

Trabajo Práctico - MySQL Veterinaria "Patitas Felices"

Objetivo

Este trabajo práctico tiene como objetivo principal que los estudiantes demuestren sus conocimientos en el diseño, creación y manipulación de bases de datos relacionales utilizando MySQL. A través de la implementación de un sistema de gestión para una veterinaria, se evaluará la capacidad de:

- Diseñar un esquema de base de datos relacional apropiado
- Crear tablas con relaciones de integridad referencial (claves foráneas)
- Realizar operaciones CRUD (Create, Read, Update, Delete)
- Aplicar consultas complejas utilizando JOINS
- Gestionar la integridad de los datos mediante transacciones y eliminaciones en cascada

Entrega

- La entrega se realizará mediante un repositorio público en GitHub.
- El repositorio debe incluir:
 - Scripts SQL con todas las consultas realizadas (archivo `.sql` o múltiples archivos organizados).
- El nombre del repositorio debe ser: `tp-mysql-[nombre-apellido]`.

Requisitos técnicos obligatorios

- MySQL
- Editor de texto o IDE para escribir scripts SQL
- Conocimientos previos en:

- SQL básico (CREATE, INSERT, UPDATE, DELETE, SELECT)
- Claves primarias y foráneas
- Relaciones entre tablas
- Consultas con JOIN

Descripción del proyecto

La veterinaria "Patitas Felices" necesita un sistema de gestión para administrar sus datos. El sistema debe permitir:

- Gestionar dueños de mascotas
- Registrar mascotas y su relación con sus dueños
- Administrar veterinarios y sus especialidades
- Registrar el historial clínico, vinculando mascotas, veterinarios y fechas

El sistema debe garantizar la integridad de los datos mediante relaciones adecuadas y permitir consultas combinadas entre varias tablas.

Ejercicios

Ejercicio 1 – Crear Base de Datos

Crear una base de datos llamada **veterinaria_patitas_felices**.

Ejercicio 2 – Crear tabla **duenos**

Crear la tabla **duenos** con las siguientes columnas:

Columna	Tipo	Restricciones
id	INT	PRIMARY KEY, AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL
apellido	VARCHAR(50)	NOT NULL
telefono	VARCHAR(20)	NOT NULL

direccion	VARCHAR(100)	
-----------	--------------	--

Ejercicio 3 – Crear tabla **mascotas**

Crear la tabla **mascotas** con las siguientes columnas:

Columna	Tipo	Restricciones
id	INT	PRIMARY KEY, AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL
especie	VARCHAR(30)	NOT NULL
fecha_nacimiento	DATE	
id_dueno	INT	FOREIGN KEY → duenos.id

Ejercicio 4 – Crear tabla **veterinarios**

Crear la tabla **veterinarios** con las siguientes columnas:

Columna	Tipo	Restricciones
id	INT	PRIMARY KEY, AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL
apellido	VARCHAR(50)	NOT NULL
matricula	VARCHAR(20)	NOT NULL, UNIQUE
especialidad	VARCHAR(50)	NOT NULL

Ejercicio 5 – Crear tabla **historial_clinico**

Crear la tabla **historial_clinico** con las siguientes columnas:

Columna	Tipo	Restricciones
id	INT	PRIMARY KEY, AUTO_INCREMENT
id_mascota	INT	FOREIGN KEY → mascotas.id

id_veterinario	INT	FOREIGN KEY → veterinarios.id
fecha_registro	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP
descripcion	VARCHAR(250)	NOT NULL

Ejercicio 6 – Insertar registros

Insertar:

- 3 dueños con información completa
- 3 mascotas, cada una asociada a un dueño
- 2 veterinarios con especialidades distintas
- 3 registros de historial clínico

Ejercicio 7 – Actualizar registros

Realizar las siguientes actualizaciones:

1. Cambiar la dirección de un dueño (por ID o nombre).
2. Actualizar la especialidad de un veterinario (por ID o matrícula).
3. Editar la descripción de un historial clínico (por ID).

Ejercicio 8 – Eliminar registros

1. Eliminar una mascota (por ID o nombre).
2. Verificar que se eliminen automáticamente los registros del historial clínico asociados (ON DELETE CASCADE).

Ejercicio 9 – JOIN simple

Consulta que muestre:

- Nombre de la mascota
- Especie

- Nombre completo del dueño (nombre + apellido)

Ejercicio 10 – JOIN múltiple con historial

Consulta que muestre todas las entradas del historial clínico con:

- Nombre y especie de la mascota
- Nombre completo del dueño
- Nombre completo del veterinario
- Fecha de registro
- Descripción

Ordenados por fecha de registro descendente (DESC).

Criterios de evaluación

Criterio	Peso
Ejercicio 01	5%
Ejercicio 02	5%
Ejercicio 03	5%
Ejercicio 04	5%
Ejercicio 05	5%
Ejercicio 06	15%
Ejercicio 07	15%
Ejercicio 08	15%
Ejercicio 09	15%
Ejercicio 10	15%

Recomendaciones

Se valorará especialmente:

- Código SQL limpio y comentado
- Organización del proyecto
- Nomenclatura clara y consistente
- Buenas prácticas:
 - Uso de índices en claves foráneas
 - Restricciones NOT NULL cuando corresponda

Recursos adicionales

- Documentación oficial de MySQL
- Tutoriales básicos de SQL

Soporte

Ante dudas, consultar:

- Documentación de MySQL
- Materiales del curso
- Instructor/a