UD3. Acceso a Bases de Datos SQL

Repaso SQL

Acceso a Datos

Helena Brau Bou

Curso 2º DAM 2025/26

Índice

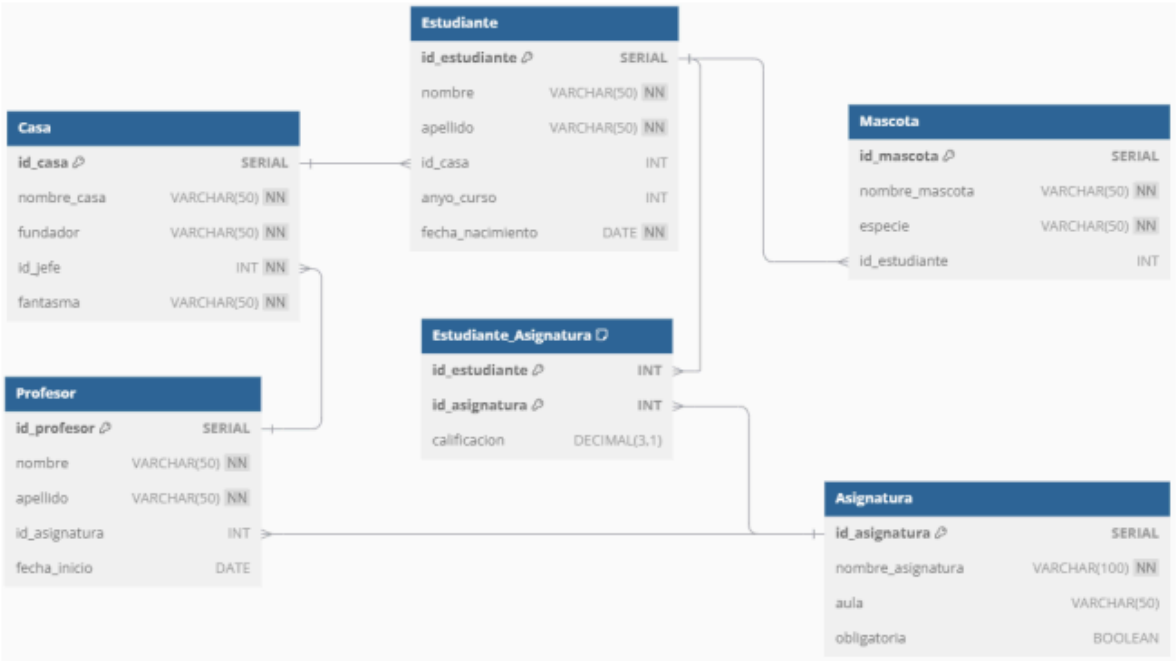
ACTIVIDAD

En esta práctica, aplicarás los conceptos aprendidos sobre bases de datos relacionales utilizando SQL. Debes resolver cada uno de los enunciados que se presentan a continuación, escribiendo las sentencias SQL necesarias para obtener los resultados solicitados.

BASE DE DATOS

Trabajarás con la base de datos **Hogwarts**, que contiene las siguientes tablas principales:

- **Casa:** información sobre las casas de Hogwarts.
- **Estudiante:** información sobre los estudiantes, incluyendo a qué casa pertenecen.
- **Mascota:** información sobre las mascotas que poseen algunos estudiantes.
- **Asignatura:** listado de las asignaturas impartidas en Hogwarts.
- **Profesor:** información sobre los profesores y las asignaturas que enseñan.
- **Estudiante_Asignatura:** calificaciones de los estudiantes en cada asignatura.



EJERCICIOS

1. Consulta básica con SELECT

Selecciona los nombres y apellidos de todos los profesores.

```
SELECT nombre, apellido
FROM Profesor;
```

2. Filtrado de datos con WHERE

Selecciona los nombres y apellidos de los estudiantes nacidos después del 1 de enero de 1980.

```
SELECT nombre, apellido
FROM Estudiante
WHERE fecha_nacimiento > '1980-01-01';
```

3. Ordenación de resultados con ORDER BY

Muestra los nombres y apellidos de los estudiantes, ordenados por su fecha de nacimiento de forma ascendente.

```
SELECT nombre, apellido
FROM Estudiante
ORDER BY fecha_nacimiento ASC;
```

4. Agrupación de resultados con GROUP BY

Muestra cuántos estudiantes hay en cada casa, mostrando el nombre de la casa y el número de estudiantes.

Opción que muestra el id_casa en vez del nombre de la casa:

```
SELECT id_casa, COUNT(*) AS num_estudiantes
FROM Estudiante
GROUP BY id_casa;
```

Opción con subconsulta:

```
SELECT (SELECT nombre_casa FROM Casa c WHERE c.id_casa =
e.id_casa) AS nombre_casa, COUNT(*) AS num_estudiantes
FROM Estudiante e
GROUP BY e.id_casa;
```

Opción con JOIN:

```
SELECT c.nombre_casa, COUNT(e.id_estudiante) AS num_estudiantes
FROM Estudiante e
JOIN Casa c ON e.id_casa = c.id_casa
GROUP BY c.nombre_casa;
```

5. Uso de funciones de agregación

Calcula la calificación media y la calificación máxima en la asignatura "Pociones".

Opción 1 (sin JOIN)

```
SELECT
   AVG(calificacion) AS calificacion_media,
   MAX(calificacion) AS calificacion_maxima
FROM Estudiante_Asignatura
WHERE id_asignatura = (
   SELECT id_asignatura
   FROM Asignatura
   WHERE nombre_asignatura = 'Pociones'
);
```

Opcion 2 (con JOIN)

```
SELECT
   AVG(ea.calificacion) AS calificacion_media,
```

```
    MAX(ea.calificacion) AS calificacion_maxima
FROM Estudiante_Asignatura ea
JOIN Asignatura a ON ea.id_asignatura = a.id_asignatura
WHERE a.nombre_asignatura = 'Pociones';
```

6. Eliminar duplicados con DISTINCT

Muestra todos los años de curso sin duplicados.

```
SELECT DISTINCT anyo_curso
FROM Estudiante;
```

7. Filtrado de patrones con LIKE

Selecciona los nombres de los estudiantes cuyo apellido empieza con la letra "P".

```
SELECT nombre
FROM Estudiante
WHERE apellido LIKE 'P%';
```

8. Uso de IN para búsqueda múltiple

Muestra los nombres y apellidos de los estudiantes que están en su 4º o 5º año.

```
SELECT nombre, apellido
FROM Estudiante
WHERE anyo_curso IN (4, 5);
```

9. Uso de operadores lógicos (AND, OR)

Selecciona los nombres y apellidos de los estudiantes que están en el 5º año y pertenecen a las casas Gryffindor o Slytherin.

Opción SIN JOIN:

```
SELECT nombre, apellido
FROM Estudiante
WHERE anyo_curso = 5
  AND id_casa IN (
      SELECT id_casa
      FROM Casa
      WHERE nombre_casa = 'Gryffindor' OR nombre_casa = 'Slytherin'
  );
```

Opción CON JOIN:

```
SELECT e.nombre, e.apellido
FROM Estudiante e
JOIN Casa c ON e.id_casa = c.id_casa
WHERE e.anyo_curso = 5
  AND (c.nombre_casa = 'Gryffindor' OR c.nombre_casa = 'Slytherin');
```

10. Limitar resultados con LIMIT

Muestra los primeros 5 estudiantes ordenados por su fecha de nacimiento.

```
SELECT nombre, apellido
FROM Estudiante
ORDER BY fecha_nacimiento ASC
LIMIT 5;
```

11. Uso de operadores de comparación

Muestra los nombres de los estudiantes cuya calificación en la asignatura de "Vuelo" es mayor o igual a 8.

```
SELECT e.nombre, e.apellido
FROM Estudiante e
JOIN Estudiante_Asignatura ea ON e.id_estudiante = ea.id_estudiante
JOIN Asignatura a ON ea.id_asignatura = a.id_asignatura
WHERE a.nombre_asignatura = 'Vuelo'
 AND ea.calificacion >= 8;
```

12. Inserción de datos con INSERT INTO

Inserta un nuevo estudiante llamado "Nymphadora Tonks" en la casa Slytherin (id_casa = 4), en el 7º año, con fecha de nacimiento '1973-11-25'.

```
INSERT INTO Estudiante (nombre, apellido, id_casa, anyo_curso,
fecha_nacimiento)
VALUES ('Nymphadora', 'Tonks', 4, 7, '1973-11-25');
```

13. Modificación de datos con UPDATE

Cambia el jefe de la casa Hufflepuff a Pomona Sprout.

```
UPDATE Casa
SET id_jefe = (
   SELECT id_profesor
   FROM Profesor
   WHERE nombre = 'Pomona' AND apellido = 'Sprout'
)
WHERE nombre_casa = 'Hufflepuff';
```

14. Eliminación de datos con DELETE

Elimina al estudiante con nombre "Tom Riddle".

```
DELETE FROM Estudiante
WHERE nombre = 'Tom' AND apellido = 'Riddle';
```

15. Uso de JOIN para combinar tablas

Selecciona los nombres y apellidos de los estudiantes junto con el nombre de su casa.

```
SELECT e.nombre, e.apellido, c.nombre_casa
FROM Estudiante e
INNER JOIN Casa c ON e.id_casa = c.id_casa;
```

16. Uso de LEFT JOIN para combinar tablas

Muestra los nombres de los estudiantes junto con los nombres de las mascotas y las asignaturas que cursan. Incluye a los estudiantes que no tienen mascota.

```
SELECT e.nombre, e.apellido, m.nombre_mascota, a.nombre_asignatura
FROM Estudiante e
LEFT JOIN Mascota m ON e.id_estudiante = m.id_estudiante
JOIN Estudiante_Asignatura ea ON e.id_estudiante = ea.id_estudiante
JOIN Asignatura a ON ea.id_asignatura = a.id_asignatura;
```

17. Uso de subconsultas

Muestra los nombres de los estudiantes que tienen una calificación superior al promedio en la asignatura "Encantamientos".

```
SELECT nombre, apellido
FROM Estudiante
WHERE id_estudiante IN (
    SELECT id_estudiante
```

```
    FROM Estudiante_Asignatura
    WHERE id_asignatura = (
        SELECT id_asignatura
        FROM Asignatura
        WHERE nombre_asignatura = 'Encantamientos'
    )
    AND calificacion > (
        SELECT AVG(calificacion)
        FROM Estudiante_Asignatura
        WHERE id_asignatura = (
            SELECT id_asignatura
            FROM Asignatura
            WHERE nombre_asignatura = 'Encantamientos'
        )
    )
);
```

18. Uso de HAVING

Muestra los nombres de las casas que tienen un número promedio de calificaciones superior a 7 en las asignaturas de los estudiantes de esa casa.

```
SELECT c.nombre_casa
FROM Casa c
JOIN Estudiante e ON c.id_casa = e.id_casa
JOIN Estudiante_Asignatura ea ON e.id_estudiante = ea.id_estudiante
GROUP BY c.nombre_casa
HAVING AVG(ea.calificacion) > 7;
```