

## PARTE 1: Consultas Básicas (SELECT, WHERE, ORDER BY)

### Ejercicio 1.1

Muestra el código, nombre y apellido de todos los profesores, ordenados alfabéticamente por apellido.

*Pista: Usa `ORDER BY` con la columna de apellidos.*

### Ejercicio 1.2

Encuentra todas las materias que tienen 4 créditos.

### Ejercicio 1.3

Lista los estudiantes que ingresaron en agosto de 2023 (mes 8).

*Pista: Usa `MONTH(fecha\_ingreso) = 8`.*

### Ejercicio 1.4

Muestra el nombre de todas las facultades cuyo decano no sea nulo.

### Ejercicio 1.5

Encuentra las inscripciones del año 2023 con nota final mayor o igual a 4.0.

## PARTE 2: Funciones y Alias

### Ejercicio 2.1

Muestra el nombre completo de cada estudiante en una sola columna llamada `nombre\_completo`, y cuántos años lleva en la universidad (columna `antiguedad`).

Formato esperado:

nombre_completo	antiguedad
Juan Carlos Rodríguez Pérez	3

*Pista: Usa `CONCAT(nombres, ' ', apellidos)` y `YEAR(CURDATE()) - YEAR(fecha\_ingreso)`.*

### Ejercicio 2.2

Cuenta cuántas materias hay en total y muestra el resultado con el alias `total\_materias`.

## Ejercicio 2.3

Muestra la nota final de cada inscripción, pero si es NULL muestra 'Sin nota'. Usa el alias ``estado_nota``.

## PARTE 3: JOINS con Dos Tablas

### Ejercicio 3.1 — INNER JOIN

Muestra el nombre de cada profesor y el nombre de la facultad a la que pertenece.  
Tablas: ``profesores`` + ``facultades``

### Ejercicio 3.2 — INNER JOIN con filtro

Muestra solo los profesores de la facultad de Ingeniería.

### Ejercicio 3.3 — LEFT JOIN

Muestra TODAS las facultades y sus profesores. Si una facultad no tiene profesores, debe aparecer con NULL.

*Pregunta: ¿Qué facultad aparece sin profesores asignados?*

### Ejercicio 3.4 — RIGHT JOIN

Muestra TODOS los profesores y las facultades. Si un profesor no tiene facultad asignada, debe aparecer con NULL.

Nota: Para probar esto, primero inserta este profesor sin facultad:

```
sql
```

```
INSERT INTO profesores (codigo, nombres, apellidos, id_facultad, especialidad)  
VALUES ('P004', 'Laura', 'Vargas Peña', NULL, 'Física Cuántica');
```

### Ejercicio 3.5 — JOIN con alias

Muestra el código del estudiante, su nombre completo (concatenado) y el nombre de su carrera. Usa alias de tabla (``e`` y ``c``).

## PARTE 4: JOINS con Tres o Más Tablas

### Ejercicio 4.1

Muestra el nombre del estudiante, la materia en la que está inscrito, y la nota final.

Tablas: `estudiantes` → `inscripciones` → `materias`

### Ejercicio 4.2

Muestra el nombre del estudiante, su carrera, su facultad, y la materia en la que está inscrito.

Tablas: `estudiantes` → `carreras` → `facultades` + `inscripciones` → `materias`

*Pista: Necesitas unir 4 tablas. Empieza con `estudiantes`, une `carreras`, luego `inscripciones`, luego `materias`, y opcionalmente `facultades`.*

### Ejercicio 4.3 — Materias sin inscripciones

Encuentra las materias que NO tienen ningún estudiante inscrito.

*Pista: Usa `LEFT JOIN` con `inscripciones` y filtra donde `id\_inscripcion IS NULL`.*

### Ejercicio 4.4 — Estudiantes sin inscripciones

Encuentra los estudiantes que NO se han inscrito en ninguna materia.

*Pregunta: ¿Qué estudiante(es) aparece(n)?*

## PARTE 5: Modificación de Datos (DML)

### Ejercicio 5.1 — UPDATE

El profesor Luis Eduardo Fernández Ruiz (P001) cambió de especialidad a "Inteligencia Artificial". Actualiza su registro.

### Ejercicio 5.2 — INSERT

Agrega una nueva materia:

- Código: WEB101
- Nombre: Programación Web I
- Carrera: Tecnología en Desarrollo de Software (id\_carrera = 2)
- Créditos: 3

### Ejercicio 5.3 — INSERT con inscripción

Inscribe al estudiante Ana Lucía Gómez Torres (`id_estudiante = 4`) en la materia Cálculo I (`id_materia = 5`), semestre 1, año 2023, con nota 4.3.

*¡Cuidado! Verifica que no exista ya esa combinación en `inscripciones`.*

### Ejercicio 5.4 — DELETE

Elimina la inscripción de María Fernanda López García (`id_estudiante = 2`) en Programación I (`id_materia = 3`).

### Ejercicio 5.5 — DELETE con verificación

Intenta eliminar la facultad de Ciencias Sociales (`id_facultad = 3`).

*Pregunta: ¿Qué error te da MySQL? ¿Por qué?*

## PARTE 6: Preguntas

1. ¿Por qué `TRUNCATE` es más rápido que `DELETE` pero más peligroso?
2. ¿Qué pasa si cambias INNER JOIN por LEFT JOIN en el ejercicio 4.3? ¿El resultado es el mismo?
3. ¿Cómo harías para mostrar estudiantes que NO tienen carrera asignada, usando solo INNER JOIN? (Se puede? o No se puede?)
4. ¿Por qué `COUNT(\*)` y `COUNT(columna)` pueden dar resultados diferentes?