarios (2r × 7c)

#	id_horario	id_asignatura	id_aula	dia_semana	hora_inicio	hora_fin	id_aula
1	3	3	3	Miércoles	12:00:00	14:00:00	3
2	8	3	3	Miércoles	12:00:00	14:00:00	3

9. **Listar los estudiantes que asistieron a una asignatura específica (ej. ID 5).**

```
SELECT e.*
FROM estudiantes e
INNER JOIN asistencia a ON e.id_estudiante = a.id_estudiante
INNER JOIN asignaturas asig ON a.id_asignatura = asig.id_asignatura
WHERE a.id_asignatura = 4 AND a.presente = 1;
```

estudiantes (1r × 6c)

#	id_estudiante	nombre	apellido	fecha_nacimiento	genero	email
1	4	Laura	Torres	2005-08-25	F	laura.torres@email.com

10. **Mostrar el promedio de notas de cada estudiante.**

```
SELECT e.*, AVG(c.nota) AS promedio_notas
FROM estudiantes e
INNER JOIN calificaciones c ON e.id_estudiante = c.id_estudiante
GROUP BY e.id_estudiante, e.nombre, e.apellido
ORDER BY promedio_notas DESC;
```

estudiantes (20r × 7c)

#	id_estudiante	nombre	apellido	fecha_nacimiento	genero	email	promedio_notas
1	16	Isabel	Rojas	2006-08-11	F	isabel.rojas@email.com	9,8
2	4	Laura	Torres	2005-08-25	F	laura.torres@email.com	9,5
3	8	Mariana	Díaz	2006-05-30	F	mariana.diaz@email.com	9,2
4	13	Ricardo	Suárez	2004-06-10	M	ricardo.suarez@email.com	9,1
5	1	Juan	Pérez	2005-04-15	M	juan.perez@email.com	9,0
6	20	Paula	Reyes	2006-11-13	F	paula.reyes@email.com	9,0
7	14	Elena	Ortega	2007-11-05	F	elena.ortega@email.com	8,7

Subconsultas

(Las subconsultas permiten obtener datos filtrando por valores de otra consulta.)

1. **Obtener el nombre del estudiante con la calificación más alta en toda la base de datos.**

```
SELECT e.*
FROM estudiantes e
WHERE e.id_estudiante = (
        SELECT c.id_estudiante
        FROM calificaciones c
        ORDER BY c.id_calificacion DESC
        LIMIT 1
);
```

estudiantes (1r × 6c)

#	id_estudiante	nombre	apellido	fecha_nacimiento	genero	email
1	20	Paula	Reyes	2006-11-13	F	paula.reyes@email.com

2. **Mostrar los cursos con más de 5 estudiantes inscritos.**

```
SELECT c.id_curso, c.nombre
FROM cursos c
WHERE c.id_curso IN (
        SELECT i.id_curso
        FROM inscripciones i
        GROUP BY i.id_curso
        HAVING COUNT(i.id_estudiante) > 5
);
```

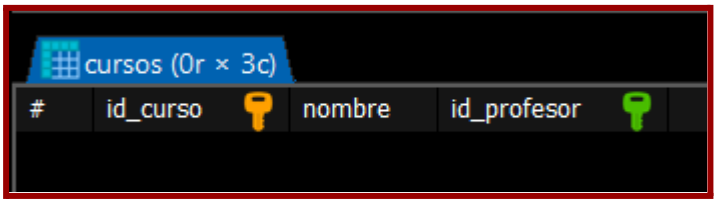
cursos (0r × 2c)

#	id_curso	nombre

*No devuelve nada

3. **Listar los profesores que tienen más de 3 cursos asignados.**

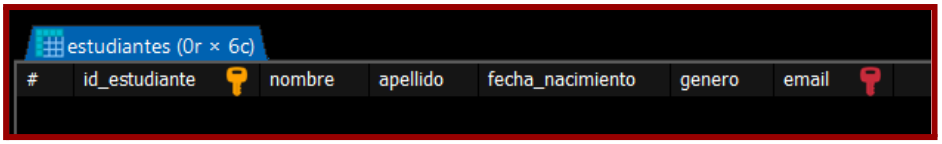
```
SELECT c.*
FROM cursos c
WHERE c.id_profesor IN (
        SELECT p.id_profesor
        FROM profesores p
        GROUP BY p.id_profesor
        HAVING COUNT(c.id_curso) > 3
);
```



*No devuelve nada

4. **Obtener el nombre de los estudiantes que nunca han pagado una cuota.**

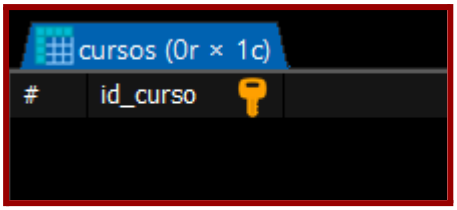
```
SELECT e.*
FROM estudiantes e
WHERE NOT EXISTS (
        SELECT 1
        FROM pagos p
        WHERE e.id_estudiante = p.id_estudiante
);
```



*No devuelve nada

5. **Listar los cursos que no tienen inscripciones.**

```
SELECT c.id_curso
FROM cursos c
WHERE NOT EXISTS (
        SELECT i.id_inscripcion
        FROM inscripciones i
        WHERE c.id_curso = i.id_curso
);
```



*No devuelve nada

6. **Mostrar los estudiantes que tienen un promedio de calificación superior a 8.**

```
SELECT e.nombre, AVG (c.nota) AS promedio
FROM estudiantes e
INNER JOIN calificaciones c ON e.id_estudiante = c.id_estudiante
GROUP BY e.id_estudiante, e.nombre
HAVING AVG (c.nota) > 8;
```

#	nombre	promedio
1	Juan	9,0
2	Ana	8,5
3	Laura	9,5
4	Mariana	9,2
5	Fernanda	8,6
6	Ricardo	9,1
7	Elena	8,7
8	Isabel	9,8
9	Francisco	8,3
10	Paula	9,0

7. **Obtener los nombres de los estudiantes que han asistido a todas sus clases.**

```
SELECT e.nombre, e.apellido
FROM estudiantes e
WHERE NOT EXISTS (
   SELECT 1
   FROM inscripciones i
   INNER JOIN asignaturas a ON i.id_curso = a.id_curso
   WHERE i.id_estudiante = e.id_estudiante
   AND NOT EXISTS (
      SELECT 1
      FROM asistencia asis
      WHERE asis.id_estudiante = e.id_estudiante
      AND asis.id_asignatura = a.id_asignatura
   )
);
```

estudiantes (5r × 2c)

#	nombre	apellido
1	Isabel	Rojas
2	Manuel	Delgado
3	Natalia	Flores
4	Francisco	Mendoza
5	Paula	Reyes

*Problema con esta preguntar en clase

8. **Listar las asignaturas que tienen menos de 3 estudiantes con calificación registrada.**

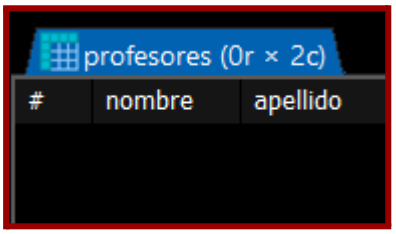
```
SELECT a.nombre
FROM asignaturas a
WHERE (
   SELECT COUNT(c.id_estudiante)
   FROM calificaciones c
   WHERE c.id_asignatura = a.id_asignatura) < 3;
```

asignaturas (35r × 1c)

#	nombre
1	Ecuaciones Diferenciales
2	Mecánica Cuántica
3	Edad Media
4	Siglo de Oro
5	ADN y Genética
6	Reacciones Químicas

9. **Mostrar los profesores que imparten cursos con más de 15 estudiantes inscritos.**

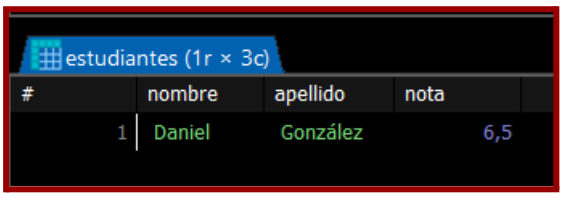
```
SELECT p.nombre, p.apellido
FROM profesores p
WHERE p.id_profesor IN (
   SELECT c.id_profesor
   FROM cursos c
   WHERE (
         SELECT COUNT(*)
         FROM inscripciones i
         WHERE i.id_curso = c.id_curso) > 15
);
```



*No devuelve nada

10. **Obtener a los estudiantes que tienen la menor calificación en alguna asignatura.**

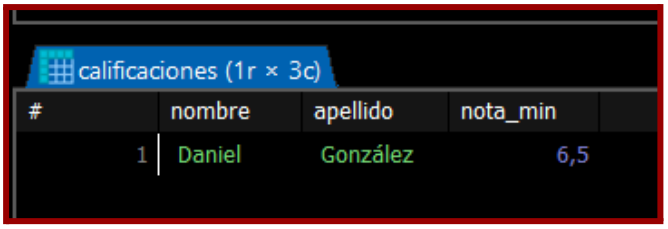
```
SELECT e.nombre, e.apellido, c.nota
FROM estudiantes e, calificaciones c
WHERE e.id_estudiante = c.id_estudiante
AND c.nota = (
         SELECT MIN(c.nota)
         FROM calificaciones c
);
```



*Preguntar por esta

OTRA FORMA DE SACAR LA CONSULTA (ANTONIO)

```
SELECT e.nombre, e.apellido, (
        SELECT c.nota
        FROM calificaciones c
        WHERE c.id_estudiante = e.id_estudiante
        AND c.nota = (
        SELECT MIN(c.nota)
        FROM calificaciones))
                AS nota_min
                FROM estudiantes e
                WHERE e.id_estudiante = (
                SELECT c.id_estudiante
                FROM calificaciones c
                WHERE c.nota = (
                SELECT MIN(c.nota)
                FROM calificaciones c
   )
);
```



calificaciones (1r × 3c)

#	nombre	apellido	nota_min
1	Daniel	González	6,5

*** Se obtiene el mismo**

Inner Join

1. **Obtener los nombres de los estudiantes junto con los cursos en los que están inscritos.**

```
SELECT e.nombre, c.nombre
FROM estudiantes e
INNER JOIN cursos c ON c.id_curso = e.id_estudiante
```

estudiantes (20r × 2c)

#	nombre	nombre
1	Juan	Álgebra
2	Ana	Física Cuántica
3	Carlos	Historia Universal
4	Laura	Literatura
5	Pedro	Biología
6	Sofía	Química Orgánica
7	Daniel	Geografía Mundial
8	Mariana	Educación Física

2. **Listar los profesores y los cursos que imparten.**

```
SELECT p.nombre, c.nombre
FROM profesores p
INNER JOIN cursos c ON c.id_curso = p.id_profesor
```

profesores (20r × 2c)

#	nombre	nombre
1	María	Álgebra
2	Javier	Física Cuántica
3	Elena	Historia Universal
4	Roberto	Literatura
5	Sofía	Biología
6	Fernando	Química Orgánica

3. **Mostrar las calificaciones con los nombres de los estudiantes y asignaturas.**

```
SELECT c.nota, e.nombre, e.apellido, a.nombre
FROM calificaciones c
INNER JOIN estudiantes e ON c.id_estudiante = e.id_estudiante
INNER JOIN asignaturas a ON c.id_asignatura = a.id_asignatura;
```

calificaciones (20r × 4c)

#	nota	nombre	apellido	nombre
1	9,0	Juan	Pérez	Ecuaciones Diferenciales
2	8,5	Ana	Gómez	Mecánica Cuántica
3	7,0	Carlos	Ramírez	Edad Media
4	9,5	Laura	Torres	Siglo de Oro
5	8,0	Pedro	Martínez	ADN y Genética
6	7,8	Sofía	López	Reacciones Químicas

4. **Listar las asistencias con el nombre del estudiante y la asignatura.**

```
SELECT e.nombre, a.nombre, asis.fecha, asis.presente
FROM asistencia asis
INNER JOIN estudiantes e ON asis.id_estudiante = e.id_estudiante
INNER JOIN asignaturas a ON asis.id_asignatura = a.id_asignatura;
```

asistencia (20r × 4c)

#	nombre	nombre	fecha	presente
1	Juan	Ecuaciones Diferenciales	2024-03-10	1
2	Ana	Mecánica Cuántica	2024-03-11	0
3	Carlos	Edad Media	2024-03-12	1
4	Laura	Siglo de Oro	2024-03-13	1
5	Pedro	ADN y Genética	2024-03-14	0
6	Sofía	Reacciones Químicas	2024-03-15	1
7	Daniel	Mapas y Cartografía	2024-03-16	0
8	Mariana	Anatomía Humana	2024-03-17	1

5. **Obtener los horarios de clase con el nombre del aula y la asignatura.**

```
SELECT h.hora_inicio, h.hora_fin, a.numero AS aula, asi.nombre
FROM horarios h
INNER JOIN aulas a ON h.id_aula = a.id_aula
INNER JOIN asignaturas asi ON h.id_asignatura = asi.id_asignatura;
```

horarios (40r × 4c)

#	hora_inicio	hora_fin	aula	nombre
1	08:00:00	10:00:00	A101	Ecuaciones Diferenciales
2	10:00:00	12:00:00	B202	Mecánica Cuántica
3	12:00:00	14:00:00	C303	Edad Media
4	14:00:00	16:00:00	D404	Siglo de Oro
5	16:00:00	18:00:00	E505	ADN y Genética
6	08:00:00	10:00:00	A101	Ecuaciones Diferenciales
7	10:00:00	12:00:00	B202	Mecánica Cuántica
8	12:00:00	14:00:00	C303	Edad Media

Left Join

1. **Listar todos los estudiantes y mostrar los cursos en los que están inscritos (incluir estudiantes sin inscripciones).**

   ```
   SELECT e.nombre, e.apellido, c.nombre
   FROM estudiantes e
   LEFT JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
   LEFT JOIN cursos c ON i.id_curso = c.id_curso;
   ```

   estudiantes (20r × 3c)

#	nombre	apellido	nombre
1	Juan	Pérez	Álgebra
2	Ana	Gómez	Física Cuántica
3	Carlos	Ramírez	Historia Universal
4	Laura	Torres	Literatura
5	Pedro	Martínez	Biología

2. **Mostrar todos los cursos y sus profesores (incluir cursos sin profesor asignado).**

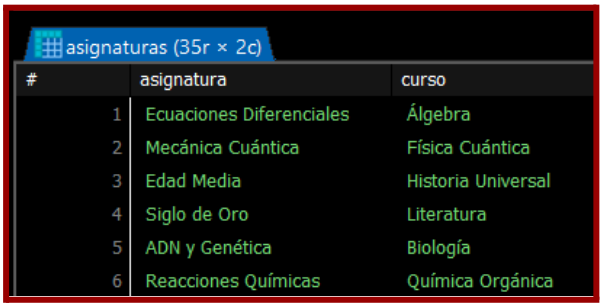
   ```
   SELECT c.nombre AS nombre_curso, p.nombre AS nombre_profesor
   FROM cursos c
   LEFT JOIN profesores p ON c.id_profesor = p.id_profesor;
   ```

   cursos (20r × 2c)

#	nombre_curso	nombre_profesor
1	Álgebra	María
2	Física Cuántica	Javier
3	Historia Universal	Elena
4	Literatura	Roberto
5	Biología	Sofía

3. **Obtener todas las asignaturas junto con el nombre del curso (incluir asignaturas sin grupo asociado).**

```
SELECT a.nombre AS asignatura, c.nombre AS curso
FROM asignaturas a
LEFT JOIN cursos c ON a.id_curso = c.id_curso;
```



asignaturas (35r × 2c)

#	asignatura	curso
1	Ecuaciones Diferenciales	Álgebra
2	Mecánica Cuántica	Física Cuántica
3	Edad Media	Historia Universal
4	Siglo de Oro	Literatura
5	ADN y Genética	Biología
6	Reacciones Químicas	Química Orgánica

Right Join

1. **Mostrar todos los pagos realizados, junto con los datos del estudiante (incluyendo pagos sin un estudiante asociado).**

   ```
   SELECT p.id_pago, p.monto, p.fecha_pago, e.nombre, e.apellido
   FROM pagos p
   RIGHT JOIN estudiantes e ON p.id_estudiante = e.id_estudiante;
   ```

pagos (20r × 5c)

#	id_pago	monto	fecha_pago	nombre	apellido
1	1	500,0	2024-04-01	Juan	Pérez
2	2	600,0	2024-04-02	Ana	Gómez
3	3	450,0	2024-04-03	Carlos	Ramírez
4	4	700,0	2024-04-04	Laura	Torres
5	5	550,0	2024-04-05	Pedro	Martínez
6	6	620,0	2024-04-06	Sofía	López

2. **Listar todos los horarios de clase y las asignaturas correspondientes (incluir horarios sin asignaturas definidas).**

   ```
   SELECT h.hora_inicio, h.hora_fin, a.numero AS aula, asi.nombre
   FROM horarios h
   RIGHT JOIN asignaturas asi ON h.id_asignatura = asi.id_asignatura
   RIGHT JOIN aulas a ON a.id_aula = h.id_aula;
   ```

horarios (40r × 4c)

#	hora_inicio	hora_fin	aula	nombre
1	08:00:00	10:00:00	A101	Ecuaciones Diferenciales
2	08:00:00	10:00:00	A101	Ecuaciones Diferenciales
3	10:00:00	12:00:00	B202	Mecánica Cuántica
4	10:00:00	12:00:00	B202	Mecánica Cuántica
5	12:00:00	14:00:00	C303	Edad Media

3. **Obtener todas las asistencias registradas junto con el nombre del estudiante (incluir asistencias sin un estudiante registrado).**

```
SELECT asis.fecha, asis.id_asignatura, asis.presente, e.nombre, e.apellido
FROM asistencia asis
RIGHT JOIN estudiantes e ON asis.id_estudiante = e.id_estudiante;
```

asistencia (20r × 5c)

#	fecha	id_asignatura	presente	nombre	apellido
1	2024-03-10	1	1	Juan	Pérez
2	2024-03-11	2	0	Ana	Gómez
3	2024-03-12	3	1	Carlos	Ramírez
4	2024-03-13	4	1	Laura	Torres
5	2024-03-14	5	0	Pedro	Martínez

Subconsultas complejas

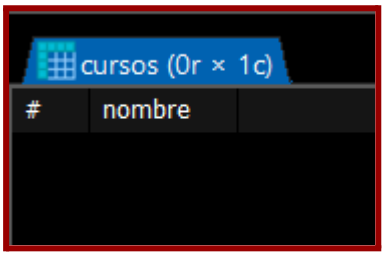
1. **Obtener el estudiante con el mayor promedio de calificaciones.**

```
SELECT e.nombre, e.apellido
FROM estudiantes e
WHERE e.id_estudiante IN (
        SELECT c.id_esudiante
        FROM calificaciones e
        GROUP BY c.id_estudiante
        HAVING AVG (c.nota) = (
                SELECT MAX (promedio)
                FROM (
                        SELECT AVG(c.nota) AS promedio
                        FROM calificaciones c
                        GROUP BY c.id_estudiante
                )
        )
);
```

*No aparece resultado

2. **Listar los cursos con más de 5 estudiantes inscritos.**

```
SELECT c.nombre
FROM cursos c
WHERE c.id_curso IN (
   SELECT i.id_curso
   FROM inscripciones i
   GROUP BY i.id_curso
   HAVING COUNT(i.id_estudiante) > 5
);
```



3. **Obtener los profesores que imparten asignaturas con un promedio de notas mayor a 8.5.**

```
SELECT p.nombre, p.apellido
FROM profesores p
WHERE p.id_profesor IN (
        SELECT c.id_profesor
        FROM cursos c
        WHERE c.id_curso IN (
                SELECT a.id_asignatura
                FROM asignaturas a
                WHERE a.id_asignatura IN (
                        SELECT ca.id_calificacion
                        FROM calificaciones ca
                        GROUP BY ca.id_calificacion
                        HAVING AVG(ca.nota) > 8.5
                )
        )
);
```

profesores (8r × 2c)

#	nombre	apellido
1	María	López
2	Roberto	Sánchez
3	Carlos	Morales
4	Pablo	Gutiérrez
5	Jorge	Navarro
6	Natalia	Domínguez
7	Lucía	Fuentes
8	Felipe	Santos

Consultas con Múltiples JOINs y Funciones Agregadas

1. **Listar los estudiantes junto con su curso y el profesor que les imparte clase.**

```
SELECT e.nombre AS nombre_estudiante, c.nombre AS nombre_curso,
p.nombre AS nombre_profesor
FROM estudiantes e
INNER JOIN inscripciones i ON e.id_estudiante = i.id_estudiante
INNER JOIN cursos c ON i.id_curso = c.id_curso
INNER JOIN profesores p ON c.id_profesor = p.id_profesor
ORDER BY p.nombre;
```

estudiantes (20r × 3c)

#	nombre_estudiante	nombre_curso	nombre_profesor
1	Gabriela	Economía	Álvaro
2	Daniel	Geografía Mundial	Andrea
3	Luis	Filosofía Moderna	Beatriz
4	Mariana	Educación Física	Carlos
5	Alejandro	Programación	Carmen
6	Hugo	Inglés Avanzado	David

2. **Obtener el promedio de notas por curso.**

```
SELECT AVG(c.nota) AS promedio
FROM calificaciones c
INNER JOIN asignaturas a ON c.id_asignatura = a.id_asignatura
INNER JOIN cursos cu ON a.id_curso = cu.id_curso
GROUP BY c.nota
ORDER BY promedio;
```

calificaciones (19r × 1c)

#	promedio
1	6,5
2	6,8
3	6,9
4	7,0
5	7,3
6	7,4

3. **Mostrar los pagos totales de cada estudiante.**

```
SELECT e.nombre, e.apellido, SUM(p.monto) AS pago_total
FROM estudiantes e
INNER JOIN pagos p ON e.id_estudiante = p.id_estudiante
GROUP BY e.id_estudiante;
```

estudiantes (20r × 3c)

#	nombre	apellido	pago_total
1	Juan	Pérez	500,0
2	Ana	Gómez	600,0
3	Carlos	Ramírez	450,0
4	Laura	Torres	700,0
5	Pedro	Martínez	550,0
6	Sofía	López	620,0

Consultas con Operaciones de Fechas y CASE

1. **Obtener la cantidad de estudiantes inscritos en el último mes.**

```
SELECT COUNT(i.id_estudiante) AS cantidad_estudiantes_inscritos
FROM inscripciones i
INNER JOIN estudiantes e ON i.id_estudiante = e.id_estudiante
WHERE i.fecha_inscripcion >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
```

inscripciones (1r × 1c)

#	cantidad_estudiantes_inscritos
1	0

2. **Listar los pagos realizados en el último año agrupados por mes.**

*Preguntar en clase

3. **Mostrar si un estudiante ha pagado o no, usando CASE.**

```
SELECT e.nombre, e.apellido,
CASE WHEN p.id_estudiante IS NOT NULL THEN 'Pagado' ELSE 'No pagado'
END AS estado_pago
FROM estudiantes e
LEFT JOIN pagos p ON e.id_estudiante = p.id_estudiante;
```

estudiantes (20r × 3c)

#	nombre	apellido	estado_pago
1	Juan	Pérez	Pagado
2	Ana	Gómez	Pagado
3	Carlos	Ramírez	Pagado
4	Laura	Torres	Pagado
5	Pedro	Martínez	Pagado

Consultas extra

1. **Mostrar los estudiantes que han pagado más del promedio general de pagos.**

```
SELECT e.nombre, e.apellido, SUM(p.monto) AS pago_ttotal
FROM estudiantes e
INNER JOIN pagos p ON e.id_estudiante = p.id_estudiante
GROUP BY e.id_estudiante
HAVING SUM(p.monto) > (
        SELECT AVG(monto)
        FROM pagos
);
```

estudiantes (8r × 3c)

#	nombre	apellido	pago_ttotal
1	Laura	Torres	700,0
2	Mariana	Díaz	730,0
3	Fernanda	Castro	690,0
4	Ricardo	Suárez	800,0
5	Elena	Ortega	670,0
6	Isabel	Rojas	890,0
7	Francisco	Mendoza	920,0
8	Paula	Reyes	950,0

2. **Obtener los cursos que tienen al menos 3 asignaturas registradas.**

```
SELECT c.nombre
FROM cursos c
INNER JOIN asignaturas a ON c.id_curso = a.id_curso
GROUP BY c.id_curso
HAVING COUNT(a.id_asignatura) >= 3;
```

cursos (0r × 1c)

#	nombre

3. **Listar los profesores que imparten cursos donde más del 50% de los estudiantes tienen un promedio superior a 8.0.**

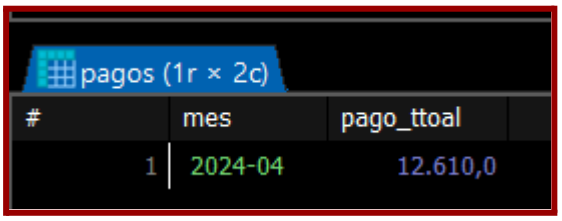
```
SELECT p.nombre, p.apellido, p.especialidad
FROM profesores p
INNER JOIN cursos c ON p.id_profesor = c.id_profesor
INNER JOIN asignaturas a ON c.id_curso = a.id_curso
INNER JOIN calificaciones cal ON a.id_asignatura = cal.id_asignatura
INNER JOIN estudiantes e ON cal.id_estudiante = e.id_estudiante
GROUP BY p.id_profesor
HAVING SUM(
        CASE WHEN cal.nota > 8.0 THEN 1 ELSE 0 END) /
        COUNT(DISTINCT e.id_estudiante) > 0.5;
```

profesores (10r × 3c)

#	nombre	apellido	especialidad
1	María	López	Matemáticas
2	Javier	Fernández	Física
3	Roberto	Sánchez	Lengua
4	Carlos	Morales	Educación Física
5	Pablo	Gutiérrez	Arte
6	Jorge	Navarro	Psicología
7	Natalia	Domínguez	Sociología
8	Lucía	Fuentes	Francés
9	Patricia	Ramos	Teatro
10	Felipe	Santos	Robótica

4. **Calcular el total de pagos por mes en el último año.**

```
SELECT DATE_FORMAT(p.fecha_pago, '%Y-%m') AS mes, SUM(p.monto) AS
pago_total
FROM pagos p
WHERE p.fecha_pago >= CURDATE() - INTERVAL 1 YEAR
GROUP BY mes
ORDER BY mes DESC;
```

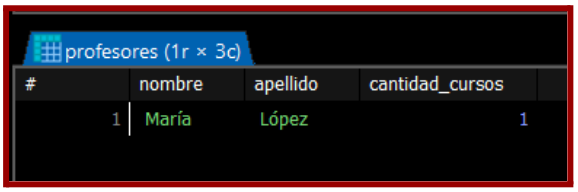


pagos (1r × 2c)

#	mes	pago_ttoal
1	2024-04	12.610,0

5. **Obtener el nombre del profesor que imparte más cursos.**

```
SELECT p.nombre, p.apellido, COUNT(c.id_curso) AS cantidad_cursos
FROM profesores p
INNER JOIN cursos c ON p.id_profesor = c.id_profesor
GROUP BY p.id_profesor
ORDER BY cantidad_cursos DESC
LIMIT 1;
```



profesores (1r × 3c)

#	nombre	apellido	cantidad_cursos
1	María	López	1