

DOCUMENTACIÓN BASE DE DATOS ZOOLÓGICO

Nombres:

- Juan Jose Garcia Aguirre
- Luis Eduardo Molina Paniagua
- Jorge Stiven Rodriguez Garzon

¿ Está conforme con el resultado obtenido según el contexto o cree que hubiera obtenido un mejor resultado con una base de datos no relacional?

Dentro del proyecto y de acuerdo con los requisitos establecidos, el equipo está bastante satisfecho con el rendimiento de la base de datos relacional utilizada. La elección de una base de datos relacional nos permitió manejar eficazmente los datos de empleados, proveedores y animales en el sistema de inventario del zoológico. La estructura clara y organizada de los datos facilitó su gestión y contribuyó a mantener la integridad de la información. A pesar de esto, reconocemos que en ciertas situaciones una base de datos no relacional podría ofrecer ventajas adicionales. Específicamente, para gestionar grandes volúmenes de datos no estructurados o semiestructurados, como registros de comportamiento animal o datos de sensores, una base de datos no relacional podría haber sido más adecuada. Con esto concluimos que considerando las necesidades actuales del proyecto y las características de los datos involucrados, creemos que la decisión de utilizar una base de datos relacional fue apropiada. Sin embargo, estamos abiertos a explorar una base de datos no relacional.

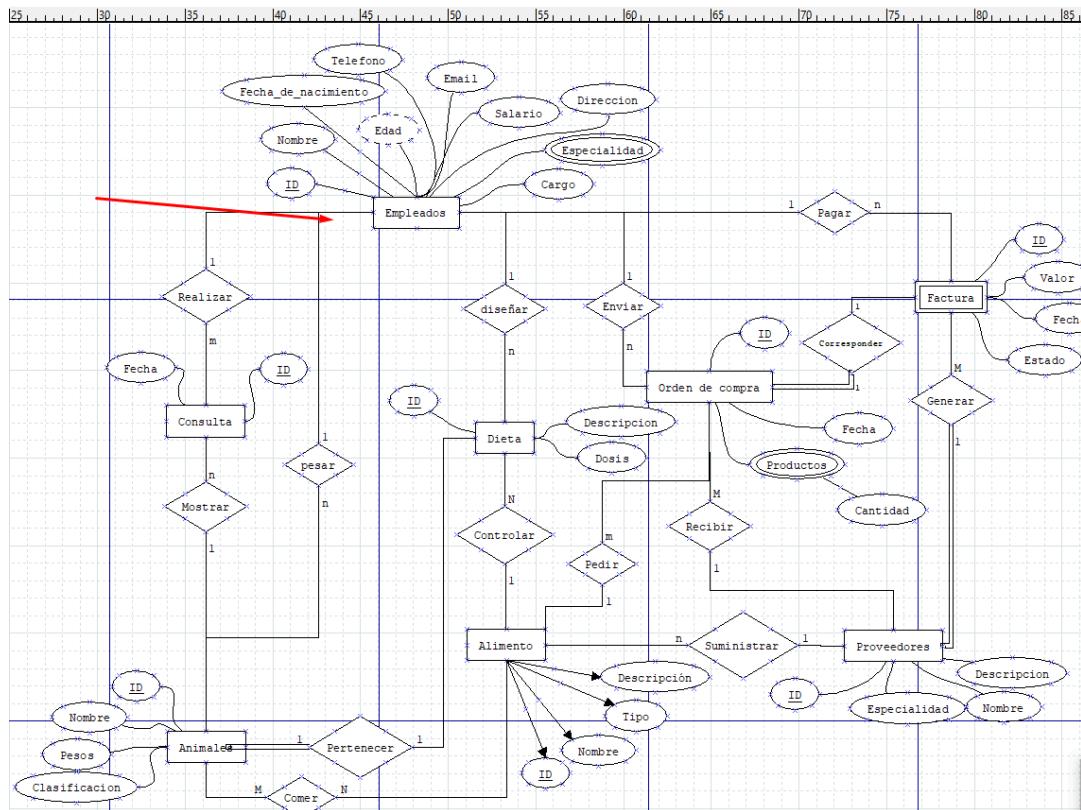
Definiciones iniciales

Para el presente trabajo tomamos como referencia iniciales.

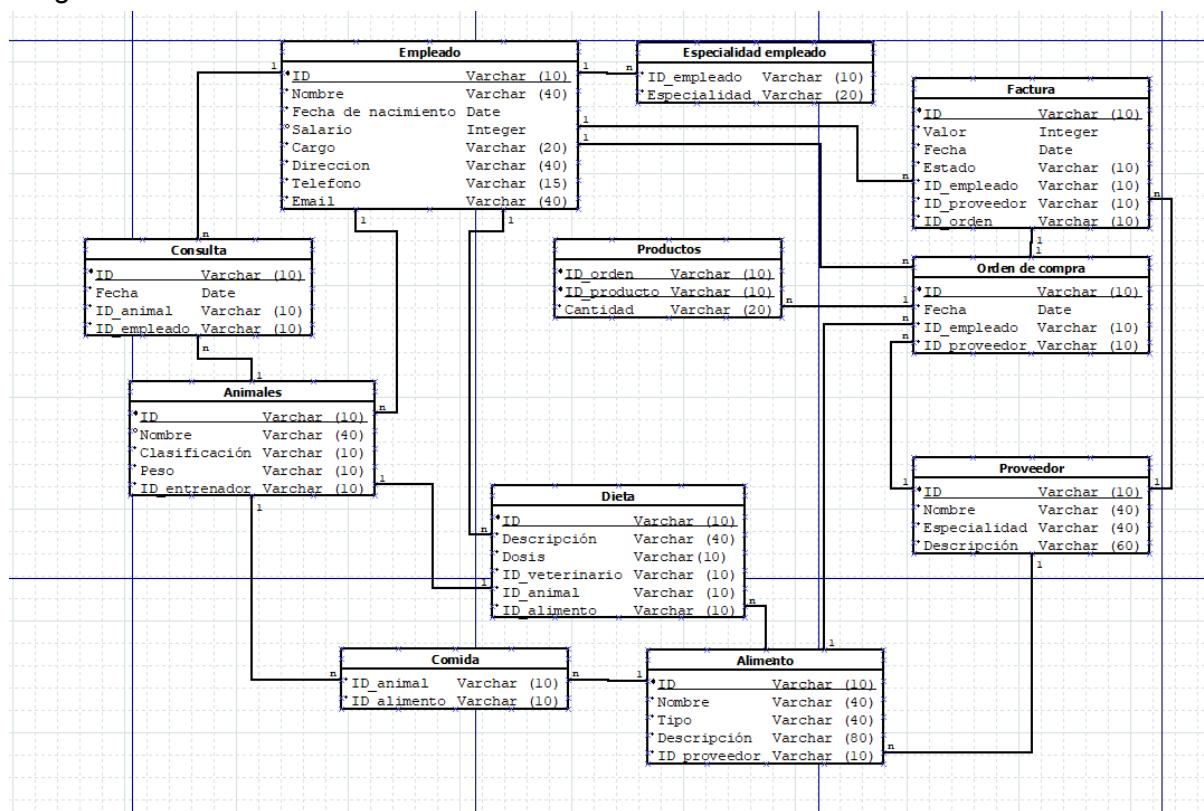
- Proyecto Reto Base de datos Taller 5 ejercicio B ZOO enlace del repositorio:
<https://github.com/Training2024SU/2024-C1-QA-BD-T05F.git>
- Página web como referencia al ejercicio: <https://www.zoologicodecali.com.co/>
- Hicimos un repositorio fork del original que próximamente haremos su respectivo PR: <https://github.com/josroga/2024-C1-QA-BD-T05F.git>
- Asesorías del Coach Juan Esteban
- Pagina de servicio rest para basarnos: <https://reqres.in/>

Después de leer indicaciones del taller iniciamos en la base del proyecto con el desarrollo del diagrama entidad relación ER.

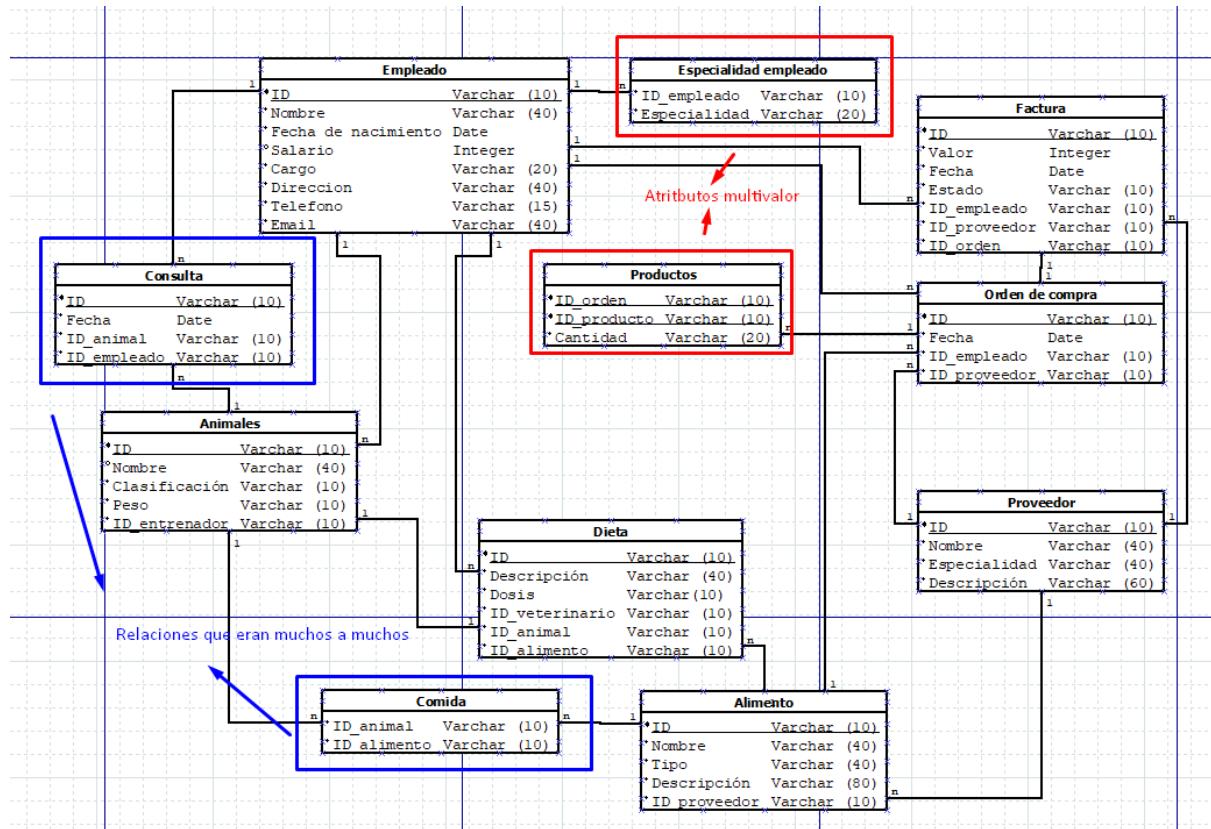
Como la base de datos se diseña enfocada al manejo de alimentos y los animales del zoológico y no para verificar otras funcionalidades del zoológico como entradas de visitantes o compras de tiquete, nuestra base de datos estará enfocada en este sentido.
No vimos necesaria la especialización o generalización ya que en la entidad Empleados, están los mismos atributos que usarán todos según su cargo



Luego de aclarar el modelo Entidad Relación se inició con la creación del Modelo Relacional



Donde algunas relaciones se convirtieron en tablas intermedias para cumplir con la reglas de la normalización y unos atributos multivaluados se agregaron su respectiva tabla.



Después las tablas se modela en SQL Workbench y se le agregan los registros a cada tabla.

Tabla Alimento

Table: alimento

Columns:

ID	Nombre	Tipo	Descripción	ID_proveedor
A1	Carne de res	Carne	Carne de res fresca y magra	P1
A1.1	Higos secos	Fruta	Higos secos para alimentar a las hojas	P1
A1.11	Arroz cocinado	Paella	Arroz cocinado como alimento para serpientes	P2
A1.12	Naranjas	Fruta	Naranjas frescas y jugosa	P2
A1.13	Ratas vivas	Rodod	Ratas vivas como alimento para aves rapaces	P2
A1.14	Bambú vivo	Planta	Bambú fresco y crudo para a perdices	P2
A1.15	Cebolla de seda	Verde	Cebolla de seda para aves rapaces	P2
A1.16	Pienso en escamas	Pienso	Pienso en escamas para peces tropicales	P3
A1.17	Carne de pollo	Carne	Carne de pollo magra y sin huesos	P3
A1.18	Semillas de girasol	Señilla	Semillas de girasol y gomina para pájaros	P3
A1.19	Agua de mar	Agua	Agua de mar para peces y tortugas	P3
A1.2	Heno de alfalfa	Forraje	Heno de alfalfa para herbívoros	P1
A2.0	Pienso granulado	Pienso	Pienso granulado para animales pequeños	P3
A3.3	Pescado blanco	Pescado	Pescado blanco fresco y sin espinas	P1
A4.1	Pulpo troceado	Verde	Masa de vegetales frescos y crujientes	P1
A4.5	Verduras frescas	Fruta	Masa de vegetales frescas y crujientes	P1
A5.6	Nectar de flores	Nectar	Nectar de diferentes flores	P1
A6.7	Insectos vivos	Insecto	Masa de insectos vivos y crujientes	P7
A8.8	Herbas aromáticas	Herba	Masa de Herbas frescas y aromáticas	P1
A8.9	Granulos de pienso	Pienso	Granulos de pienso balanceado para herbívoros	P1

Tabla Animales

The screenshot shows the MySQL Workbench interface with the 'animales' table selected in the 'schemas' tree on the left. The main window displays the table structure and data in a grid format.

Table Structure:

```
CREATE TABLE `animales` (
  `ID` int(10) unsigned NOT NULL,
  `Nombre` varchar(100) NOT NULL,
  `Clasificacion` varchar(50) NOT NULL,
  `Peso` decimal(10,2) NOT NULL,
  `ID_entrenador` int(10) unsigned NOT NULL
)
```

Table Data:

ID	Nombre	Clasificacion	Peso	ID_entrenador
A1	Leon	Mamifero	200 kg	E2
A10	Tortuga	Reptil	50 kg	E12
A11	Aguila	Ave	9 kg	E16
A12	Aguila	Ave	8 kg	E16
A13	Tiburon	pez	350 kg	E2
A14	Hipopotamo	Mamifero	1800 kg	E5
A15	Hipopotamo	Mamifero	1800 kg	E2
A16	Tiburon	Pez	400 kg	E2
A17	Oso de anteojos	Mamifero	300 kg	E10
A18	Tortuga marina	Reptil	200 kg	E12
A19	pez payaso	peces	0.1 kg	E19
A2	Leon	Mamifero	250 kg	E5
A20	pez payaso	peces	0.1 kg	E2
A21	Buho	Ave	7 kg	E10
A3	Elefante	Mamifero	5000 kg	E5
A4	Jirafa	Mamifero	1200 kg	E2
A5	Leopardo	Mamifero	60 kg	E7
A6	Elefante	Mamifero	4000 kg	E10
A7	Jirafa	Mamifero	1100 kg	E2
A8	Cocodrilo	Reptil	1000 kg	E10
A9	Serpiente	Reptil	10 kg	E12
*	NULL	NULL	NULL	NULL

Tabla comida

The screenshot shows the MySQL Workbench interface with the 'comida' table selected in the database definition tab. The table structure is as follows:

```

CREATE TABLE `comida` (
  `ID_animal` int(10) NOT NULL,
  `ID_alimento` int(10) NOT NULL
)

```

The data grid displays the following rows:

ID_animal	ID_alimento
A1	AL1
A2	AL1
A5	AL1
A8	AL1
A11	AL11
A12	AL11
A9	AL11
A19	AL16
A20	AL16
A1	AL17
A2	AL17
A5	AL17
A8	AL17
A13	AL3
A10	AL5
A14	AL5
A15	AL5

Tabla consulta

The screenshot shows the MySQL Workbench interface with the 'consulta' table selected in the database definition tab. The table structure is as follows:

```

CREATE TABLE `consulta` (
  `ID` int(10) NOT NULL,
  `Fecha` date NOT NULL,
  `ID_animal` int(10) NOT NULL,
  `ID_empleado` int(10) NOT NULL
)

```

The data grid displays the following rows:

ID	Fecha	ID_animal	ID_empleado
C01	2024-04-04	A17	E10
C02	2024-04-10	A10	E12
C03	2024-04-07	A9	E12
C04	2024-04-16	A16	E2
C05	2024-04-12	A12	E16
C06	2024-04-09	A6	E10
C07	2024-04-09	A10	E12
C08	2024-04-07	A2	E5
C09	2024-04-08	A1	E2
C10	2024-04-03	A15	E2
C11	2024-04-15	A13	E2
C12	2024-04-13	A15	E2
C13	2024-04-13	A2	E5
C14	2024-04-14	A1	E2
C15	2024-04-17	A5	E7
C16	2024-04-16	A2	E5
C17	2024-04-05	A20	E2
C18	2024-04-16	A7	E2
C19	2024-04-03	A5	E7
C20	2024-04-02	A17	E10

Tabla dieta

MySQL Workbench Screenshot showing the 'dieta' table:

ID	Descripcion	Dosis	ID_veterinario	ID_animal	ID_alimento
D1	Dieta para leones	200g	E4	A1	AL1
D10	Dieta para gorilas	500g	E14	A10	AL10
D11	Dieta para serpientes	100g	E16	A11	AL11
D12	Dieta para osos	400g	E18	A12	AL12
D13	Dieta para aves rapaces	300g	E2	A13	AL13
D14	Dieta para koalas	200g	E5	A14	AL14
D15	Dieta para tortugas	100g	E15	A15	AL15
D16	Dieta para pandas	600g	E17	A16	AL16
D17	Dieta para suricatas	150g	E18	A17	AL17
D18	Dieta para hipopótamos	800g	E20	A18	AL18
D19	Dieta para rinocerontes	900g	E2	A19	AL19
D2	Dieta para elefantes	5kg	E5	A2	AL2
D20	Dieta para peces	50g	E5	A20	AL20
D3	Dieta para pingüinos	100g	E6	A3	AL3
D4	Dieta para tigres	300g	E7	A4	AL4
D5	Dieta para jirafas	7kg	E8	A5	AL5
D6	Dieta para monos	150g	E10	A6	AL6
D7	Dieta para cocodrilos	400g	E2	A7	AL7
D8	Dieta para canguros	250g	E4	A8	AL8
D9	Dieta para delfines	1kg	E12	A9	AL9
*	NULL	NULL	NULL	NULL	NULL

Table: proveedor

Tabla Empleado

MySQL Workbench Screenshot showing the 'empleado' table:

ID	nombre	fecha_nac	salario	cargo	direccion	telefono	email
E1	Juan Pérez	1985-07-15	2500	Gerente	Calle 123, Ciudad	5551234	juan@mail.com
E10	Marcela Torres	1995-02-08	2250	Entrenador	Calle 321, Pueblo	5550123	marcela@mail.com
E11	Luisa Ramírez	1990-11-05	2600	cuidador	Carrera 10, Ciudad	5552468	luisa@mail.com
E12	Diego Sánchez	1987-07-20	2800	Entrenador	Avenida 15, Pueblo	5551357	diego@mail.com
E13	Carolina Herrera	1993-04-18	2450	Veterinaria	Calle 20, Ciudad	5553698	carolina@mail.com
E14	Roberto Martín	1985-12-30	2700	Logístico	Ruta 5, Pueblo	5558024	roberto@mail.com
E15	Valeria Gutiérrez	1988-09-15	2300	Veterinaria	Avenida 25, Ciudad	5559753	valeria@mail.com
E16	Gabriel Díaz	1992-06-12	2550	Entrenador	Calle 30, Pueblo	5557412	gabriel@mail.com
E17	Isabel Jiménez	1986-08-28	2750	Veterinaria	Avenida 35, Ciudad	5553692	isabel@mail.com
E18	Francisco Ruiz	1994-03-22	2400	Cuidador	Carrera 40, Pueblo	5558023	francisco@mail.com
E19	Camila Vargas	1991-10-17	2650	Entrenadora	Ruta 55, Ciudad	5551478	camila@mail.com
E2	Maria García	1990-02-28	2200	Entrenador	Av. Principal, Pueblo	5555678	mariaG@mail.com
E20	Alejandro Castro	1989-04-05	2300	Veterinaria	Avenida 60, Pueblo	5559632	alejandro@mail.com
E3	Carlos Ruiz	1988-11-10	2800	Logístico	Calle 456, Ciudad	5559012	carlosruiz@mail.com
E4	Laura Martínez	1987-09-03	2600	Veterinario	Av. Central, Pueblo	5553456	lmartinez@mail.com
E5	Andrés López	1993-04-20	2300	Entrenador	Calle 789, Ciudad	5557890	andres@mail.com
E6	Ana Gómez	1984-12-12	2700	Veterinario	Calle 987, Pueblo	5556789	ana@mail.com
E7	Pedro López	1991-08-25	2400	Entrenador	Av. Libertad, Ciudad	5552345	pedro@mail.com
E8	Sofía Rodríguez	1989-03-17	2900	Cuidador	Calle 654, Pueblo	5554567	sofia@mail.com
E9	Javier Fernández	1986-06-30	2550	Veterinario	Av. Principal, Ciudad	5557891	javier@mail.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: proveedor

Tabla especialidad-empleado

The screenshot shows the SQL Server Management Studio interface with the following details:

- Object Explorer:** Shows the database structure under the 'zoologico' schema, including tables like 'alimento', 'animales', 'consulta', 'dieta', 'empleado', 'especialidad_empleado', 'factura', 'orden_de_compra', and 'productos', along with views, stored procedures, and functions.
- Query Editor:** Displays the query: `SELECT * FROM zoologico.especialidad_empleado;`
- Result Grid:** Shows the data from the query:

ID_empleado	especialidad
E4	animales grandes
E4	Cirugía
E6	animales pequeños
E6	Nutrición
E6	fauna silvestre
E7	felinos
E9	fauna silvestre
E9	Cirugía
E9	Cirugía
E9	animales pequeños
E10	aves
E12	aves
E13	Aves
E15	Peces Grandes
E15	Peces pequeños
E16	Entrenador felinos
E17	animales grandes

Tabla factura

The screenshot shows the SQL Server Management Studio interface with the following details:

- Object Explorer:** Shows the database structure under the 'zoologico' schema, including tables like 'alimento', 'animales', 'consulta', 'dieta', 'empleado', 'especialidad_empleado', 'factura', 'orden_de_compra', and 'productos', along with views, stored procedures, and functions.
- Query Editor:** Displays the query: `SELECT * FROM zoologico.factura;`
- Result Grid:** Shows the data from the query:

ID	Valor	Fecha	Estado	ID_empleado	ID_proveedor	ID_orden
F1	15000	2024-04-01	Pendiente	E1	P1	OC1
F10	28000	2024-04-10	Pagada	E1	P2	OC10
F11	23000	2024-04-11	Pendiente	E1	P3	OC11
F12	26000	2024-04-12	Pagada	E1	P1	OC12
F13	32000	2024-04-13	Pendiente	E1	P1	OC13
F14	27000	2024-04-14	Pagada	E1	P2	OC14
F15	30000	2024-04-15	Pendiente	E1	P2	OC15
F16	24000	2024-04-16	Pagada	E1	P3	OC16
F17	29000	2024-04-17	Pendiente	E1	P3	OC17
F18	31000	2024-04-18	Pagada	E1	P3	OC18
F19	33000	2024-04-19	Pendiente	E1	P1	OC19
F2	20000	2024-04-02	Pagada	E1	P1	OC2
F20	34000	2024-04-20	Pagada	E1	P1	OC20
F3	30000	2024-04-03	Pendiente	E1	P1	OC3
F4	25000	2024-04-04	Pagada	E1	P1	OC4
F5	18000	2024-04-05	Pendiente	E1	P2	OC5
F6	22000	2024-04-06	Pagada	E1	P2	OC6
F7	27000	2024-04-07	Pendiente	E1	P1	OC7

Tabla orden de compra

Screenshot of MySQL Workbench showing the 'orden_de_compra' table.

Query:

```
1 • SELECT * FROM zoologico.productos;
```

Table Structure:

ID_orden	ID_alimento	Cantidad
OC1	AL1	50
OC2	AL3	30
OC3	AL2	100
OC4	AL4	80
OC5	AL7	200
OC6	AL8	40
OC7	AL12	60
OC8	AL11	150
OC9	AL6	70
OC10	AL9	90
OC11	AL5	120
OC12	AL17	110
OC13	AL17	75
OC14	AL13	200
OC15	AL19	85
OC16	AL15	60
OC17	AL1	90
OC18	AL19	110
OC19	AL9	75

Tabla Proveedor

Screenshot of MySQL Workbench showing the 'proveedor' table.

Query:

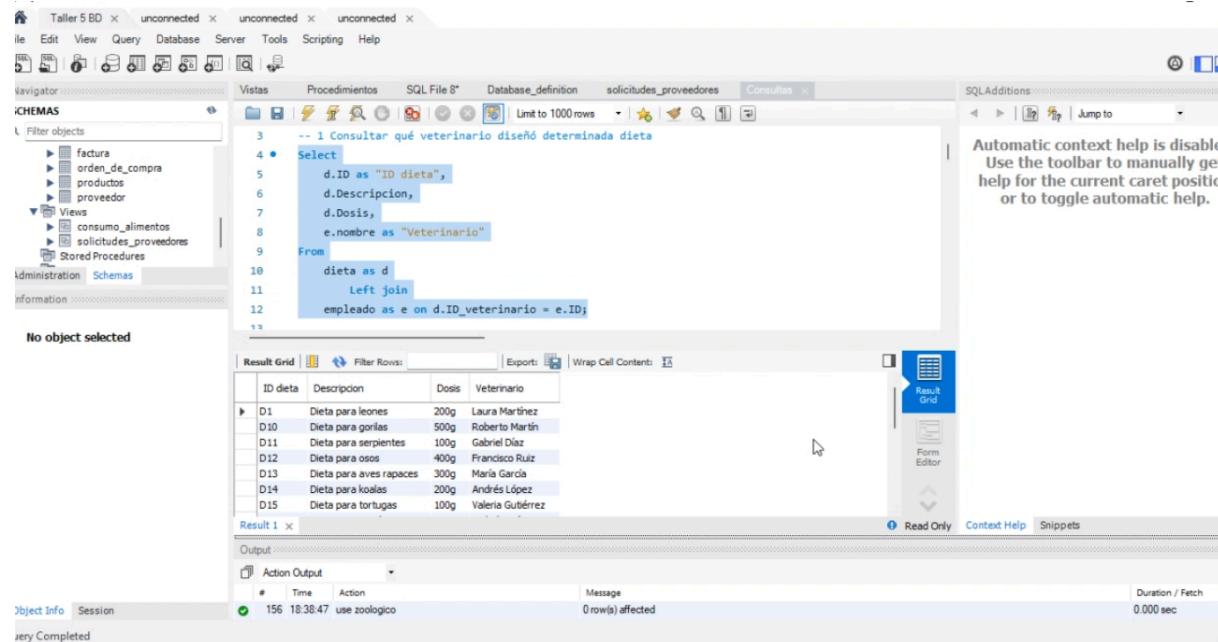
```
1 • SELECT * FROM zoologico.proveedor;
```

Table Structure:

ID	Nombre	Especialidad	Descripcion
P1	Tierragro	Mamiferos	mascotas y aves
P10	FurryFriends	Mamiferos, Aves	Alimentos y juguetes para mascotas
P11	FeehersFriends	Aves	Alimentos aves
P12	Reptileaven	Mamiferos, Peces	Reptiles y alimentos
P13	PawPrints	Mamiferos, Peces	Productos personalizados
P14	FinsAndScales	Peces	Peces tropicales y alimentos
P15	CritterCare	Mamiferos	Productos para mascotas
P16	TallWaggers	Mamiferos	Alimentos y accesorios para perros
P17	Purfect	Mamiferos, Aves	Productos para gatos y aves
P18	WhiskerWonders	Mamiferos, Aves	Juguetes para gatos y aves
P19	WildWings	Aves	Alimentos y accesorios para aves
P2	VetSupplies Ltd.	Mamiferos	Alimentos mascotas
P20	FurFeathers	Mamiferos, Peces	Productos para mascotas
P3	HabitatWorld S.A.	Peces	Alimentos peces
P4	PetPharma	Mamiferos	Medicamentos y suplementos
P5	AquaTerra	Peces	Aquarios y alimentos para peces
P6	PlusPets	Mamiferos	Juguetes y accesorios para mascotas
P7	GreenGrass	Mamiferos, Aves	Plantas y suministros de jardin
P8	BirdieSupplies	Aves	Alimentos para aves
P9	ExoticPets	Mamiferos, Aves	Animales exóticos y alimentos
P10	NULL	NULL	NULL

Explicación de Base de Datos

Ejemplo de consulta 1



The screenshot shows the SQL Server Management Studio interface. The query window contains the following SQL code:

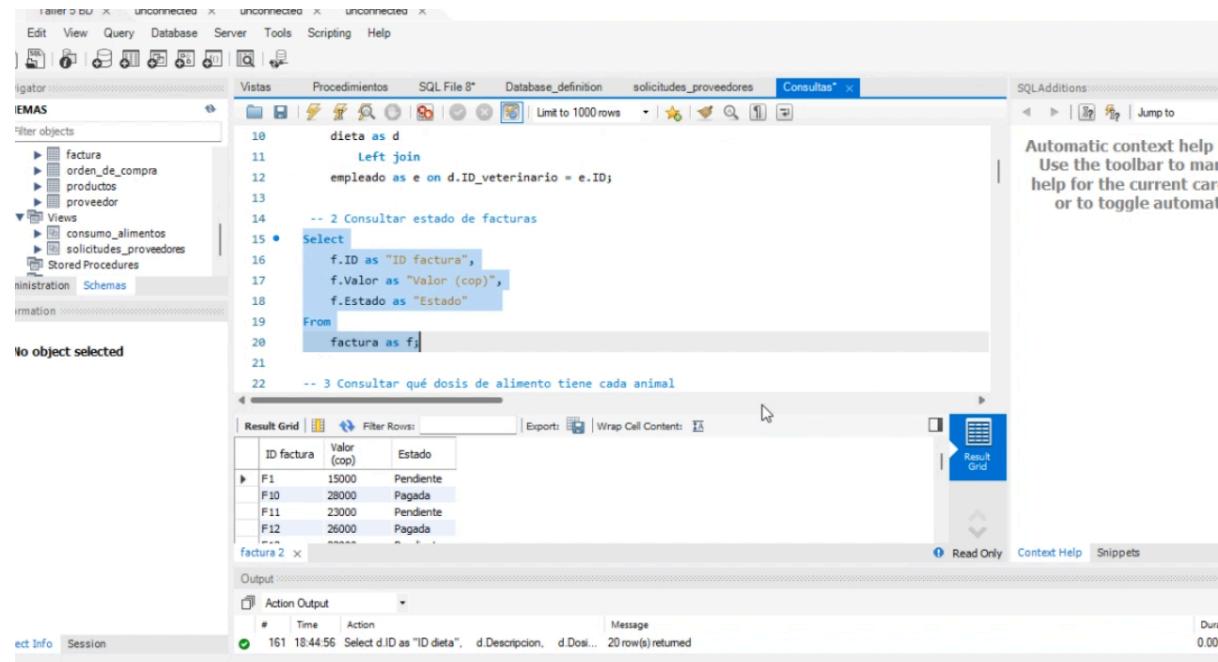
```
-- 1 Consultar qué veterinario diseñó determinada dieta
Select
    d.ID as "ID dieta",
    d.Descripcion,
    d.Dosis,
    e.nombre as "Veterinario"
From
    dieta as d
    Left join
        empleado as e on d.ID_veterinario = e.ID;
```

The results grid displays the following data:

ID dieta	Descripcion	Dosis	Veterinario
D1	Diet para leones	200g	Laura Martínez
D10	Diet para gorilas	500g	Roberto Martín
D11	Diet para serpientes	100g	Gabriel Díaz
D12	Diet para osos	400g	Francisco Ruiz
D13	Diet para aves rapaces	300g	Maria García
D14	Diet para koalas	200g	Andrés López
D15	Diet para tortugas	100g	Valeria Gutiérrez

The status bar at the bottom shows: Query Completed.

consulta 2



The screenshot shows the SQL Server Management Studio interface. The query window contains the following SQL code:

```
-- 1 Consultar qué veterinario diseñó determinada dieta
Select
    d.ID as "ID dieta",
    Left join
        empleado as e on d.ID_veterinario = e.ID;
-- 2 Consultar estado de facturas
Select
    f.ID as "ID factura",
    f.Valor as "Valor (cop)",
    f.Estado as "Estado"
From
    factura as f;
-- 3 Consultar qué dosis de alimento tiene cada animal
```

The results grid displays the following data:

ID factura	Valor (cop)	Estado
F1	15000	Pendiente
F10	28000	Pagada
F11	23000	Pendiente
F12	26000	Pagada

The status bar at the bottom shows: 161 10:44:56 Select d.ID as "ID dieta", d.Descripcion, d.Dosi... 20 row(s) returned.

Consulta 3

MySQL Workbench interface showing the results of Consulta 3. The query retrieves animal names, food names, and doses from three tables: animales, dieta, and alimento.

```
22 -- 3 Consultar qué dosis de alimento tiene cada animal
23 • SELECT
24     A.Nombre AS Nombre_Animal,
25     AL.Nombre AS Nombre_Alimento,
26     D.dosis
27 FROM
28     animales A
29 JOIN
30     dieta D ON A.ID = D.ID_animal
31 JOIN
32     alimento AL ON D.ID_alimento = AL.ID;
33
```

Nombre_Animal	Nombre_Alimento	Dosis
Leon	Carne de res	200g
Tortuga	Hojas de eucalipto	500g
Aguila	Ratas congeladas	100g
Aguila	Naranjas	400g
Tiburon	Ratoncitos vivos	300g
Hipopotamo	Bambu	200g
Hipopotamo	Gusanos de seda	100g

Result 5 | Read Only | Context Help | Snippets

Action Output | Time | Action | Message | Duration / Fetch | Object Info | Session | Completed | 162 18:46:33 Select f.ID as "ID factura", f.Valor as "Valor (cop)"... 20 rows returned | 0.000 sec / 0.000 sec | 6:52 p.m. 4/04/2024

Consulta 4

MySQL Workbench interface showing the results of Consulta 4. The query retrieves information about food items and their descriptions from four tables: Orden_de_compra, productos, Alimento, and Alimento.

```
36 • SELECT
37     O.ID AS ID_Orden,
38     A.ID AS ID_Alimento,
39     A.Nombre AS Nombre_Alimento,
40     A.Tipo AS Tipo_Alimento,
41     A.Descripcion AS Descripcion_Alimento
42 FROM
43     Orden_de_compra O
44     INNER JOIN
45     productos P ON O.ID = P.ID_orden
46     INNER JOIN
47     Alimento A ON P.ID_alimento = A.ID;
```

ID_Orden	ID_Alimento	Nombre_Alimento	Tipo_Alimento	Descripcion_Alimento
OC12	AL17	Carne de pollo	Carne	Carne de pollo magra y sin huesos
OC13	AL17	Carne de pollo	Carne	Carne de pollo magra y sin huesos
OC14	AL13	Ratoncitos vivos	Roedor	Ratoncitos vivos como alimento para aves rapaces
OC19	AL9	Granulos de pienso	Pienso	Granulos de pienso balanceado para herbivoros
OC20	AL2	Heno de alfalfa	Forraje	Heno de alfalfa para herbivoros
OC3	AL2	Heno de alfalfa	Forraje	Heno de alfalfa para herbivoros
OC4	AL4	Frutas tropicales	Fruta	Mezcla de frutas tropicales frescas

Result 6 | Read Only | Context Help | Snippets

Action Output | Time | Action | Message | Duration / Fetch | Object Info | Session | Completed | 168 18:51:59 SELECT A.Nombre AS Nombre_Animal, AL.Nombre... 20 rows returned | 0.000 sec / 0.000 sec | 6:52 p.m. 4/04/2024

Consulta 5

```

51 -- 5 Consultar la cantidad de alimentos ordenada a los proveedores
52 •
53     Select
54         p.ID as "ID proveedor",
55         p.nombre as "Nombre proveedor",
56         a.nombre as "Alimento",
57         pr.Cantidad as "Cantidad (kg)",
58         o.ID as "ID orden"
59
60     From
61         proveedor as p
62             left join
63             orden_de_compra as o on p.ID = o.ID_proveedor
64                 left join
65                 productos as pr on o.ID = pr.ID_orden
66
67
68
69
70
71
72
73
74
75
76
77
78
    
```

ID proveedor	Nombre proveedor	Alimento	Cantidad (kg)	ID orden
P1	Tierrago	Carne de res	50	OC1
P1	Tierrago	Carne de pollo	110	OC12
P1	Tierrago	Carne de pollo	75	OC13
P1	Tierrago	Granulos de pienso	75	OC19
P1	Tierrago	Pescado blanco	30	OC2
P1	Tierrago	Heno de alfalfa	100	OC20

Consulta 6

Hay empleados que no han pesado ningun animal por eso hay valores nulos

```

68 -- 6 Consultar que empleado pesó al animal
69
70 •
71     Select
72         e.nombre as "Nombre empleado",
73         a.Nombre as "Animal",
74         a.ID as "ID del animal",
75         a.Peso as "Peso del animal"
76
77     From
78         empleado as e
79             left join
80                 animales as a on e.ID=a.ID_entrenador;
    
```

Nombre empleado	Animal	ID del animal	Peso del animal
Carolina Herrera	NULL	NULL	NULL
Robero Martín	NULL	NULL	NULL
Valeria Gutiérrez	NULL	NULL	NULL
Gabriel Díaz	Aguila	A11	9 kg
Gabriel Díaz	Aguila	A12	8 kg
Isabel Jiménez	NULL	NULL	NULL
Francisco Ruiz	NULL	NULL	NULL
Camila Vargas	pez payaso	A19	0.1 kg
Maria García	Leon	A1	200 kg
Maria García	Thiuron	A13	150 kg

Consulta 7

```

-- 7 Consultar que empleados son veterinarios
SELECT e.*
FROM Empleado e
WHERE e.cargo = 'Veterinario' OR e.cargo = 'veterinaria';

-- 8 Consultar cuantos animales hay por clasificación
SELECT Clasificacion, COUNT(*) AS Cantidad
FROM Animales
GROUP BY Clasificacion;

```

Result Grid:

ID	nombre	fecha_nac	salario	cargo	direccion	telefono	email
E13	Carolina Herrera	1993-04-18	2450	Veterinaria	Calle 20, Ciudad	5553698	carolina@mail.com
E15	Valeria Gutiérrez	1988-09-15	2300	Veterinaria	Avenida 25, Ciudad	5559753	valeria@mail.com
E17	Isabel Jiménez	1986-08-28	2750	Veterinaria	Avenida 35, Ciudad	5553692	isabel@mail.com
E20	Alejandro Castro	1989-04-05	2300	Veterinario	Avenida 60, Pueblo	5559632	alejandro@mail.com
E4	Laura Martínez	1987-09-03	2600	Veterinario	Av. Central, Pueblo	5553456	lamarinez@mail.com
E6	Ana Gómez	1984-12-12	2700	Veterinario	Calle 987, Pueblo	5556789	ana@mail.com
E9	Javier Fernández	1986-06-30	2550	Veterinario	Av. Principal, Ciudad	5557891	javier@mail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Consulta 8

```

-- 7 Consultar que empleados son veterinarios
WHERE e.cargo = 'Veterinario' OR e.cargo = 'veterinaria';

-- 8 Consultar cuantos animales hay por clasificación
SELECT Clasificacion, COUNT(*) AS Cantidad
FROM Animales
GROUP BY Clasificacion;

-- 9 Consultar la cantidad de comida suministrada por cada proveedor

```

Result Grid:

Clasificacion	Cantidad
Mamifero	10
Reptil	4
Ave	3
pez	2
peces	2

Consulta 9

Taller 5 BD × unconnected × unconnected × unconnected ×

Edit View Query Database Server Tools Scripting Help

Procedimientos SQL File 8* Database_definition solicitudes_proveedores Consultas* animales empleado

MAS

Other objects

zoologico

- Tables
 - alimento
 - animales
 - comida
 - consulta
 - dietas
 - empleado

Administration Schemas Information

Table: comida

Columns:

- ID_animal varchar(10) PK
- ID_alimento varchar(10) PK

```

95
96 • SELECT
97     p.Nombre AS Proveedor, COUNT(c.ID_alimento) AS Cantidad_Comida
98 FROM
99     Proveedor p
100    LEFT JOIN
101        Alimento a ON p.ID = a.ID_proveedor
102    LEFT JOIN
103        Comida c ON a.ID = c.ID_alimento
104 GROUP BY p.Nombre;
  
```

Result Grid Filter Rows Export Wrap Cell Content

Proveedor	Cantidad_Comida
Tierragro	11
FurryFriends	0
FeatheredFriends	0
ReptileHaven	0
PawPrints	0
FinsAndScales	0
CritterCare	0
TailWaggers	0
PurrFect	0
WhiskerWonders	0
WildWings	0

Result 13 × Read Only

Output

Action Output

Consulta 10

Taller 5 BD × unconnected × unconnected × unconnected ×

Edit View Query Database Server Tools Scripting Help

Procedimientos SQL File 8* Database_definition solicitudes_proveedores Consultas* animales empleado SQLAdditions

MAS

Other objects

zoologico

- Tables
 - alimento
 - animales
 - comida
 - consulta
 - dietas
 - empleado

Administration Schemas Information

Table: consulta

Columns:

- ID_animal varchar(10) PK
- ID_alimento varchar(10) PK

```

100
101 • SELECT
102     cargo, COUNT(*) AS Cantidad_Personas
103 FROM
104     Empleado
105    INNER JOIN cargo
106
107
108
109
110
111
112
113
114
115
116
117
  
```

Result Grid Filter Rows Export Wrap Cell Content

cargo	Cantidad_Personas
Entrenador	1
Entrenador	6
Veterinaria	3
Veterinaria	3
Logista	2

Result 14 × Read Only Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

File Edit View Query Database Server Tools Scripting Help

Schemas

zoologico

- Tables
 - alimento
 - animales
 - comida
 - consulta
 - dietas
 - empleado

Administration Schemas Information

Table: comida

Columns:

- ID_animal varchar(10) PK
- ID_alimento varchar(10) PK

```

94
95 • 9 Consultar la cantidad de comida suministrada por cada proveedor
96
97
98 • SELECT
99     p.Nombre AS Proveedor,
100     COUNT(c.ID_alimento) AS Cantidad_Comida
101
102 FROM
103     Proveedor p
104    LEFT JOIN
105        Alimento a ON p.ID = a.ID_proveedor
106    LEFT JOIN
107        Comida c ON a.ID = c.ID_alimento
108
109 GROUP BY
110     p.Nombre
111
112 ORDER BY
113     Cantidad_Comida DESC;
  
```

Result Grid Filter Rows Export Wrap Cell Content

Proveedor	Cantidad_Comida
Tierragro	11
HabitatWorld S.A.	6
VetSupplies Ltd.	3
FurryFriends	0
FeatheredFriends	0

Result 14 Result 15 × Read Only Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Vistas

Esta vista es importante porque proporciona información sobre cuánto se está consumiendo de cada tipo de alimento por cada clasificación en el zoológico. Esto es esencial para monitorear la dieta de los animales y también para tener en cuenta estos consumos para registros de gastos, órdenes de compra, ETC

- ```
CREATE VIEW Consumo_Alimentos AS
SELECT
 A.Clasificacion AS Clasificacion_Animal,
 AL.Tipo AS Tipo_Alimento,
 COUNT(*) AS Cantidad
FROM
 Animales A
 INNER JOIN
 Dieta D ON A.ID = D.ID_animal
 INNER JOIN
 Alimento AL ON D.ID_alimento = AL.ID
GROUP BY A.Clasificacion, AL.Tipo;
```

Esta vista es crucial para el proceso de gestión de alimentos del zoológico. Permite rastrear cuántas veces se solicita alimento a cada proveedor, lo que ayuda a evaluar la eficacia de los proveedores y a gestionar adecuadamente los niveles de existencias de alimentos en el zoológico.

- ```
CREATE VIEW Solicitudes_Proveedores AS
SELECT
    P.Nombre AS Nombre_Proveedor,
    COUNT(*) AS Solicitudes
FROM
    Orden_de_compra OC
    INNER JOIN
    Proveedor P ON OC.ID_proveedor = P.ID
GROUP BY P.Nombre;
```

Esta vista proporciona una visión integrada de la actividad de los veterinarios en el zoológico, mostrando la cantidad de consultas realizadas y las dietas diseñadas por cada veterinario. Permite evaluar la carga de trabajo de los veterinarios y su contribución al cuidado y bienestar de los animales

```

CREATE VIEW Actividad_Veterinarios AS
SELECT
    E.nombre AS Nombre_Veterinario,
    COUNT(C.ID) AS Consultas_Realizadas,
    COUNT(D.ID) AS Dietas_Diseñadas
FROM
    Empleado E
    LEFT JOIN
    Consulta C ON E.ID = C.ID_empleado
    LEFT JOIN
    Dieta D ON E.ID = D.ID_veterinario
WHERE E.cargo = 'Veterinario'
GROUP BY E.nombre;

```

Esta vista muestra todas las facturas que están en estado "Por pagar". Es útil para el gerente del zoológico tener una visión clara de las facturas pendientes de pago y proceder con este

```

CREATE VIEW Facturas_Por_Pagar AS
SELECT *
FROM
    Factura
WHERE Estado = 'Por pagar';

```

Esta vista es importante para monitorear la salud y el desarrollo de las especies en el zoológico. Proporciona información sobre el peso promedio de cada especie, lo que ayuda a los entrenadores y veterinarios a evaluar su estado de salud y detectar cualquier cambio significativo en su peso que pueda indicar problemas de salud o bienestar

```

CREATE VIEW Peso_Promedio_Especies AS
SELECT
    Animales.Clasificacion AS Especie,
    AVG(CAST(Animales.Peso AS DECIMAL(10, 2))) AS Peso_Promedio
FROM
    Animales
GROUP BY Animales.Clasificacion;

```

Ejemplo de vistas en Funcionamiento

MySQL Workbench

Taller 5 BD × unconnected × unconnected × unconnected ×

File Edit View Query Database Server Tools Scripting Help

Vistas consumo_alimentos

Automatic context help is disabled
Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export | Wrap Cell Content |

Clasificación_Animal	Tipo_Alimento	Cantidad
Mamífero	Carna	2
Reptil	Forraje	1
Ave	Pájaro	1
Ave	Fruta	1
pec	Pájaro	1
Mamífero	Pájaro	1
Mamífero	Proteína	2
Pez	Pienso	1
Reptil	Semilla	1
ave	Aloja	1

Output

Action Output

- # Time Action Message Duration / Fetch
- 185 19:01:03 SELECT p.Nombre AS Proveedor, COUNT(*) AS Cantidad_Personas F... 20 row(s) returned 0.000 sec / 0.000 sec
- 186 19:01:39 SELECT cargo, COUNT(*) AS Cantidad_Personas F... 7 row(s) returned 0.000 sec / 0.000 sec
- 187 19:02:18 CREATE VIEW Consumo_Alimentos AS SELECT A... Error Code: 1050. Table 'Consumo_Alimentos' already exists 0.000 sec
- 188 19:02:42 SELECT * FROM zoologico.consumo_alimentos LIMI... 18 row(s) returned 0.000 sec / 0.000 sec

Taller 5 BD × unconnected × unconnected × unconnected ×

File Edit View Query Database Server Tools Scripting Help

Vistas solicitudes_proveedores

Automatic context help is disabled
Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export | Wrap Cell Content |

Nombre_Proveedor	Solicitudes
Tierrago	11
VetSolutions Ltd.	5
HabitatWorld S.A.	4

Output

Action Output

- # Time Action Message Duration / Fetch
- 187 19:02:18 CREATE VIEW Consumo_Alimentos AS SELECT A... Error Code: 1050. Table 'Consumo_Alimentos' already exists 0.000 sec
- 188 19:02:42 SELECT * FROM zoologico.consumo_alimentos LIMI... 18 row(s) returned 0.000 sec /
- 189 19:04:14 CREATE VIEW Solicitudes_Proveedores AS SELEC... Error Code: 1050. Table 'Solicitudes_Proveedores' already exists 0.000 sec
- 190 19:04:33 SELECT * FROM zoologico.solicitudes_proveedores ... 3 row(s) returned 0.000 sec /

Procedimientos

Ejemplo de procedimientos en funcionamientos

Procedimiento 1

The screenshot shows the 'Procedimientos' (Stored Procedures) node in the left sidebar of the SQL Server Management Studio interface. A new stored procedure is being created with the following code:

```
1 • use zoologicos
2 DELIMITER //
3
4 • CREATE PROCEDURE AgregarDieta(
5     IN descripcion_dieta VARCHAR(40),
6     IN dosis_dieta VARCHAR(10),
7     IN ID_veterinario VARCHAR(10),
8     IN ID_animal VARCHAR(10),
9     IN ID_alimento VARCHAR(10)
10 )
11 BEGIN
12     -- Insertar nueva dieta en la tabla Dieta
13     INSERT INTO Dieta (Descripcion, Dosis, ID_veterinario, ID_animal, ID_alimento)
14         VALUES (descripcion_dieta, dosis_dieta, ID_veterinario, ID_animal, ID_alimento);

```

A call dialog box titled 'Call stored procedure zoologico.AgregarDieta' is open, prompting for parameter values. The parameters are:

- descripcion_dieta [IN] VARCHAR(40)
- dosis_dieta [IN] VARCHAR(10)
- ID_veterinario [IN] VARCHAR(10)
- ID_animal [IN] VARCHAR(10)
- ID_alimento [IN] VARCHAR(10)

Buttons for 'Execute' and 'Cancel' are visible at the bottom right of the dialog.

Procedimiento 2

The screenshot shows the 'Procedimientos' (Stored Procedures) node in the left sidebar. A new stored procedure is being created with the following code:

```
19 DELIMITER //
20
21 • CREATE PROCEDURE CargarOrdenCompra(
22     IN fecha_orden DATE,
23     IN id_empleado VARCHAR(10),
24     IN id_proveedor VARCHAR(10),
25     IN productos_lista TEXT,
26     IN cantidades_lista TEXT
27 )
28 BEGIN
29     DECLARE productos_array TEXT DEFAULT productos_lista;
30     DECLARE cantidades_array TEXT DEFAULT cantidades_lista;
31     DECLARE productos_count INT;
32     DECLARE i INT DEFAULT 1;
33     DECLARE producto_id VARCHAR(10);
34     DECLARE cantidad VARCHAR(20);

```

A call dialog box titled 'Call stored procedure zoologico.CargarOrdenCompra' is open, prompting for parameter values. The parameters are:

- fecha_orden [IN] DATE
- id_empleado [IN] VARCHAR(10)
- id_proveedor [IN] VARCHAR(10)
- productos_lista [IN] TEXT
- cantidades_lista [IN] TEXT

Buttons for 'Execute' and 'Cancel' are visible at the bottom right of the dialog.

Procedimiento 3

The screenshot shows the 'Procedimientos' (Stored Procedures) node in the left sidebar. A new stored procedure is being created with the following code:

```
58 DELIMITER //
59
60 • CREATE PROCEDURE AlimentarAnimal(
61     IN id_animal VARCHAR(10),
62     IN id_alimento VARCHAR(10),
63     IN cantidad_alimento INT,
64     IN fecha_alimentacion DATE
65 )
66 BEGIN
67     -- Insertar registro en la tabla Comida para registrar la alimentación del animal
68     INSERT INTO Comida (ID_animal, ID_alimento)
69         VALUES (id_animal, id_alimento);
70
71     -- Actualizar la tabla Dieta si es necesario
72     UPDATE Dieta

```

A call dialog box titled 'Call stored procedure zoologico.AlimentarAnimal' is open, prompting for parameter values. The parameters are:

- id_animal [IN] VARCHAR(10)
- id_alimento [IN] VARCHAR(10)
- cantidad_alimento [IN] INT
- fecha_alimentacion [IN] DATE

Buttons for 'Execute' and 'Cancel' are visible at the bottom right of the dialog.

Triggers

Aquí se genera una tabla para guardar los cambios. En cuando se inserta una nueva dieta, se actualiza o se elimina.

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' section, the 'zoológico' schema is selected. Under 'Tables', several tables are listed: 'alimento', 'animales', 'comida', 'consulta', 'dieta', 'empleado', 'especialidad_empleo', 'factura', 'orden_de_compra', 'productos', 'proveedor', and 'registrocambiosdieta'. A 'Stored Procedures' folder contains a procedure named 'AgregaDieta'. In the center pane, the 'Triggers' tab is selected. The code editor displays the following SQL script:

```
-- Crear una tabla para almacenar los cambios en la dieta
CREATE TABLE RegistroCambiosDieta (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    TipoCambio VARCHAR(10), -- Puede ser 'INSERT', 'DELETE' o 'UPDATE',
    FechaCambio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Detalles VARCHAR(255)
);

-- Trigger para registrar inserciones en la tabla de dieta
DELIMITER $$ 
CREATE TRIGGER tr_insertar_dieta
AFTER INSERT ON Dieta
FOR EACH ROW
BEGIN
    INSERT INTO RegistroCambiosDieta (TipoCambio, Detalles)
    VALUES ('INSERT', CONCAT('Se agregó la dieta con ID ', NEW.ID));
END$$
DELIMITER ;

-- Trigger para registrar eliminaciones en la tabla de dieta
```

Below the code editor, the 'Output' pane shows the results of a recent query:

#	Time	Action	Message	Duration / Fetch
218	19:24:48	INSERT INTO Dieta (Nombre, Descripcion) VALUES (...)	Error Code: 1054. Unknown column 'Nombre' in field list!	0.000 sec

Plan de pruebas y matriz de riesgo.

Estas fueron planteadas desde la pensando en el Zoológico de Cali, una aplicación web, que por medio de una interfaz se pudieran realizar todas las funciones de almacenamiento de datos para alimentos y animales.

En la matriz de riesgo se agregaron datos de riesgos de producto y proyecto de acuerdo a los que consideramos necesarios ya que no había una cantidad definida de los mismos.

Escenarios de prueba

En los escenarios de prueba nos basamos en una página web que se conectaba al back-end y a la base de datos por medio de servicio tipos rest. Se hicieron pensando en el testeo de estas mismas con su funcionalidad en la base de datos.

```
Users / Jorge / OneDrive / Documentos / JORGE / SOFKA / TrainingQA / T3 / 2024-CT-QA-BD-T05P / ZOO / Escenarios_de_prueba_ZOO.feature

Scenario: Listar Empleados Registrados en el Sistema

@E2
Scenario: Mostrar Detalles de un Animal unico en el Sistema
Given El usuario con rol de cuidador esta en la pagina de detalles de animales
And Existe un animal unico registrado en el sistema con el codigo 2
When busca el animal con codigo 2 en el sistema
Then El sistema deberia mostrar los detalles completos del animal con codigo 2
And Los detalles incluyen informacion como especie, estado de salud, alimento

@E3
Scenario: Buscar Detalles de un Animal unico No Encontrado en el Sistema
Given El usuario con rol de cuidador esta en la pagina de detalles de animales
And No existe un animal registrado en el sistema con el codigo 23
When intenta buscar el animal con codigo 23 en el sistema
Then Deberia ver un mensaje indicando que el animal no fue encontrado

@E4
Scenario: Crear un Empleado Mediante Solicitud POST
Given El usuario con rol de administrador crea un usuario nuevo esta en la pagina de gestion de empleados
When El usuario envia una solicitud POST con los datos del nuevo empleado
Then Deberia procesarse la solicitud y crea exitosamente el nuevo empleado
And mostrar un codigo de estado HTTP 201 que indica el exito de la solicitud

@E5 Scenario: Actualizar Cargo de un Empleado Mediante Solicitud PUT
Given El usuario con rol de administrador esta en la pagina de gestion de empleados
And Existe un empleado registrado en el sistema con el ID 2 y el cargo Cuidador
When El usuario envia una solicitud PUT para actualizar el cargo del empleado con el ID 2
Then El sistema deberia procesar la solicitud y actualiza exitosamente el cargo del empleado con el ID 2
And Devolver un codigo de estado HTTP 200 que significa que todo salio bien

@E6
Scenario: Eliminar un Proveedor Mediante Solicitud DELETE
Given El usuario con rol de administrador esta en la pagina de gestion de proveedores
And Existe un proveedor registrado en el sistema con el ID 5 y el nombre Tierragro
When Envia una solicitud DELETE para eliminar el proveedor con el ID 5
Then El sistema deberia procesar la solicitud y eliminar exitosamente el proveedor con el ID 5
And Debe devolver un codigo de estado HTTP 204 que significa exitoso la eliminacion

@E7
Scenario: Intentar Eliminar un Proveedor No Existente Mediante Solicitud DELETE
Given El usuario con rol de administrador esta en la pagina de gestion de proveedores
And No existe un proveedor registrado en el sistema con el ID 10
When Envia una solicitud DELETE para eliminar el proveedor con el ID 10
Then Se deberia procesar la solicitud y no encontrar el proveedor con el ID 10 para eliminarlo
And Deberia verse un codigo de estado HTTP 404 que significa no encontrado
```

Como se puede ver en los escenarios de prueba hay unos que piden más el servicio put, get, o post de manera más literaria y otros en los que está explícito esas pruebas.

Pedido	Respuesta
GET LISTAR USUARIOS	
GET USUARIO ÚNICO	
GET USUARIO ÚNICO NO ENCONTRADO	
GET LISTA <RECURSO>	
GET <RECURSO> ÚNICO	
GET <RECURSO> ÚNICO NO ENCONTRADO	
POST CREAR	
PUT ACTUALIZAR	
PATCH ACTUALIZAR	
DELETE BORRAR	
POST REGISTRO - EXITOSO	
POST REGISTRO - FALLIDO	
POST INICIO DE SESIÓN CORRECTO	
POST INICIAR SESIÓN - SIN ÉXITO	
GET RESPUESTA TARDÍA	
/api/usuarios?page=2	
200	
{ "página": 2, "por_página": 6, "total": 12, "total_páginas": 2, "datos": [{ "identificación": 7, "correo_electrónico": "michael.lawson@reqres.in", "first_name": "Michael", "last_name": "Lawson", "avatar": "https://reqres.in/img/faces/michael.lawson.jpg" }, { "identificación": 8, "correo_electrónico": "lindsay.ferguson@reqres.in", "first_name": "Lindsay", "apellidos": "Ferguson", "avatar": "https://reqres.in/img/faces/lindsay.ferguson.jpg" }, { "identificación": 9, "correo_electrónico": "tobias.funke@reqres.in", "first_name": "Tobias", "last_name": "Funke", "avatar": "https://reqres.in/img/faces/tobias.funke.jpg" }, { "identificación": 10, "correo_electrónico": "byron.fields@reqres.in", "first_name": "Byron", "last_name": "Fields", "avatar": "https://reqres.in/img/faces/byron.fields.jpg" }, { "identificación": 11, "correo_electrónico": "george.edwards@reqres.in", "first_name": "George", "last_name": "Edwards", "avatar": "https://reqres.in/img/faces/george.edwards.jpg" }] }	