

# Practica 1: Bases De Datos Relacionales



**Alejandro Silva Rodríguez**

**Marta Cuevas Rodríguez**

*Almacenes De Datos*  
Universidad de Málaga

Septiembre 2024

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Parques Naturales</b>	<b>2</b>
2.1. Requisitos De Datos . . . . .	2
2.2. Diseño Lógico . . . . .	3
2.3. Diseño Entidad Relación . . . . .	4
2.4. Implementación . . . . .	4
2.5. Consultas . . . . .	6
<b>3. Liga De Futbol</b>	<b>7</b>
<b>4. Acceso al Repositorio</b>	<b>7</b>

## 1. Introducción

En esta práctica, se diseñará y creará una base de datos para gestionar información sobre parques naturales en Andalucía y la liga española de fútbol de primera división. El objetivo es elaborar un modelo entidad-relación y un modelo relacional que muestren las características y relaciones de las entidades involucradas, utilizando herramientas de diseño de bases de datos.

Además, se generará el DDL (Data Definition Language) para SQL Server, que permitirá crear la base de datos y las tablas necesarias. Se incluirán datos sobre los parques naturales, su gestión y las especies que los habitan, así como información sobre los equipos de fútbol, sus jugadores y los partidos.

Este ejercicio ayudará a aplicar los conceptos de bases de datos y ofrecerá una experiencia práctica en el uso de herramientas y lenguajes de consulta.

## 2. Parques Naturales

En esta sección se explicará como se diseñó e implementó la base de datos de parques naturales y las consultas sobre la misma.

### 2.1. Requisitos De Datos

La Junta de Andalucía desea mantener la información sobre los parques naturales que hay en su comunidad autónoma. En particular sería necesario conocer el nombre del parque (que es único), su teléfono, dirección administrativa, una dirección web, un correo electrónico, su fecha de declaración como parque natural, la extensión (en hectáreas) de cada zona protegida, las especies animales que contiene, la población estimada de cada una de ellas y la dirección gestora del parque. Esta dirección gestora del parque está coordinada por un presidente y un número no determinado de consejeros. De todos estos miembros de la dirección gestora se desea conocer el DNI, nombre, fecha de nacimiento, dirección y teléfono de cada uno de ellos. Cada persona puede ser a lo sumo consejero en un parque y presidente de otro. De las especies guardamos su nombre científico y el común (ambos son únicos), el número de años de vida media.

Con objeto de poder determinar el estado de salud del parque necesitamos información sobre la interacción del hombre con el entorno. Para ello se almacenan datos sobre los municipios donde está ubicado el parque: número de municipios que abarca, nombre de cada uno de ellos, enlace a su web, fichero con la foto de su escudo, partido que gobierna en la alcaldía, número de habitantes y gasto de agua medio por habitante.

Las especies tienen una extensión (en hectáreas) necesaria para desarrollarse en libertad, dato que aparece en los estudios generales sobre cada especie. Sin embargo, el dato de si la especie está superpoblada en cada parque se guardará explícitamente, porque puede depender de factores como si el parque es montañoso, si tiene acuíferos, etc y por tanto precisa de la opinión última de un experto. Tenga en cuenta que cada municipio abarca a lo sumo un sólo parque.

## 2.2. Diseño Lógico

Para crear el diseño lógico de la base de datos en Oracle Datamodeler [2] empezaremos representando las figuras más relevantes como entidades, entre ellas encontramos a especie animal, municipio, parque natural y persona (figura 1). Cada entidad tendrá atributos que solo pertenecen a ella y no se relacionan con ninguna entidad más excepto del atributo numero de municipios que abarca de un parque natural, tendrá que ir actualizandose mediante un trigger cada vez que se le añade o elimina una relación municipio. También tendremos una entidad intermedia entre especie animal y parque natural que almacenará la relación entre estas y los atributos extensión, población estimada y is\_superpoblado (booleano).

Por último, tenemos relaciones entre parque natural y municipio y dos entre parque natural y persona. Una relación 1:N representará los consejeros del parque natural y otra 1:1 representará el presidente del parque.

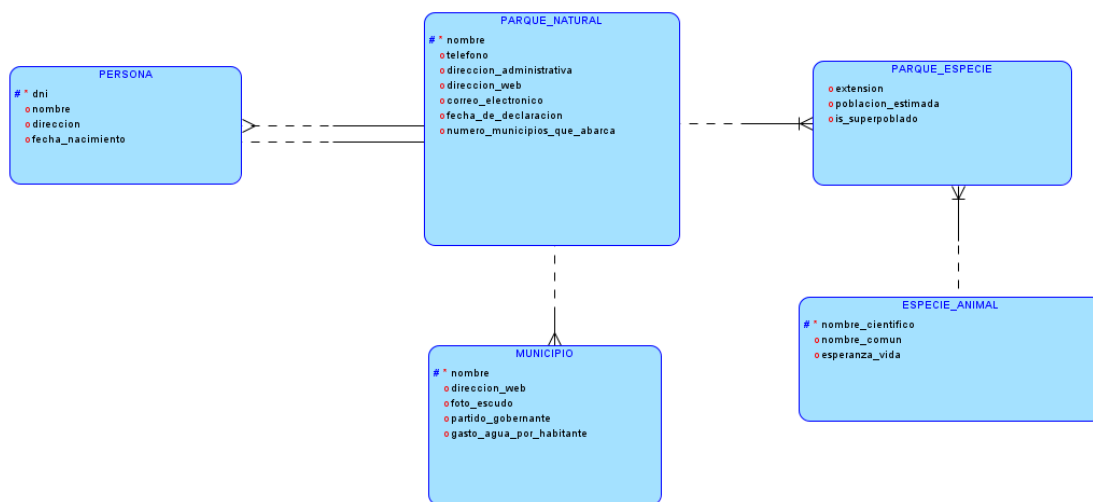


Figura 1: Diagrama Lógico De Parques Naturales

## 2.3. Diseño Entidad Relación

En el diseño entidad relación tenemos que tener en cuenta que las claves y restricciones están donde se necesitan para poder hacer consultas en la base de datos. Observamos (figura 2) que parque natural tiene una clave foránea dni que representa al unico presidente de esta entidad, mientras que persona tiene una clave foránea nombre de parque natural que relaciona a los consejeros con esta. Municipio obtiene la clave foránea de parque natural debido a su particion en la parte N de la relación. La entidad intermedia de especie animal y parque natural obtiene sus dos claves forneas que pasan a ser su clave principal.

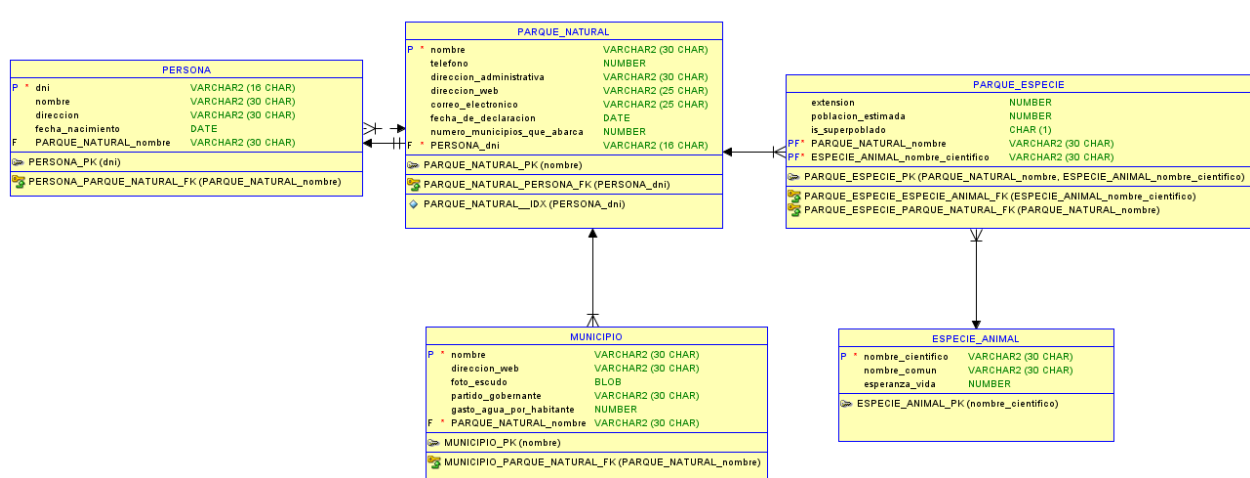


Figura 2: Diagrama Relacional De Parques Naturales

## 2.4. Implementación

En la implementación de la base de datos generaremos el ddl (data definition language) de la base de datos e insertaremos tuplas en la misma. En el listing 1 encontramos una pequeña parte del mismo donde se crea la tabla de especie animal. Esto se ejecutará en Microsoft SQL Server [1] donde se alojará la base de datos.

```

1  CREATE TABLE ESPECIE_ANIMAL
2
3  (
4    nombre_cientifico VARCHAR (30) NOT NULL ,
5    nombre_comun VARCHAR (30) ,
6    esperanza_vida NUMERIC (28)
7  )
8  GO
9
10 ALTER TABLE ESPECIE_ANIMAL ADD CONSTRAINT ESPECIE_ANIMAL_PK PRIMARY KEY
11     CLUSTERED (nombre_cientifico)
12 WITH (
13     ALLOW_PAGE_LOCKS = ON ,
14     ALLOW_ROW_LOCKS = ON )
15 GO
  
```

Listing 1: Definicion De Datos De Parques Naturales

En el listing 2 se observan ejemplos de las inserciones que se hacen sobre la base de datos y el trigger que añadimos para la actualización del número de municipios. Es crucial seguir un orden logico a la hora de la inserción ya que algunas entidades dependen de otras.

```

1  -- Insertar especies animales
2  INSERT INTO ESPECIE_ANIMAL (nombre_cientifico, nombre_comun,
3      esperanza_vida) VALUES
4      ('Ursus arctos', 'Oso pardo', 25),
5  -- Insertar personas (presidentes) - un presidente por parque natural
6  INSERT INTO PERSONA (dni, nombre, direccion, fecha_nacimiento) VALUES
7      ('34567890C', 'Luis Torres', 'Calle Larga 789', '1985-02-20'), -- Cabo de
8      Gata
9  -- Insertar parques naturales
10 INSERT INTO PARQUE_NATURAL (nombre, telefono, direccion_administrativa,
11     direccion_web, correo_electronico, fecha_de_declaracion,
12     numero_municipios_que_abarca, PERSONA_dni) VALUES
13     ('Sierra Nevada', 123456789, 'Granada', 'www.sierranevada.com',
14     'Picos de Europa', 321321321, 'Asturias', 'www.picosdeeuropa.com', '
15     contacto@picos.com', '1999-10-05', 0, '45678901D');
16 -- Insertar consejeros
17 INSERT INTO PERSONA (dni, nombre, direccion, fecha_nacimiento,
18     PARQUE_NATURAL_nombre) VALUES
19     ('23456789K', 'Valeria Torres', 'Avenida Costa 234', '1988-09-17', 'Cabo
20     de Gata'),
21 go
22 -- Actualizar el numero de municipios del parque al insertar un municipio
23 al que pertenece
24 CREATE TRIGGER actualizar_numero_municipios_insert
25 ON MUNICIPIO
26 AFTER INSERT
27 AS
28 BEGIN
29     -- Sumar 1 a cada parque natural por cada municipio insertado
30     UPDATE PN
31     SET PN.numero_municipios_que_abarca = PN.numero_municipios_que_abarca + (
32     SELECT COUNT(*)
33     FROM inserted I
34     WHERE I.PARQUE_NATURAL_nombre = PN.nombre
35     )
36     FROM PARQUE_NATURAL PN
37     WHERE EXISTS (
38     SELECT 1 FROM inserted I WHERE I.PARQUE_NATURAL_nombre = PN.nombre
39     );
40 END;
41 GO
42 -- Insertar municipios
43 INSERT INTO MUNICIPIO (nombre, direccion_web, foto_escudo,
44     partido_gobernante, gasto_agua_por_habitante, PARQUE_NATURAL_nombre)
45     VALUES
46     ('Granada', 'www.granada.es', NULL, 'PSOE', 50.25, 'Sierra Nevada'),
47 -- Insertar especies en parques
48 INSERT INTO PARQUE_ESPECIE (extension, poblacion_estimada, is_superpoblado
49     , PARQUE_NATURAL_nombre, ESPECIE_ANIMAL_nombre_cientifico) VALUES
50     (10000, 200, 0, 'Sierra Nevada', 'Ursus arctos'),

```

Listing 2: Carga de datos

## 2.5. Consultas

Para probar el correcto funcionamiento de la base de datos realizaremos consultas (listing 3) que involucren varias tablas.

```
1
2      -- Consulta 1: Seleccionar los animales que se pueden encontrar en Sierra
      Nevada
3      SELECT e.nombre_cientifico, e.nombre_comun, p.nombre
4      FROM PARQUE_NATURAL p, PARQUE_ESPECIE pe, ESPECIE_ANIMAL e
5      WHERE e.nombre_cientifico=pe.ESPECIE_ANIMAL_nombre_cientifico and pe.
      PARQUE_NATURAL_nombre=p.nombre and p.nombre='Sierra Nevada';
6
7      -- Consulta 2: Seleccionar el nombre de los consejeros de Do ana
8
9      SELECT p.nombre, c.nombre
10     FROM PARQUE_NATURAL p, PERSONA c
11     WHERE c.PARQUE_NATURAL_nombre=p.nombre and p.nombre='Do ana';
12
13     -- Consulta 3: Seleccionar el nombre del presidente de Do ana
14
15     SELECT p.nombre, c.nombre
16     FROM PARQUE_NATURAL p, PERSONA c
17     WHERE c.dni=p.PERSONA_dni and p.nombre='Do ana';
18
19     -- Consulta 4: Seleccionar el nombre de los municipios que abarca do ana
      junto al numero de municipios que abarca
20
21     SELECT m.nombre, p.nombre, p.numero_municipios_que_abarca
22     FROM PARQUE_NATURAL p, MUNICIPIO m
23     WHERE p.nombre=m.PARQUE_NATURAL_nombre and p.nombre='Do ana';
24
25     -- Consulta 5: Seleccionar el nombre de los presidentes de parques en los
      que se encuentra algun tipo de oso
26
27     SELECT e.nombre_comun, presi.nombre, p.nombre
28     FROM PERSONA presi, ESPECIE_ANIMAL e, PARQUE_NATURAL p, PARQUE_ESPECIE pe
29     WHERE presi.dni=p.PERSONA_dni and p.nombre=pe.PARQUE_NATURAL_nombre and pe.
      ESPECIE_ANIMAL_nombre_cientifico=e.nombre_cientifico and e.
      nombre_comun LIKE 'Oso%';
```

Listing 3: Consultas Sobre Parques Naturales

### 3. Liga De Futbol

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### 4. Acceso al Repositorio

Toda la información adicional, incluyendo el código fuente y la documentación completa de este proyecto, está disponible en el repositorio de GitHub [3].

## Referencias

- [1] Microsoft Corporation. *Microsoft SQL Server 2022 Documentation*. Microsoft Corporation, Redmond, Washington, 2022. Versión SQL Server 2022.
- [2] Oracle Corporation. *Oracle SQL Developer Data Modeler User's Guide*. Oracle Corporation, Redwood Shores, California, 2024. Versión 23.2.
- [3] Alex Silva. `Practical_almacenes_de_datos`. [https://github.com/AlexSilvaa9/Practical\\_almacenes\\_de\\_datos](https://github.com/AlexSilvaa9/Practical_almacenes_de_datos), 2024. *ltimo acceso 10 octubre 2024*.