

PROYECTO FINAL - PARQUE NATURAL



SEGUNDA PRE- ENTREGA

CURSO SQL

FECHA: 01/08/2024

INTEGRANTE (Nombre y apellido, Mail):

Agustina Luz Pellizzaro agusluzpelli@gmail.com

ÍNDICE

1. INTRODUCCIÓN	2
2. OBJETIVO	2
3. HERRAMIENTAS UTILIZADAS	2
4. TEMÁTICA DE LA BASE DE DATOS	2
5. DER	3
6. LISTADO DE TABLAS	4
7. CREACIÓN DE TABLAS	9

1. INTRODUCCIÓN

En el presente proyecto se expone y explica la creación de una base de datos relacional, con sus respectivos componentes, utilizando MySQL como sistema de gestión de la misma. La base de datos permite representar y dar soporte a los diferentes parques nacionales y reservas naturales que posean una diversidad de flora y fauna que están abiertos al público y se promueva el turismo y una variedad de actividades recreativas y educativas. Se explicará a lo largo del documento cuál fue el proceso completo para crear la base de datos, carga de datos, creación de diferentes procedimientos y diferentes objetos para poder administrar la reserva natural.

2. OBJETIVO

El propósito de este proyecto es poder desarrollar de manera completa una base de datos basada en un dominio real como es un parque nacional, priorizando la utilidad que el modelo le puede aportar a los diferentes actores involucrados. Esto permitirá poder aplicar los conocimientos adquiridos durante el curso "SQL" de Coderhouse en un caso práctico.

3. HERRAMIENTAS UTILIZADAS

Para el desarrollo del siguiente proyecto, se utilizaron las herramientas:

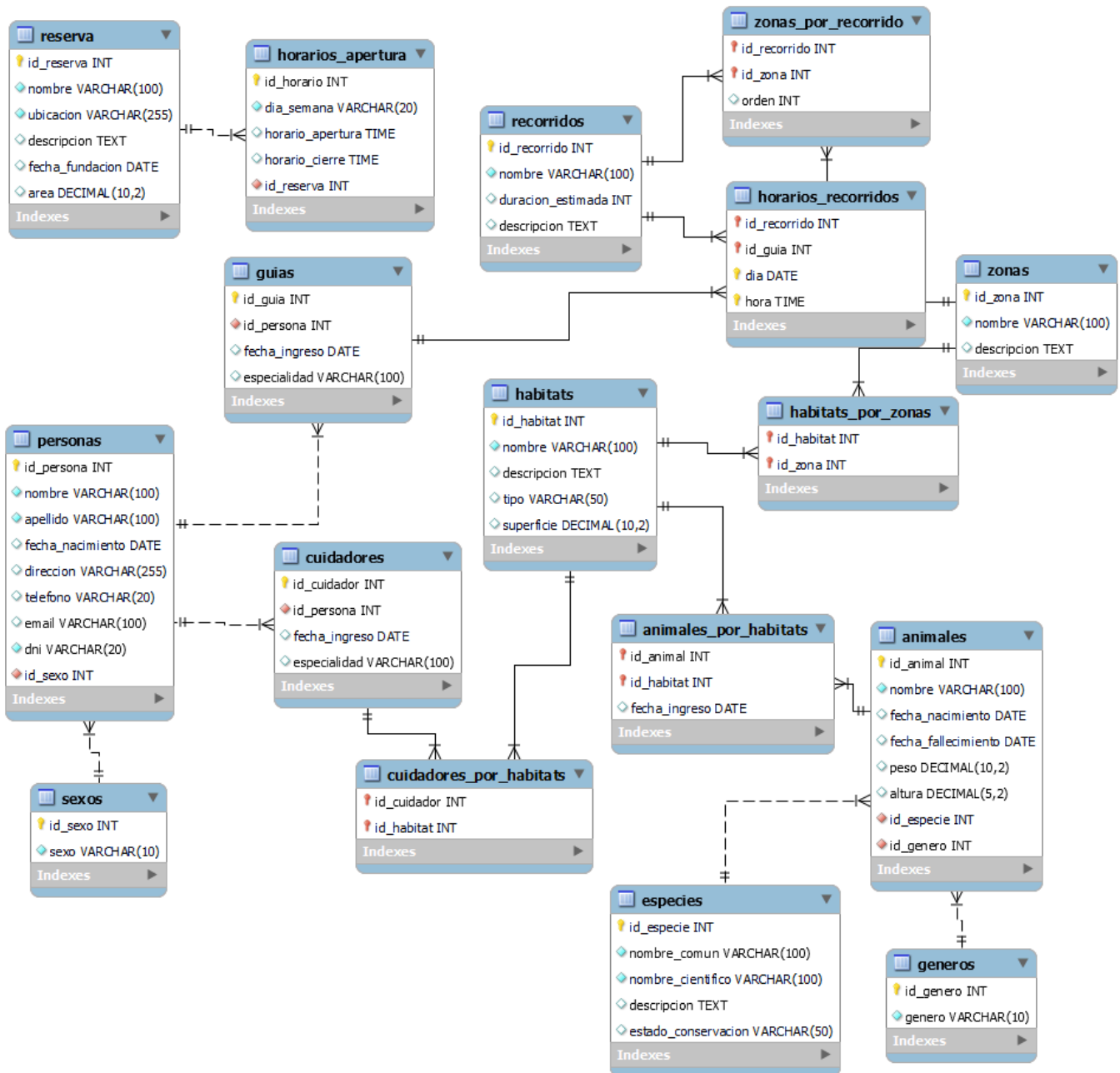
- MySQL Workbench
- Github: https://github.com/aguspellizzaro/proyecto_final_reserva_natural_pellizzaro
- Drive:

4. TEMÁTICA DE LA BASE DE DATOS

La temática es una base de datos que permita representar cualquier parque natural o reserva natural que posea nuestro país, donde el objetivo de las mismas sea poder brindar información a la población sobre las diferentes características del parque y a su vez colaborar a la gestión y conservación del parque. Para este tipo de entidades se suelen manejar grandes volúmenes de datos relacionados con la biodiversidad, especialmente con la fauna del lugar y las diferentes actividades que se pueden realizar. Por lo tanto, decidí crear una base de datos que contenga información sobre las diferentes especies de animales, sus hábitats, los recorridos disponibles en el parque, los guías y cuidadores, los horarios de apertura, y los visitantes. Una base de datos que incluya toda esta información y más puede ser muy útil para recolectar y acceder fácilmente a datos clave, permitiendo así descubrir insights valiosos para la gestión y conservación del parque, y al mismo tiempo proveer información a la población y visitas del parque sobre la importancia del cuidado de la biodiversidad e información educativa.

5. DER

Se desarrolló el diagrama de entidad relación en el MySQL Workbench. Se adjunta a continuación la imagen:



El archivo .mwb y el archivo .png se encuentran dentro del github.

6. LISTADO DE TABLAS

El listado de tablas se encuentra dentro del github con formato .excel.

Se adjunta de igual forma tabla:

Tabla	Descripción	Columna	Descripción	Tipo de Dato	Primary Key	Foreign Key	Uniq ue	Not Null	Auto Increment
reserva	Contiene información sobre las reservas naturales	id_reserva	Identificador único de la reserva	INT	Sí			Sí	Sí
		nombre	Nombre de la reserva	VARCHAR(100)				Sí	
		ubicacion	Ubicación de la reserva	VARCHAR(255)				Sí	
		descripcion	Descripción de la reserva	TEXT					
		fecha_fundacion	Fecha de fundación de la reserva	DATE					
		area	Área de la reserva en metros cuadrados	DECIMAL(10, 2)					
generos	Contiene los géneros de animales	id_genero	Identificador único del género	INT	Sí			Sí	Sí
		genero	Nombre del género	VARCHAR(10)				Sí	
sexos	Contiene los sexos de personas y animales	idsexo	Identificador único del sexo	INT	Sí			Sí	Sí
		sexo	Nombre del sexo	VARCHAR(10)				Sí	
especies	Contiene las especies de animales	id_especie	Identificador único de la especie	INT	Sí			Sí	Sí
		nombre_comun	Nombre común de la especie	VARCHAR(100)				Sí	
		nombre_cientifico	Nombre científico de la especie	VARCHAR(100)				Sí	

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		descripcion	Descripción de la especie	TEXT					
		estado_conservacion	Estado de conservación de la especie	VARCHAR(50)					
animales	Contiene información sobre los animales	id_animal	Identificador único del animal	INT	Sí			Sí	Sí
		nombre	Nombre del animal	VARCHAR(100)				Sí	
		fecha_nacimiento	Fecha de nacimiento del animal	DATE					
		fecha_fallecimiento	Fecha de fallecimiento del animal	DATE					
		peso	Peso del animal en kg	DECIMAL(10, 2)					
		altura	Altura del animal en metros	DECIMAL(5, 2)					
		id_especie	Identificador de la especie	INT		Sí (especies.id_especie)		Sí	
		id_genero	Identificador del género del animal	INT		Sí (generos.id_genero)		Sí	
habitats	Contiene información sobre los hábitats de las reservas	id_habitat	Identificador único del hábitat	INT	Sí			Sí	Sí
		nombre	Nombre del hábitat	VARCHAR(100)				Sí	
		descripcion	Descripción del hábitat	TEXT					
		tipo	Tipo de hábitat	VARCHAR(50)					
		superficie	Superficie del hábitat en metros cuadrados	DECIMAL(10, 2)					
zonas	Contiene información sobre	id_zona	Identificador único de la zona	INT	Sí			Sí	Sí

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

	las zonas dentro de las reservas								
		nombre	Nombre de la zona	VARCHAR(100)			Sí		
		descripcion	Descripción de la zona	TEXT					
habitats_por_zonas	Relaciona hábitats con zonas	id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
		id_zona	Identificador de la zona	INT	Sí	Sí (zonas.id_zona)			
animales_por_habitats	Relaciona animales con hábitats	id_animal	Identificador del animal	INT	Sí	Sí (animales.id_animal)			
		id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
		fecha_ingreso	Fecha de ingreso del animal al hábitat	DATE					
personas	Contiene información sobre las personas	id_persona	Identificador único de la persona	INT	Sí			Sí	Sí
		nombre	Nombre de la persona	VARCHAR(100)			Sí		
		apellido	Apellido de la persona	VARCHAR(100)			Sí		
		fecha_nacimiento	Fecha de nacimiento de la persona	DATE					
		direccion	Dirección de la persona	VARCHAR(255)					
		telefono	Teléfono de la persona	VARCHAR(20)					
		email	Correo electrónico de la persona	VARCHAR(100)					

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		dni	Documento nacional de identidad	VARCHAR(20)			Sí	Sí	
		id_sexo	Identificador del sexo de la persona	INT		Sí (sexos.id_sexo)		Sí	
cuidadores	Contiene información sobre los cuidadores de las reservas	id_cuidador	Identificador único del cuidador	INT	Sí			Sí	Sí
		id_persona	Identificador de la persona	INT		Sí (personas.id_persona)			
		fecha_ingreso	Fecha de ingreso como cuidador	DATE					
		especialidad	Especialidad del cuidador	VARCHAR(100)					
cuidadores_por_habitats	Relaciona cuidadores con hábitats	id_cuidador	Identificador del cuidador	INT	Sí	Sí (cuidadores.id_cuidador)			
		id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
guias	Contiene información sobre los guías de los recorridos	id_guia	Identificador único del guía	INT	Sí			Sí	Sí
		id_persona	Identificador de la persona	INT		Sí (personas.id_persona)			
		fecha_ingreso	Fecha de ingreso como guía	DATE					
		especialidad	Especialidad del guía	VARCHAR(100)					
recorridos	Contiene información sobre los recorridos dentro de las reservas	id_recorrido	Identificador único del recorrido	INT	Sí			Sí	Sí

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		nombre	Nombre del recorrido	VARCHAR(100)				Sí	
		duracion_estimada	Duración estimada del recorrido en minutos	INT					
		descripcion	Descripción del recorrido	TEXT					
zonas_por_recorrido	Relaciona zonas con recorridos	id_recorrido	Identificador del recorrido	INT	Sí	Sí (recorridos.id_recorrido)			
		id_zona	Identificador de la zona	INT	Sí	Sí (zonas.id_zona)			
		orden	Orden del recorrido dentro de la zona	INT					
horarios_apertura	Contiene los horarios de apertura de las reservas	id_horario	Identificador único del horario	INT	Sí			Sí	Sí
		dia_semana	Día de la semana	VARCHAR(20)				Sí	
		horario_apertura	Hora de apertura	TIME					
		horario_cierre	Hora de cierre	TIME					
		id_reserva	Identificador de la reserva	INT		Sí (reserva.id_reserva)		Sí	
horarios_recorridos	Contiene los horarios de recorridos con guías	id_recorrido	Identificador del recorrido	INT		Sí (recorridos.id_recorrido)			
		id_guia	Identificador del guía	INT		Sí (guias.id_guia)		Sí	
		dia	Día del horario	DATE					
		hora	Hora del horario	TIME					

7. CREACIÓN DE TABLAS

El archivo .sql con el script de creación de tablas se encuentra dentro del github:

Se adjunta de igual script:

```
CREATE DATABASE IF NOT EXISTS trabajo_final_pellizzaro;
USE trabajo_final_pellizzaro;
```

```
CREATE SCHEMA IF NOT EXISTS reservas_naturales;
USE reservas_naturales;
```

-- Tabla de Reserva

```
CREATE TABLE IF NOT EXISTS reserva (
  id_reserva INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  ubicacion VARCHAR(255) NOT NULL,
  descripcion TEXT,
  fecha_fundacion DATE,
  area DECIMAL(10, 2) -- Área en metros cuadrados
);
```

-- Tabla de Generos

```
CREATE TABLE IF NOT EXISTS generos (
  id_genero INT AUTO_INCREMENT PRIMARY KEY,
  genero VARCHAR(10) NOT NULL
);
```

-- Tabla de Sexos

```
CREATE TABLE IF NOT EXISTS sexos (
  id_sexo INT AUTO_INCREMENT PRIMARY KEY,
  sexo VARCHAR(10) NOT NULL
);
```

-- Tabla de Especies de Animales

```
CREATE TABLE IF NOT EXISTS especies (
  id_especie INT AUTO_INCREMENT PRIMARY KEY,
  nombre_comun VARCHAR(100) NOT NULL,
  nombre_cientifico VARCHAR(100) NOT NULL,
  descripcion TEXT,
  estado_conservacion VARCHAR(50)
);
```

```
CREATE INDEX idx_especies_nombre_comun ON especies(nombre_comun);
CREATE INDEX idx_especies_nombre_cientifico ON especies(nombre_cientifico);
```

-- Tabla de Animales

```
CREATE TABLE IF NOT EXISTS animales (
```

```

id_animal INT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
fecha_nacimiento DATE,
fecha_fallecimiento DATE,
peso DECIMAL(10, 2), -- Peso del animal en kg
altura DECIMAL(5, 2), -- Altura del animal en metros
id_especie INT NOT NULL,
id_genero INT NOT NULL,
CONSTRAINT fk_animales_especie FOREIGN KEY (id_especie) REFERENCES especies(id_especie),
CONSTRAINT fk_animales_genero FOREIGN KEY (id_genero) REFERENCES generos(id_genero)
);

CREATE INDEX idx_animales_nombre ON animales(nombre);

-- Tabla de Habitats
CREATE TABLE IF NOT EXISTS habitats (
    id_habitat INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    tipo VARCHAR(50),
    superficie DECIMAL(10, 2) -- Superficie en metros cuadrados
);

CREATE INDEX idx_habitats_nombre ON habitats(nombre);

-- Tabla de Zonas
CREATE TABLE IF NOT EXISTS zonas (
    id_zona INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);

CREATE INDEX idx_zonas_nombre ON zonas(nombre);

-- Tabla de Habitats por Zonas
CREATE TABLE IF NOT EXISTS habitats_por_zonas (
    id_habitat INT,
    id_zona INT,
    PRIMARY KEY (id_habitat, id_zona),
    CONSTRAINT fk_habitats_por_zonas_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat),
    CONSTRAINT fk_habitats_por_zonas_zona FOREIGN KEY (id_zona) REFERENCES zonas(id_zona)
);

-- Tabla de Animales por Habitats
CREATE TABLE IF NOT EXISTS animales_por_habitats (
    id_animal INT,

```

```

id_habitat INT,
fecha_ingreso DATE,
PRIMARY KEY (id_animal, id_habitat),
        CONSTRAINT fk_animales_por_habitats_animal FOREIGN KEY (id_animal) REFERENCES
animales(id_animal),
        CONSTRAINT fk_animales_por_habitats_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat)
);

```

-- Tabla de Personas

```

CREATE TABLE IF NOT EXISTS personas (
    id_persona INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE,
    direccion VARCHAR(255),
    telefono VARCHAR(20),
    email VARCHAR(100),
    dni VARCHAR(20) UNIQUE NOT NULL,
    id_sexo INT NOT NULL,
    CONSTRAINT fk_personas_sexo FOREIGN KEY (id_sexo) REFERENCES sexos(id_sexo)
);

```

```

CREATE INDEX idx_personas_dni ON personas(dni);

```

-- Tabla de Cuidadores

```

CREATE TABLE IF NOT EXISTS cuidadores (
    id_cuidador INT AUTO_INCREMENT PRIMARY KEY,
    id_persona INT NOT NULL,
    fecha_ingreso DATE,
    especialidad VARCHAR(100),
    CONSTRAINT fk_cuidadores_persona FOREIGN KEY (id_persona) REFERENCES personas(id_persona)
);

```

-- Tabla de Cuidadores por Habitats

```

CREATE TABLE IF NOT EXISTS cuidadores_por_habitats (
    id_cuidador INT,
    id_habitat INT,
    PRIMARY KEY (id_cuidador, id_habitat),
        CONSTRAINT fk_cuidadores_por_habitats_cuidador FOREIGN KEY (id_cuidador) REFERENCES
cuidadores(id_cuidador),
        CONSTRAINT fk_cuidadores_por_habitats_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat)
);

```

-- Tabla de Guías

```

CREATE TABLE IF NOT EXISTS guias (

```

```
id_guia INT AUTO_INCREMENT PRIMARY KEY,
id_persona INT NOT NULL,
fecha_ingreso DATE,
especialidad VARCHAR(100),
CONSTRAINT fk_guias_persona FOREIGN KEY (id_persona) REFERENCES personas(id_persona)
);
```

-- Tabla de Recorridos

```
CREATE TABLE IF NOT EXISTS recorridos (
    id_recorrido INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    duracion_estimada INT, -- en minutos
    descripcion TEXT
);
```

```
CREATE INDEX idx_recorridos_nombre ON recorridos(nombre);
```

-- Tabla de Zonas por Recorrido

```
CREATE TABLE IF NOT EXISTS zonas_por_recorrido (
    id_recorrido INT,
    id_zona INT,
    orden INT,
    PRIMARY KEY (id_recorrido, id_zona),
    CONSTRAINT fk_zonas_por_recorrido_recorrido FOREIGN KEY (id_recorrido) REFERENCES
    recorridos(id_recorrido),
    CONSTRAINT fk_zonas_por_recorrido_zona FOREIGN KEY (id_zona) REFERENCES zonas(id_zona)
);
```

-- Tabla de Horarios de Apertura

```
CREATE TABLE IF NOT EXISTS horarios_apertura (
    id_horario INT AUTO_INCREMENT PRIMARY KEY,
    dia_semana VARCHAR(20) NOT NULL,
    horario_apertura TIME,
    horario_cierre TIME,
    id_reserva INT NOT NULL,
    CONSTRAINT fk_horarios_apertura_reserva FOREIGN KEY (id_reserva) REFERENCES
    reserva(id_reserva)
);
```

```
CREATE INDEX idx_horarios_apertura_dia_semana ON horarios_apertura(dia_semana);
```

-- Tabla de Horarios de Recorridos

```
CREATE TABLE IF NOT EXISTS horarios_recorridos (
    id_recorrido INT,
    id_guia INT,
    dia DATE,
    hora TIME,
```

```
PRIMARY KEY (id_recorrido, id_guia, dia, hora),  
    CONSTRAINT fk_horarios_recorridos_recorrido FOREIGN KEY (id_recorrido) REFERENCES  
recorridos(id_recorrido),  
    CONSTRAINT fk_horarios_recorridos_guia FOREIGN KEY (id_guia) REFERENCES guias(id_guia)  
);  
  
CREATE INDEX idx_horarios_recorridos_dia_hora ON horarios_recorridos(dia, hora);
```

8. CARGA DE TABLAS

Se adjunta el script .sql dentro del github con la carga de los datos de forma manual y el borrado de tablas por si es necesario. De igual forma, aca esta una parte del script:

```
-- Borrado de datos de todas las tablas usando TRUNCATE
DELETE FROM horarios_apertura WHERE id_horario > 0;
DELETE FROM horarios_recorridos WHERE id_guia > 0;
DELETE FROM zonas_por_recorrido WHERE id_recorrido >0;
DELETE FROM recorridos WHERE id_recorrido >0;
DELETE FROM guias WHERE id_guia >0;
DELETE FROM cuidadores_por_habitats WHERE id_cuidador >0;
DELETE FROM cuidadores WHERE id_cuidador >0;
DELETE FROM personas WHERE id_persona >0;
DELETE FROM animales_por_habitats WHERE id_animal >0;
DELETE FROM habitats_por_zonas WHERE id_habitat >0;
DELETE FROM zonas WHERE id_zona >0;
DELETE FROM habitats WHERE id_habitat >0;
DELETE FROM animales WHERE id_animal >0;
DELETE FROM especies WHERE id_especie >0;
DELETE FROM sexos WHERE id_sexo >0;
DELETE FROM generos WHERE id_genero >0;
DELETE FROM reserva WHERE id_reserva >0;

-- Inserción de datos
INSERT INTO reserva (nombre, ubicacion, descripcion, fecha_fundacion, area)
VALUES ('Reserva Natural Agustina', 'Ubicación X', 'Esta es una reserva de prueba creada para el trabajo final', '2024-01-01',
1000.00);

INSERT INTO horarios_apertura (dia_semana, horario_apertura, horario_cierre,id_reserva)
VALUES
('Lunes', '08:00:00', '18:00:00', 1),
('Martes', '08:00:00', '18:00:00', 1),
('Miércoles', '08:00:00', '18:00:00', 1),
('Jueves', '08:00:00', '18:00:00', 1),
('Viernes', '08:00:00', '18:00:00', 1),
('Sábado', '08:00:00', '20:00:00', 1),
('Domingo', '08:00:00', '20:00:00', 1);

INSERT INTO generos (genero)
VALUES ('Macho'), ('Hembra');

INSERT INTO especies (nombre_comun, nombre_cientifico, descripcion, estado_conservacion)
VALUES
('Tigre', 'Panthera tigris', 'Descripción del tigre', 'En Peligro'),
('Elefante', 'Loxodonta africana', 'Descripción del elefante', 'Vulnerable'),
('León', 'Panthera leo', 'Descripción del león', 'Vulnerable'),
('Cebra', 'Equus quagga', 'Descripción de la cebra', 'Preocupación menor');
```

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

```
('Jirafa', 'Giraffa camelopardalis', 'Descripción de la jirafa', 'Vulnerable'),
('Puma', 'Puma concolor', 'Descripción del puma', 'Preocupación menor'),
('Jaguar', 'Panthera onca', 'Descripción del jaguar', 'Casi amenazado'),
('Gorila', 'Gorilla gorilla', 'Descripción del gorila', 'En Peligro'),
('Pavo real', 'Pavo cristatus', 'Descripción del pavo real', 'No evaluado'),
('Avestruz', 'Struthio camelus', 'Descripción del avestruz', 'Preocupación menor');
```

```
INSERT INTO animales (nombre, fecha_nacimiento, fecha_fallecimiento, peso, altura, id_especie, id_genero)
VALUES
```

```
('Tigre1', '2010-05-15', NULL, 200.5, 1.2, 1, 1),
('Tigre2', '2012-07-20', NULL, 180.0, 1.1, 1, 2),
('Elefante1', '2005-03-10', NULL, 3000.0, 3.0, 2, 1),
('León1', '2015-09-25', NULL, 190.0, 1.0, 3, 1),
('León2', '2016-02-12', NULL, 180.0, 0.9, 3, 2),
('Elefante2', '2010-12-01', NULL, 2800.0, 2.8, 2, 2),
('Tigre3', '2011-08-30', NULL, 190.0, 1.0, 1, 1),
('Cebra1', '2013-10-20', NULL, 200.0, 1.5, 4, 1),
('Cebra2', '2014-01-15', NULL, 190.0, 1.4, 4, 2),
('Jirafa1', '2010-07-01', NULL, 1500.0, 5.5, 5, 1),
('Jirafa2', '2012-09-05', NULL, 1400.0, 5.3, 5, 2),
('Puma1', '2015-04-10', NULL, 70.0, 0.8, 6, 1),
('Puma2', '2016-06-20', NULL, 65.0, 0.7, 6, 2),
('Jaguar1', '2014-03-15', NULL, 90.0, 0.9, 7, 1),
('Jaguar2', '2015-05-25', NULL, 85.0, 0.8, 7, 2),
('Gorila1', '2008-12-12', NULL, 180.0, 1.7, 8, 1),
('Gorila2', '2010-02-18', NULL, 170.0, 1.6, 8, 2),
('PavoReal1', '2019-06-05', NULL, 5.0, 0.4, 9, 1),
('PavoReal2', '2020-08-10', NULL, 4.5, 0.4, 9, 2),
('Avestruz1', '2017-04-01', NULL, 100.0, 2.0, 10, 1),
('Avestruz2', '2018-05-15', NULL, 95.0, 1.9, 10, 2);
```

```
INSERT INTO habitats (nombre, descripcion, tipo, superficie)
```

```
VALUES
```

```
('Habitat Tigres', 'Habitat para tigres', 'Jaula', 100.00),
('Habitat Leones', 'Habitat para leones', 'Jaula', 120.00),
('Habitat Avestruces', 'Habitat para avestruces', 'Jaula', 80.00),
('Habitat Pumas', 'Habitat para pumas', 'Jaula', 90.00),
('Habitat Jirafas', 'Habitat para jirafas', 'Jaula', 200.00);
```

```
INSERT INTO zonas (nombre, descripcion)
```

```
VALUES
```

```
('Zona Africana', 'Zona de animales africanos'),
('Zona Asiática', 'Zona de animales asiáticos'),
('Zona Americana', 'Zona de animales americanos');
```

```
INSERT INTO habitats_por_zonas (id_habitat, id_zona)
```

```
VALUES
```

```
(1, 1),
(2, 1),
(3, 2),
```



```
(4, 3),
(5, 1);
```

```
INSERT INTO animales_por_habitats (id_animal, id_habitat, fecha_ingreso)
VALUES
```

```
(1, 1, '2014-01-01'),
(2, 1, '2015-03-01'),
(3, 2, '2016-06-15'),
(4, 2, '2017-08-01'),
(5, 3, '2018-10-15'),
(6, 3, '2019-12-01'),
(7, 1, '2020-02-15'),
(8, 1, '2021-04-01'),
(9, 4, '2014-05-01'),
(10, 4, '2015-07-01'),
(11, 5, '2016-08-01'),
(12, 5, '2017-10-01'),
(13, 5, '2018-11-01'),
(14, 5, '2019-12-01'),
(15, 5, '2020-01-01'),
(16, 5, '2021-02-01'),
(17, 5, '2022-03-01'),
(18, 5, '2023-04-01'),
(19, 5, '2024-05-01'),
(20, 5, '2025-06-01');
```

```
INSERT INTO recorridos (nombre, duracion_estimada, descripcion)
VALUES
```

```
('Safari Africano', 120, 'Recorrido en vehículo por la zona africana'),
('Paseo por la Sabana', 90, 'Caminata entre jirafas y elefantes'),
('Sendero de las Aves', 60, 'Visita guiada por el aviario'),
('Exploración de Primates', 75, 'Observación de primates en su hábitat');
```

```
INSERT INTO zonas_por_recorrido (id_recorrido, id_zona, orden)
VALUES
```

```
(1, 1, 1),
(1, 2, 2),
(2, 1, 1),
(2, 2, 2),
(3, 3, 1),
(4, 1, 1),
(4, 3, 2);
```

```
INSERT INTO sexos (sexo)
```

```
VALUES ('Masculino'), ('Femenino'), ('Otro');
```

La tabla “personas” se cargó importando un archivo de formato csv.

Para comenzar se generó un conjunto de datos aleatorios utilizando: <https://www.mockaroo.com/>. Adjunto imagen de cómo se utilizó la aplicación web:

The screenshot shows the Mockaroo web application interface. It features a table with columns: Field Name, Type, and Options. The fields are configured as follows:

Field Name	Type	Options
id	Row Number	blank: 0 %
nombre	First Name	blank: 0 %
apellido	Last Name	blank: 0 %
fecha_nacimiento	Datetime	01/01/1950 to 01/01/2000 format: yyyy-mm-dd blank: 0 %
direccion	Street Address	blank: 0 %
telefono	Phone	format: #####-#### blank: 0 %
email	Email Address	blank: 0 %
dni	Number	min: 40000000 max: 49999999 decimals: 0 blank: 0 %
id_sexo	Number	min: 1 max: 2 decimals: 0 blank: 0 %

At the bottom, there are controls for generating data: # Rows: 1000, Format: CSV, Line Ending: Unix (LF), Include: header, BOM. Buttons include GENERATE DATA, PREVIEW, SAVE AS..., DERIVE FROM EXAMPLE..., and MORE.

A partir del csv generado (adjunto .csv en el github) se realizó la importación en el MySQL, previo validar que no hubiera datos cargados en la tabla.

Paso a paso:

1- Seleccionar el archivo “MOCK_PERSONAS.csv”:

The screenshot shows the 'Table Data Import' window. It has a title bar with a minus, maximize, and close button. The main content area has the heading 'Select File to Import'. Below this, it says 'Table Data Import allows you to easily import CSV, JSON datafiles. You can also create destination table on the fly.' There is a text input field for 'File Path:' containing 'D:\Downloads\MOCK_PERSONAS.csv' and a 'Browse...' button next to it.

2- Seleccionar el destino de los datos:

Select Destination

Select destination table and additional options.

☒ Use existing table: reservas_naturales.personas

☐ Create new table: reservas_naturales , MOCK_PERSONAS

☐ Truncate table before import

3- Configurar la importación, matcheando las columnas de la tabla con las del csv correspondientes. No marcar ID ya que es incremental esa columna:

Configure Import Settings

Detected file format: csv

Encoding: utf-8

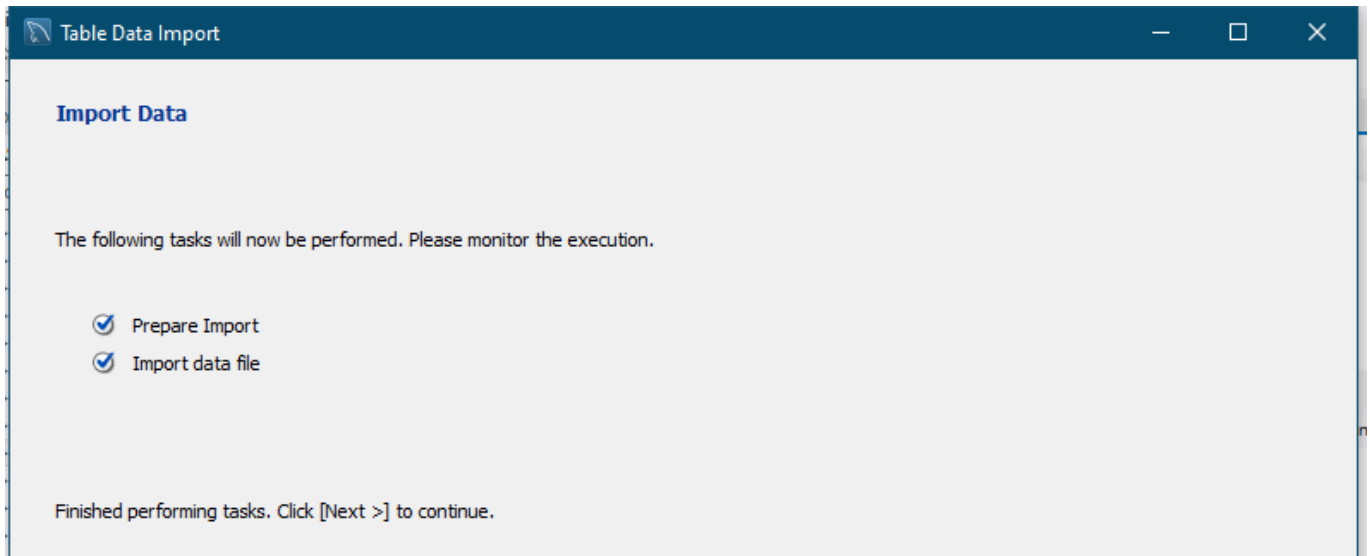
Columns:

Source Column	Dest Column
<input type="checkbox"/> id	id_persona
<input checked="" type="checkbox"/> nombre	nombre
<input checked="" type="checkbox"/> apellido	apellido
<input checked="" type="checkbox"/> fecha_nacimiento	fecha_nacimiento
<input checked="" type="checkbox"/> direccion	direccion
<input checked="" type="checkbox"/> telefono	telefono

id	nombre	apellido	fecha_naci...	direccion	telefono	email	dni	id_sexo
1	Benedict	Duffit	22-05-1996	9758 Fairvi...	4541255152	bduffit0@q...	41234042	1
2	Eunice	Mabb	18-04-1956	532 Lake Vi...	4207768718	emabb1@h...	48210601	2
3	Kathy	Itzhaiek	04-12-1989	00602 Don...	8273963109	kitzhaiek2...	42498846	2
4	Mandel	Leechman	21-11-1993	513 Transp...	2283106336	mleechman...	48678455	2
5	Brnaba	Attow	23-09-1970	3744 Melro...	9313165300	battow4@b...	40104471	1

< Back Next > Cancel

4- Continuar con Next:



5- Ejecutamos



Aclaración: En el script entregado de carga de datos, se incluyó el script de inserción de los datos (todos los datos insertados mediante INSERT) que se generaron mediante el csv, para que se pueda ejecutar todo el script de manera más fácil.

Luego se cargaron las tablas restantes que necesitaban de la tabla personas:

```
INSERT INTO cuidadores (id_persona, fecha_ingreso, especialidad)
VALUES
(1, '2010-01-01', 'Felinos'),
(2, '2012-03-15', 'Grandes herbívoros'),
(3, '2015-06-01', 'Aves'),
(4, '2018-09-01', 'Primates');
```

```
INSERT INTO cuidadores_por_habitats (id_cuidador, id_habitat)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4);
```

```
INSERT INTO guias (id_persona, fecha_ingreso, especialidad)
VALUES
```

```
(1, '2010-01-01', 'Safari africano'),
(2, '2012-03-15', 'Jirafas y elefantes'),
(3, '2015-06-01', 'Aves exóticas'),
(4, '2018-09-01', 'Primates americanos');
```

```
INSERT INTO horarios_recorridos (id_recorrido, id_guia, dia, hora)
VALUES
```

```
(1, 1, '2024-01-01', '10:00:00'),
(1, 1, '2024-01-01', '14:00:00'),
(1, 2, '2024-01-01', '12:00:00'),
(1, 2, '2024-01-01', '16:00:00'),
(2, 3, '2024-01-01', '09:00:00'),
(2, 3, '2024-01-01', '11:00:00'),
(2, 4, '2024-01-01', '13:00:00'),
(2, 4, '2024-01-01', '15:00:00'),
(3, 1, '2024-01-01', '11:00:00'),
(3, 1, '2024-01-01', '13:00:00'),
(3, 2, '2024-01-01', '10:00:00'),
(3, 2, '2024-01-01', '12:00:00'),
(4, 3, '2024-01-01', '10:30:00'),
(4, 3, '2024-01-01', '12:00:00'),
(4, 4, '2024-01-01', '11:00:00'),
(4, 4, '2024-01-01', '13:00:00');
```

9. VISTAS

Se adjunta script .sql de creación de las vistas en el github.

1. Vista: VT_animales_x_habitats_x_Zonas

Descripción: Esta vista VT_animales_x_habitats_x_Zonas relaciona información detallada de animales con sus respectivos hábitats y zonas geográficas. Su propósito principal es proporcionar una visión integral de la distribución de los animales vivos dentro de la reserva natural, permitiendo identificar rápidamente en qué hábitat y zona se encuentra cada animal.

Tablas que la componen:

- animales
- especies
- animales_por_habitats
- habitats
- habitats_por_zonas
- zonas

Propósito en el Proyecto: Esta vista es esencial para el seguimiento y la gestión de los animales en el proyecto, asegurando que cada uno esté correctamente ubicado y categorizado dentro de su hábitat y zona respectiva. Es útil para tareas como la planificación de recorridos o la gestión del espacio de los animales.

```
1 • SELECT * FROM reservas_naturales.vt_animales_x_habitats_x_zonas;
```

id_animal	animal	nombre comun especie	nombre cientifico especie	descripcion especie	habitat	descripcion habitat	zona
1	Tigre1	Tigre	Panthera tigris	Descripción del tigre	Habitat Tigres	Habitat para tigres	Zona Africana
2	Tigre2	Tigre	Panthera tigris	Descripción del tigre	Habitat Tigres	Habitat para tigres	Zona Africana
7	Tigre3	Tigre	Panthera tigris	Descripción del tigre	Habitat Tigres	Habitat para tigres	Zona Africana
8	Cebra1	Cebra	Equus quagga	Descripción de la cebra	Habitat Tigres	Habitat para tigres	Zona Africana
3	Elefante1	Elefante	Loxodonta africana	Descripción del elefante	Habitat Leones	Habitat para leones	Zona Africana
4	León1	León	Panthera leo	Descripción del león	Habitat Leones	Habitat para leones	Zona Africana
11	Jirafa2	Jirafa	Giraffa camelopardalis	Descripción de la jirafa	Habitat Jirafas	Habitat para jirafas	Zona Africana
12	Puma1	Puma	Puma concolor	Descripción del puma	Habitat Jirafas	Habitat para jirafas	Zona Africana
13	Puma2	Puma	Puma concolor	Descripción del puma	Habitat Jirafas	Habitat para jirafas	Zona Africana
14	Jaguar1	Jaguar	Panthera onca	Descripción del jaguar	Habitat Jirafas	Habitat para jirafas	Zona Africana
15	Jaguar2	Jaguar	Panthera onca	Descripción del jaguar	Habitat Jirafas	Habitat para jirafas	Zona Africana
16	Gorila1	Gorila	Gorilla gorilla	Descripción del gorila	Habitat Jirafas	Habitat para jirafas	Zona Africana
17	Gorila2	Gorila	Gorilla gorilla	Descripción del gorila	Habitat Jirafas	Habitat para jirafas	Zona Africana
18	PavoReal1	Pavo real	Pavo cristatus	Descripción del pavo real	Habitat Jirafas	Habitat para jirafas	Zona Africana
19	PavoReal2	Pavo real	Pavo cristatus	Descripción del pavo real	Habitat Jirafas	Habitat para jirafas	Zona Africana
20	Avestruz1	Avestruz	Struthio camelus	Descripción del avestruz	Habitat Jirafas	Habitat para jirafas	Zona Africana
5	León2	León	Panthera leo	Descripción del león	Habitat Avest...	Habitat para avest...	Zona Asiática
6	Elefante2	Elefante	Loxodonta africana	Descripción del elefante	Habitat Avest...	Habitat para avest...	Zona Asiática

2. Vista: VT_zonas_animales

Descripción: La vista VT_zonas_animales ofrece una visión consolidada de las zonas geográficas, mostrando la cantidad de animales vivos presentes en cada una y listando las especies que se encuentran en ellas. Es una herramienta clave para el análisis de biodiversidad dentro de las zonas y para evaluar la concentración de animales.

Tablas que la componen:

- zonas
- habitats_por_zonas
- habitats
- animales_por_habitats
- animales.
- especies

Propósito en el Proyecto: Esta vista permite obtener un resumen claro y rápido de la distribución de animales por zona. Es especialmente útil para la gestión de recursos y para garantizar que las zonas no estén sobrepobladas, ayudando en la toma de decisiones estratégicas.

```
1 • SELECT * FROM reservas_naturales.vt_zonas_animales;
```

<					
Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	id_zona	zona	descripcion	cantidad de animales	especies en zona
	1	Zona Africana	Zona de animales africanos	16	Avestruz, Cebra, Elefante, Gorila, Jaguar, Jirafa, León, Pavo real, Puma, Tigre
	2	Zona Asiática	Zona de animales asiáticos	2	Elefante, León
▶	3	Zona Americana	Zona de animales americanos	2	Cebra, Jirafa

3. Vista: vt_personas_guias_recorridos_horarios

Descripción: La vista vt_personas_guias_recorridos_horarios reúne información sobre los guías del proyecto, detallando sus datos personales, especialidades, recorridos asignados y horarios en los que llevan a cabo los recorridos. Esto facilita la planificación y coordinación de actividades guiadas.

Tablas que la componen:

- personas
- guias
- horarios_recorridos
- recorridos
- cuidadores

Propósito en el Proyecto: Esta vista es necesaria para la organización y programación de las actividades guiadas, permitiendo una gestión eficiente de los recursos humanos y asegurando que los recorridos se realicen en los horarios estipulados. Es fundamental para el buen funcionamiento de las actividades de guía en la reserva.

```
1 • SELECT * FROM reservas_naturales.vt_personas_guias_recorridos_horarios;
```

Result Grid											
Filter Rows:											
Export: Wrap Cell Content:											
id_persona	nombre_persona	apellido_persona	fecha_nacimiento	id_guia	fecha_ingreso_guia	especialidad_guia	id_recorrido	nombre_recorrido	duracion	descripcion_recorrido	hor
3	Teddie	Fulford	1975-10-30	3	2015-06-01	Aves exóticas	2	Paseo por la Sabana	90 min...	Caminata entre jirafas y elefantes	202
3	Teddie	Fulford	1975-10-30	3	2015-06-01	Aves exóticas	4	Exploración de Primates	75 min...	Observación de primates en su hábitat	202
4	Rahel	Mowat	1958-05-30	4	2018-09-01	Primates americanos	2	Paseo por la Sabana	90 min...	Caminata entre jirafas y elefantes	202
4	Rahel	Mowat	1958-05-30	4	2018-09-01	Primates americanos	4	Exploración de Primates	75 min...	Observación de primates en su hábitat	202

10. FUNCIONES Y PROCEDIMIENTOS

Adjunto script .sql de la creación de funciones y procedimientos. A su vez se adjunta script .sql para poder probar el funcionamiento de los mismos.

FUNCIONES

1. Función: calcular_indice_biodiversidad

Descripción: La función calcular_indice_biodiversidad calcula el índice de biodiversidad de un hábitat específico, que representa la diversidad relativa de especies en comparación con la cantidad de animales presentes.

Tablas utilizadas:

1. animales
2. animales_por_habitats

Result Grid	
	indice_biodiversidad
▶	50.00

2. Función: recorrido_mayor_duracion_promedio

Descripción: La función recorrido_mayor_duracion_promedio identifica el recorrido que tiene la mayor duración promedio y devuelve su nombre.

Tablas utilizadas:

1. recorridos
2. horarios_recorridos

	recorrido_mayor_duracion_promedio
▶	Safari Africano

PROCEDIMIENTOS

3. Procedimiento: analisis_distribucion_animales_zona_estado_conservacion

Descripción: Este procedimiento analiza la distribución de animales en diferentes zonas, desglosado por su estado de conservación.

Tablas utilizadas:

1. zonas
2. habitats_por_zonas
3. habitats
4. animales_por_habitats
5. animales
6. especies

Result Grid Filter Rows: Export: Wrap Cell Content:							
	id_zona	nombre_zona	total_animales	animales_en_peligro	animales_vulnerables	animales_preocupacion_menor	animales_no_evaluado
▶	1	Zona Africana	16	5	3	4	2
	3	Zona Americana	2	0	1	1	0
	2	Zona Asiática	2	0	2	0	0

4. Procedimiento: informe_detallado_recorridos_guia_animales

Descripción: Este procedimiento genera un informe detallado de los recorridos guiados, incluyendo información sobre guías, zonas, hábitats y animales observados.

Tablas utilizadas:

1. recorridos
2. horarios_recorridos
3. guias
4. personas
5. zonas_por_recorrido
6. zonas
7. habitats_por_zonas
8. habitats
9. animales_por_habitats
10. animales
11. especies

Result Grid Filter Rows: Export: Wrap Cell Content:							
	id_recorrido	nombre_recorrido	duracion	descripcion_recorrido	nombre_guia	especialidad_guia	cant
▶	1	Safari Africano	120	Recorrido en vehículo por la zona africana	Vanya Crosse	Safari africano	36
	1	Safari Africano	120	Recorrido en vehículo por la zona africana	Alfred Bohills	Jirafas y elefantes	36
	2	Paseo por la Sabana	90	Caminata entre jirafas y elefantes	Teddie Fulford	Aves exóticas	36
	2	Paseo por la Sabana	90	Caminata entre jirafas y elefantes	Rahel Mowat	Primates americanos	36