

UNIVERSIDAD AUTÓNOMA DE BENI
“JOSÉ BALLIVIAN”
FACULTAD DE INGENIERÍA Y TECNOLOGÍA
CARRERA INGENIERÍA DE SISTEMAS



**Diseño e Implementación de la Base de Datos
para la Gestión y Monitoreo de Salud del Ganado
en el Sector Agropecuario del Departamento del
Beni, Bolivia**

Universitario (a): Osias Peña Chaurara

Diego Andrés Mercado Pinto

Docente: Ing. Hermes Rodríguez Rivero

Materia: BASE DE DATOS I

DICIEMBRE 2023

INDICE

Contenido

Diseño e Implementación de la Base de Datos para la Gestión y Monitoreo de Salud del Ganado en el Sector Agropecuario del Departamento del Beni, Bolivia	1
Introducción	3
Objetivo	4
II. Diseño de la base de datos	4
2.1 Planteamiento del problema:	4
2.1.1. Identificación de las entidades y relaciones	5
2.2. Diseño Conceptual Del Modelo Entidad Relación:	6
2.3. Modelo Entidad Relación Al Modelo Relacional:	7
2.4. Normalización:	8
2.4.1. Primera forma normal	8
2.4.2. Segunda forma normal	8
2.4.3. Tercera forma normal	8
2.5. Todas Las Tablas Normalizadas:	10
III. Implementación del Sistema de Gestor de Base de Datos	12
3.1. Selección del Sistema	12
3.2. Instalación y Configuración	12
3.3. Creación de las Tablas	12
3.3.1. Creación de la Base de Datos	12
3.3.2. Creación de la Tabla ESTANCIA	13
3.3.3. Creación de la Tabla RAZA	13
3.3.4. Creación de la Tabla ENFERMEDAD	13
3.3.5. Creación de la Tabla ANIMAL	14
3.3.6. Creación de la Tabla ENFERMEDAD_X_ANIMAL	14

3.3.7. Creación de la Tabla MEDICAMENTO	15
3.3.8. Creación de la Tabla MEDICAMENTO_X_ENFERMEDAD	15
3.3.9. Creación de la Tabla MEDICAMENTO_X_ANIMAL	15
3.4. Carga Inicial de Datos en la Base de Datos	16
3.4.1. Inserción de Datos en la Tabla "ESTANCIA":	16
3.4.2. Inserción de Datos en la Tabla "RAZA":	16
3.4.3. Inserción de Datos en la Tabla "ENFERMEDAD":	16
3.4.4. Inserción de Datos en la Tabla "ANIMAL":	16
3.4.5. Inserción de Datos en la Tabla "ENFERMEDAD_X_ANIMAL":	16
3.4.6. Inserción de Datos en la Tabla "MEDICAMENTO":	17
3.4.7. Inserción de Datos en la Tabla "MEDICAMENTO_X_ENFERMEDAD":	17
3.4.8. Inserción de Datos en la Tabla "MEDICAMENTO_X_ANIMAL":	17
IV. Consultas SQL para manipulación y obtención de información	17
4.1. 5 consultas con una tabla	17
4.2. 5 Consultas multitaslas composición interna	18
4.3. 5 Consultas multitaslas composición externa	19
4.4. 5 Consultas con anidadas o subconsultas	20
4.5. 5 Consultas con operadores de conjuntos	22
4.6. 5 Consultas con funciones agregadas	23
V. Pruebas y validación del sistema	24
5.1. Realización de pruebas	24
	26
VI. Conclusiones y recomendaciones	26
VII. Referencias bibliográficas	27
Bibliografía	27

Introducción

El Departamento del Beni, en Bolivia, destaca por ser una región donde la ganadería desempeña un papel fundamental en su desarrollo socioeconómico. Sin embargo, a pesar de su importancia, el sector agropecuario enfrenta desafíos significativos en cuanto a la gestión eficiente del ganado bovino.

La gestión de la salud del ganado bovino es un tema de vital importancia en el sector agropecuario del Departamento del Beni, Bolivia. La ganadería desempeña un papel económico sin precedentes en el desarrollo de esta región, con una población de bovinos que supera los 2,8 millones y representa el 32% del hato ganadero de Bolivia (Oscar Durán, 2021). Actualmente, la falta de un sistema centralizado dificulta la toma de decisiones informadas y limita la maximización de la productividad en el sector. Para abordar esta problemática, es crucial contar con una solución integral que permita optimizar las operaciones relacionadas con la salud y seguimiento de cada bovino en la región.

En la región del Beni, se enfrentan desafíos relacionados con enfermedades comunes que afectan al ganado bovino, como la brucelosis, leptospirosis, rinotraqueítis infecciosa bovina (IBR), diarrea viral bovina (DVB), tricomoniasis, campilobacteriosis, carbúnculo, mastitis, diarrea negra, piroplasmosis, rabia bovina, anemia infecciosa, fiebre aftosa, entre otras (TOTALPEC, 2022).

Para abordar estas enfermedades y garantizar la salud del ganado, es fundamental que los productores implementen una gestión eficiente basada en buenas prácticas, como la cuarentena y la vacunación oportuna de animales nuevos. Además, es necesario establecer planes sanitarios y protocolos que prevengan la aparición y propagación de estas y otras enfermedades.

Objetivo

El objetivo principal de este proyecto es desarrollar una solución basada en una base de datos relacional que mejore la gestión del sector agropecuario en el Departamento del Beni, Bolivia. A lo largo del proyecto, se realizará un diseño detallado de la base de datos, incluyendo la normalización, así como su implementación en un sistema de gestor de base de datos. Además, se llevarán a cabo consultas SQL para manipular y obtener información relevante de la base de datos.

La implementación de esta solución integral brindará beneficios significativos para el sector ganadero del Departamento del Beni. Al contar con un sistema centralizado y eficiente, se facilitará la toma de decisiones informadas, se optimizará el seguimiento de la salud de cada bovino y se maximizará la productividad en el sector agropecuario.

En este proyecto, nos enfocaremos en desarrollar una solución integral que mejore la gestión de la salud del ganado bovino en el Departamento del Beni. Mediante el diseño de una base de datos relacional y la implementación de consultas SQL, buscamos optimizar las operaciones relacionadas con la salud y seguimiento de cada bovino, permitiendo una toma de decisiones informada y una maximización de la productividad en el sector agropecuario.

II. Diseño de la base de datos

2.1 Planteamiento del problema:

El sector agropecuario en el Departamento del Beni, Bolivia, se enfrenta a desafíos en la gestión eficiente del ganado bovino. Se requiere una solución integral que permita optimizar las operaciones relacionadas con la salud y seguimiento de cada bovino en la región.

Actualmente, la falta de un sistema centralizado dificulta la toma de decisiones informadas y la maximización de la productividad.

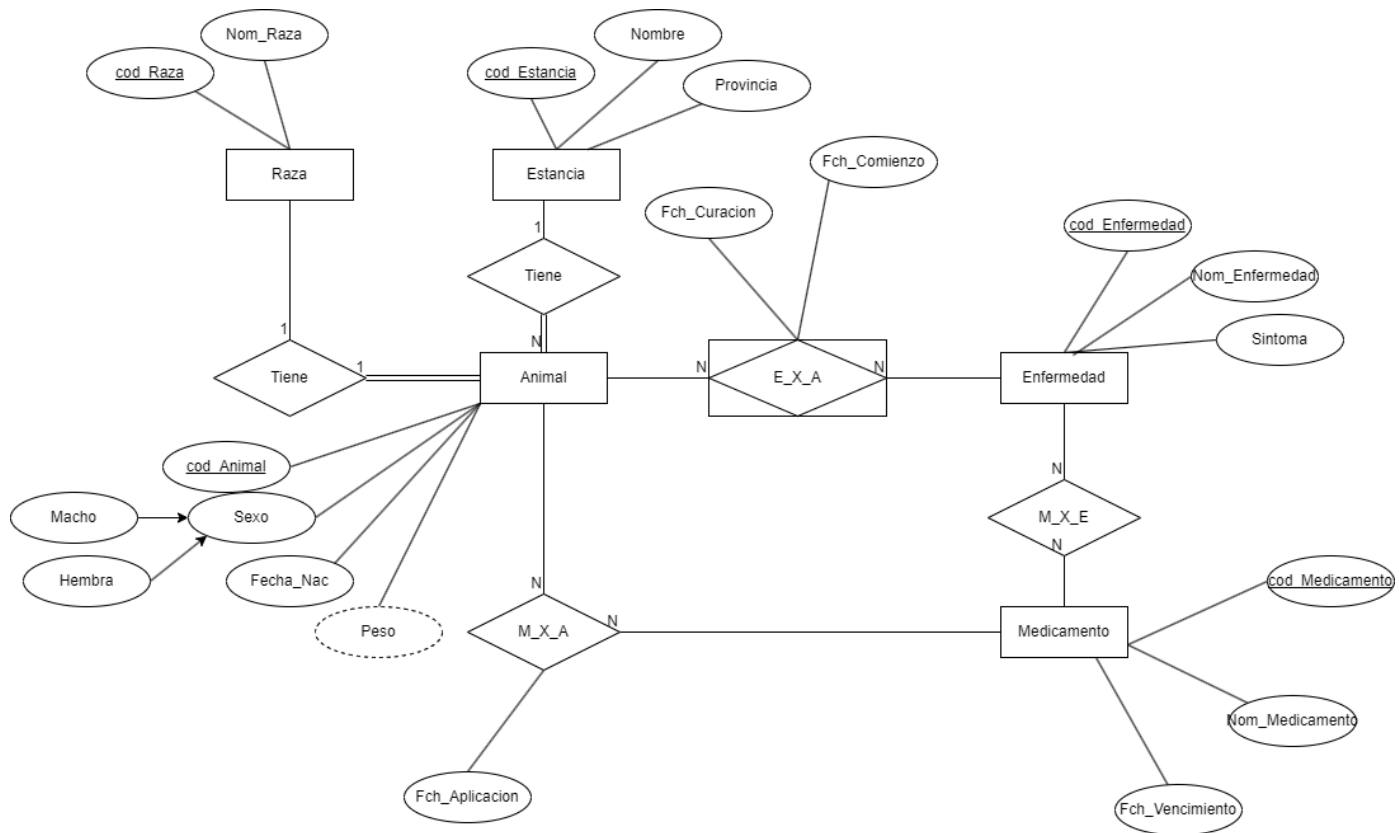
2.1.1. Identificación de las entidades y relaciones

Necesidades Específicas en el Contexto del Departamento del Beni:

- **Estancias:** Se requiere un seguimiento preciso de cada estancia, considerando su ubicación geográfica y otros detalles específicos relevantes para el contexto del Departamento del Beni.
- **Razas:** La identificación y clasificación de las razas presentes en el Departamento, permitiendo comprender la diversidad genética del ganado.
- **Animales:** Una gestión individualizada de los animales, incorporando detalles como el identificador único (Chip), datos de nacimiento, peso y su relación con las estancias y razas correspondientes.
- **Enfermedades:** Un sistema que registre las enfermedades específicas en el ganado, incluyendo síntomas, fechas de diagnóstico y curación, con un enfoque en las particularidades epidemiológicas del Departamento del Beni.
- **Medicamentos:** La administración eficiente de medicamentos, con un registro detallado que contemple la disponibilidad y aplicación de tratamientos, considerando las necesidades específicas del Departamento.

Objetivo del Proyecto: El proyecto tiene como objetivo principal diseñar e implementar una Base de Datos que optimice la gestión del sector agropecuario en el Departamento del Beni. Se busca proporcionar una herramienta robusta que permita a los productores y profesionales del sector tomar decisiones informadas, mejorar la salud del ganado y contribuir al desarrollo sostenible de las explotaciones ganaderas en esta región de Bolivia.

2.2. Diseño Conceptual Del Modelo Entidad Relación:



2.3. Modelo Entidad Relación Al Modelo Relacional:

ESTANCIA:

<u>Cod_Estancia</u>	Nombre

RAZA:

<u>Cod_Raza</u>	Nom_Raza
-----------------	----------

ANIMAL:

<u>Cod_Animal</u>	Sexo	Fecha_Nac	Peso	Cod_Estancia	Cod_Raza

ENFEMEDAD:

<u>Cod_Enfermedad</u>	Nom_Enfermedad	Síntoma

E_X_A:

<u>Cod_Animal</u>	<u>Cod_Enfermedad</u>	Fch_Coniezo	Fch_Curacion	Efectivo

MEDICAMENTO:

<u>Cod_Medicamento</u>	Nom_Medicamento	Vencimiento

M_X_E:

<u>Cod_Medicamento</u>	<u>Cod_Enfermedad</u>

M_X_A:

<u>Cod_Animal</u>	<u>Cod_Medicamento</u>	Fch_Aplicacion

2.4. Normalización:

2.4.1. Primera forma normal

- Elimine los grupos repetidos de las tablas individuales.
- Cree una tabla independiente para cada conjunto de datos relacionados.

- Identifique cada conjunto de datos relacionados con una clave principal.

2.4.2. Segunda forma normal

- Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
- Relacione estas tablas con una clave externa.

2.4.3. Tercera forma normal

- Elimine los campos que no dependen de la clave.

Cada nivel de normalización exige cierta complejidad, pero debes analizar si en realidad necesitas llegar hasta la tercera forma normal, debes buscar lo que mejor se adapte a la empresa (ED_TEAM, 2023).

Verificación de que cada tabla cumpla con las Tres formas normales:

ESTACIA
Cod_Estancia (PK)
Nombre
ENFERMEDAD
Cod_Enfermedad(PK)
RAZA
Nom_Enfermedad
Cod_Raza (PK)
Sintoma
Nom_Raza
MEDICAMENTO
Cod_Medicamento (PK)
Nom_Medicamento
ANIMAL
Fch_Vencimiento
Cod_Animal (PK)
Sexo
Fecha_Nac
Peso
Cod_Raza (FK)
Cod_Estancia (FK)

ENFERMEDAD_X_
ANIMAL
Cod_Animal(FK)
Cod_Enfermedad(FK)
Fch_Comienzo
Fch_Curacion
Efectivo
MEDICAMENTO_X_
ENFERMEDAD
Cod_Medicamento(FK)
Cod_Animal(FK)

MEDICAMENTO_X_
ANIMAL
Cod_Animal(FK)
Cod_Medicamento(FK)
Fch_Aplicacion

TABLAS	1FN	2FN	3FN
--------	-----	-----	-----

ESTANCIA	SI	SI	SI
RAZA	SI	SI	SI
ANIMAL	SI	SI	SI
ENFERMEDAD	SI	SI	SI
ENFERMEDAD X ANIMAL	NO	NO	NO
MEDIACMENTO	SI	SI	SI
MEDICAMENTO X ENFERMEDAD	SI	SI	SI
MEDICAMENTO X ANIMAL	NO	NO	NO

En conclusión:

Tenemos que las Tablas ENFERMEDAD_X_ANIMAL y MEDICAMENTO_X_ANIMAL No están normalizadas por lo que no cumplen con la 1FN “Identifique cada conjunto de datos relacionados con una clave principal”, por ende, le asignaremos una clave primaria a cada tabla.

ENFERMEDAD_X_ ANIMAL
ID_EA (CK) Cod_Animal(FK) Cod_Enfermedad(FK) Fch_Comienzo Fch_Curacion Efectivo

MEDICAMENTO_X_ ANIMAL
ID_MA(PK) Cod_Animal(FK) Cod_Medicamento(FK) Fch_Aplicacion

De esta forma las Tablas cumplen con la 1FN, 2FN y con la 3FN.

2.5. Todas Las Tablas Normalizadas:

ESTANCIA (Cod_Estancia, Nombre)

RAZA (Cod_Raza, Nom_Raza)

ANIMAL (Cod_Animal, Sexo, Fecha_Nac, peso, cod_Estancia, cod_Raza)

ENFERMEDAD_X_ANIMAL (ID_EA, Fch_Enfermedad, Fch_Curación, Efectivo, Cod_Enfermedad, Cod_Animal)

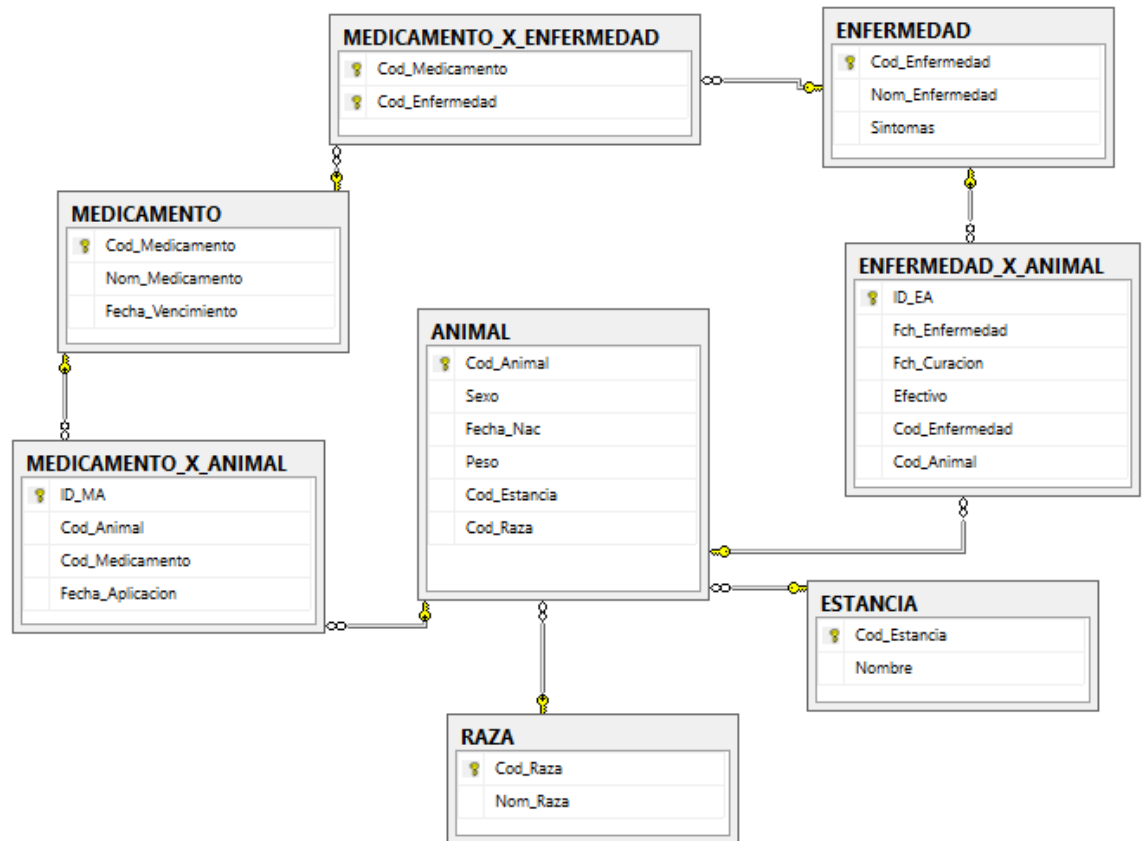
ENFERMEDAD (Cod_Enfermedad, Nom_Enfermedad, Síntomas)

MEDICAMENTO_X_ENFERMEDAD (Cod_Medicamento, Cod_Enfermedad)

MEDICAMENTO (Cod_Medicamento, Nom_Medicamento, Fecha_Vencimiento)

MEDICAMENTO_X_ANIMAL (ID_MA, Cod_Animal, Cod_Medicamento, Fecha_Aplicación)

ESQUEMA RELACIONAL SQL:



Activar Wi

III. Implementación del Sistema de Gestor de Base de Datos

3.1. Selección del Sistema

La presente etapa se enfoca en la meticulosa evaluación y selección del Sistema de Gestión de Base de Datos (SGBD) más congruente con las particularidades y requisitos específicos de la empresa agropecuaria en el Departamento del Beni. Este proceso implica un análisis detallado de las características y capacidades de diversos SGBD disponibles en el mercado, con el objetivo de elegir la plataforma que maximice la eficiencia y coherencia del sistema.

3.2. Instalación y Configuración

Una vez seleccionado el SGBD óptimo, se procederá a su implementación técnica mediante la instalación y configuración adecuada. Esta fase reviste importancia sustancial, ya que sienta las bases para un entorno de base de datos operativo y ajustado a los requerimientos específicos del proyecto agropecuario. La configuración del sistema se ejecutará conforme a las pautas establecidas por las mejores prácticas de gestión de bases de datos.

3.3. Creación de las Tablas

Este segmento de la implementación se centra en la definición e implementación de la estructura de la base de datos. Se llevará a cabo un minucioso diseño, identificando entidades cruciales y estableciendo relaciones intrínsecas entre ellas. La creación de tablas, en concordancia con los principios de normalización, será ejecutada con el propósito de optimizar la eficiencia del almacenamiento y la consulta de datos.

3.3.1. Creación de la Base de Datos

```
-- Crear la base de datos
CREATE DATABASE BD_SECTOR_GANADERO;
GO

-- Usar la base de datos
USE BD_SECTOR_GANADERO;
GO
```

La creación de la base de datos "BD_SECTOR_GANADERO" establece el entorno central para la gestión de información relacionada con el sector ganadero en el Departamento del Beni.

3.3.2. Creación de la Tabla ESTANCIA

```
-- Tabla ESTANCIA
CREATE TABLE ESTANCIA (
    Cod_Estancia INT PRIMARY KEY,
    Nombre VARCHAR(50)
);
GO
```

La tabla "ESTANCIA" contiene información sobre las estancias ganaderas, identificadas por un código único y acompañadas de un nombre descriptivo.

3.3.3. Creación de la Tabla RAZA

```
-- Tabla RAZA
CREATE TABLE RAZA (
    Cod_Raza INT PRIMARY KEY,
    Nom_Raza VARCHAR(50)
);
GO
```

La tabla "RAZA" almacena datos sobre las razas de ganado, con un código único y un nombre que las identifica.

3.3.4. Creación de la Tabla ENFERMEDAD

```
-- Tabla ENFERMEDAD
CREATE TABLE ENFERMEDAD (
    Cod_Enfermedad INT PRIMARY KEY,
    Nom_Enfermedad VARCHAR(50),
    Sintomas VARCHAR(100)
);
GO
```

La tabla "ENFERMEDAD" registra información sobre enfermedades que pueden afectar al ganado, incluyendo un código, nombre y descripción de los síntomas.

3.3.5. Creación de la Tabla ANIMAL

```
-- Tabla ANIMAL
CREATE TABLE ANIMAL (
    Cod_Animal INT PRIMARY KEY,
    Sexo CHAR(1),
    Fecha_Nac DATE,
    Peso DECIMAL(10, 2),
    Cod_Estancia INT,
    Cod_Raza INT,
    FOREIGN KEY (Cod_Estancia) REFERENCES ESTANCIA(Cod_Estancia),
    FOREIGN KEY (Cod_Raza) REFERENCES RAZA(Cod_Raza)
);
GO
```

La tabla "ANIMAL" almacena datos individuales de los animales, incluyendo su código único, sexo, fecha de nacimiento, peso, y referencias a la estancia y raza correspondientes.

3.3.6. Creación de la Tabla ENFERMEDAD_X_ANIMAL

```
-- Tabla ENFERMEDAD_X_ANIMAL
CREATE TABLE ENFERMEDAD_X_ANIMAL (
    ID_EA INT PRIMARY KEY,
    Fch_Enfermedad DATE,
    Fch_Curacion DATE,
    Efectivo BIT,
    Cod_Enfermedad INT,
    Cod_Animal INT,
    FOREIGN KEY (Cod_Enfermedad) REFERENCES ENFERMEDAD(Cod_Enfermedad),
    FOREIGN KEY (Cod_Animal) REFERENCES ANIMAL(Cod_Animal)
);
GO
```

La tabla "ENFERMEDAD_X_ANIMAL" establece relaciones entre enfermedades y animales, registrando fechas de enfermedad, curación y la efectividad del tratamiento.

3.3.7. Creación de la Tabla MEDICAMENTO

```
-- Tabla MEDICAMENTO
CREATE TABLE MEDICAMENTO (
    Cod_Medicamento INT PRIMARY KEY,
    Nom_Medicamento VARCHAR(50),
    Fecha_Vencimiento DATE
);
GO
```

La tabla "MEDICAMENTO" contiene información sobre medicamentos utilizados en el tratamiento del ganado, con un código único, nombre y fecha de vencimiento.

3.3.8. Creación de la Tabla MEDICAMENTO_X_ENFERMEDAD

```
-- Tabla MEDICAMENTO_X_ENFERMEDAD
CREATE TABLE MEDICAMENTO_X_ENFERMEDAD (
    Cod_Medicamento INT,
    Cod_Enfermedad INT,
    PRIMARY KEY (Cod_Medicamento, Cod_Enfermedad),
    FOREIGN KEY (Cod_Medicamento) REFERENCES MEDICAMENTO(Cod_Medicamento),
    FOREIGN KEY (Cod_Enfermedad) REFERENCES ENFERMEDAD(Cod_Enfermedad)
);
GO
```

La tabla "MEDICAMENTO_X_ENFERMEDAD" establece relaciones entre medicamentos y enfermedades, permitiendo un registro detallado de los tratamientos aplicados.

3.3.9. Creación de la Tabla MEDICAMENTO_X_ANIMAL

```
-- Tabla MEDICAMENTO_X_ANIMAL
CREATE TABLE MEDICAMENTO_X_ANIMAL (
    ID_MA INT PRIMARY KEY,
    Cod_Animal INT,
    Cod_Medicamento INT,
    Fecha_Aplicacion DATE,
    FOREIGN KEY (Cod_Animal) REFERENCES ANIMAL(Cod_Animal),
    FOREIGN KEY (Cod_Medicamento) REFERENCES MEDICAMENTO(Cod_Medicamento)
);
GO
```

La tabla "MEDICAMENTO_X_ANIMAL" establece relaciones entre medicamentos y animales, registrando fechas de aplicación y detalles específicos de los tratamientos.

3.4. Carga Inicial de Datos en la Base de Datos

Una vez establecida la estructura de la base de datos, se procederá a la carga inicial de datos, incorporando información relevante al sector agropecuario del Departamento del Beni, específicamente en la esfera ganadera bovina. Este proceso, de naturaleza crítica, requerirá una atención meticulosa para garantizar la coherencia, integridad y calidad de los datos que constituirán la columna vertebral del sistema de gestión.

Proceso de Inserción de Datos para el Proyecto de Base de Datos:

3.4.1. Inserción de Datos en la Tabla "ESTANCIA":

```
INSERT INTO ESTANCIA (Cod_Estancia, Nombre) VALUES (1, 'Nombre_Estancia_1');
INSERT INTO ESTANCIA (Cod_Estancia, Nombre) VALUES (2, 'Nombre_Estancia_2');
```

3.4.2. Inserción de Datos en la Tabla "RAZA":

```
INSERT INTO RAZA (Cod_Raza, Nom_Raza) VALUES (1, 'Criollo');
INSERT INTO RAZA (Cod_Raza, Nom_Raza) VALUES (2, 'Nelore');
```

3.4.3. Inserción de Datos en la Tabla "ENFERMEDAD":

```
INSERT INTO ENFERMEDAD (Cod_Enfermedad, Nom_Enfermedad, Sintomas) VALUES
(1, 'Fiebre aftosa', 'Ampollas en la boca y pezuñas');
```

3.3.4. Inserción de Datos en la Tabla "ANIMAL":

```
INSERT INTO ANIMAL (Cod_Animal, Sexo, Fecha_Nac, Peso, Cod_Estancia, Cod_Raza)
VALUES (1, 'M', '2022-01-01', 500.75, 1, 1);
```

3.4.5. Inserción de Datos en la Tabla "ENFERMEDAD_X_ANIMAL":

```
INSERT INTO ENFERMEDAD_X_ANIMAL (ID_EA, Fch_Enfermedad, Fch_Curacion,
Efectivo, Cod_Enfermedad, Cod_Animal) VALUES (1, '2023-02-01', '2023-02-10', 1, 1, 1);
```

3.4.6. Inserción de Datos en la Tabla "MEDICAMENTO":

```
INSERT INTO MEDICAMENTO (Cod_Medicamento, Nom_Medicamento,
Fecha_Vencimiento) VALUES (28, 'Vacuna Aftogen', '2024-07-31');
-- Continuar con los demás medicamentos
```

3.4.7. Inserción de Datos en la Tabla "MEDICAMENTO_X_ENFERMEDAD":

```
INSERT INTO MEDICAMENTO_X_ENFERMEDAD (Cod_Medicamento,
Cod_Enfermedad) VALUES (28, 1);
```

3.4.8. Inserción de Datos en la Tabla "MEDICAMENTO_X_ANIMAL":

```
INSERT INTO MEDICAMENTO_X_ANIMAL (ID_MA, Cod_Animal, Cod_Medicamento,
Fecha_Aplicacion) VALUES (1, 1, 28, '2024-03-01');
```

IV. Consultas SQL para manipulación y obtención de información

En el contexto del proyecto "Diseño e Implementación de una Base de Datos para Optimizar la Gestión del Sector Agropecuario en el Departamento del Beni, Bolivia, específicamente en el área ganadera bovina", se presentan a continuación una serie de consultas SQL destinadas a la manipulación y obtención de información pertinente.

4.1. 5 consultas con una tabla

-----I 5 consultas con una tabla-----

--- 1 Devuelve un listado con todos los animales ordenados por fecha de nacimiento, mostrando primero los más recientes.

```
SELECT * FROM ANIMAL
ORDER BY Fecha_Nac DESC;
```

--- 2 Devuelve un listado con los identificadores de las enfermedades -- que han afectado a algún animal. Ten en cuenta que no deben mostrarse identificadores repetidos.

```
SELECT DISTINCT Cod_Enfermedad FROM ENFERMEDAD_X_ANIMAL;
```

--- 3 Devuelve un listado de todos los animales que hayan tenido alguna enfermedad en el año 2023, cuyo peso sea superior a 600.

```
SELECT * FROM ANIMAL
WHERE Cod_Animal IN (SELECT Cod_Animal FROM ENFERMEDAD_X_ANIMAL WHERE
YEAR(Fch_Enfermedad) = 2023)
AND Peso > 600;
```

--- 4 devuelve un listado DE los animales
-- que han recibido algún medicamento con fecha de aplicación entre '2023-01-01' y '2024-01-01'

```
SELECT ANIMAL.Cod_Animal, ANIMAL.Sexo, ANIMAL.Fecha_Nac, ANIMAL.Peso,
ANIMAL.Cod_Estancia, ANIMAL.Cod_Raza
FROM ANIMAL
JOIN MEDICAMENTO_X_ANIMAL ON ANIMAL.Cod_Animal = MEDICAMENTO_X_ANIMAL.Cod_Animal
WHERE Fecha_Aplicacion BETWEEN '2023-01-01' AND '2024-01-01';
```

--- 5 Devuelve todos los datos de los animales de mayor peso.

```
SELECT TOP 2 * FROM ANIMAL
ORDER BY Peso DESC;
```

4.2. 5 Consultas multitaslas composición interna

----- II 5 consultas multitaslas composición interna-----

-- 1 Mostrar el código de estancia y el nombre de todas las estancias junto con el código y nombre de la raza de los animales que se encuentran en cada una de ellas.

```
SELECT ESTANCIA.Cod_Estancia, ESTANCIA.Nombre, ANIMAL.Cod_Raza, RAZA.Nom_Raza
FROM ESTANCIA
JOIN ANIMAL ON ESTANCIA.Cod_Estancia = ANIMAL.Cod_Estancia
JOIN RAZA ON ANIMAL.Cod_Raza = RAZA.Cod_Raza;
```

-- 2 Obtener el código y nombre de todas las enfermedades junto con el código de animal y nombre de la estancia donde se encuentran los animales afectados.

```
SELECT ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad,
ENFERMEDAD_X_ANIMAL.Cod_Animal, ESTANCIA.Nombre
FROM ENFERMEDAD
JOIN ENFERMEDAD_X_ANIMAL ON ENFERMEDAD.Cod_Enfermedad =
ENFERMEDAD_X_ANIMAL.Cod_Enfermedad
JOIN ANIMAL ON ENFERMEDAD_X_ANIMAL.Cod_Animal = ANIMAL.Cod_Animal
JOIN ESTANCIA ON ANIMAL.Cod_Estancia = ESTANCIA.Cod_Estancia;
```

-- 3 Listar el código y nombre de los medicamentos, así como el código y nombre de las enfermedades asociadas a cada medicamento.

```

SELECT MEDICAMENTO.Cod_Medicamento, MEDICAMENTO.Nom_Medicamento,
MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad
FROM MEDICAMENTO
JOIN MEDICAMENTO_X_ENFERMEDAD ON MEDICAMENTO.Cod_Medicamento =
MEDICAMENTO_X_ENFERMEDAD.Cod_Medicamento
JOIN ENFERMEDAD ON MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad = ENFERMEDAD.Cod_Enfermedad;

```

-- 4 Mostrar el código y nombre de las estancias que tienen animales que han sufrido enfermedades, junto con el código y nombre de las enfermedades que han afectado a esos animales.

```

SELECT DISTINCT ESTANCIA.Cod_Estancia, ESTANCIA.Nombre,
ENFERMEDAD_X_ANIMAL.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad
FROM ESTANCIA
JOIN ANIMAL ON ESTANCIA.Cod_Estancia = ANIMAL.Cod_Estancia
JOIN ENFERMEDAD_X_ANIMAL ON ANIMAL.Cod_Animal = ENFERMEDAD_X_ANIMAL.Cod_Animal
JOIN ENFERMEDAD ON ENFERMEDAD_X_ANIMAL.Cod_Enfermedad = ENFERMEDAD.Cod_Enfermedad;

```

--5 Obtener el código y nombre de los medicamentos, así como el código de animales a los que se les ha aplicado cada medicamento.

```

SELECT MEDICAMENTO.Cod_Medicamento, MEDICAMENTO.Nom_Medicamento,
MEDICAMENTO_X_ANIMAL.Cod_Animal
FROM MEDICAMENTO
JOIN MEDICAMENTO_X_ANIMAL ON MEDICAMENTO.Cod_Medicamento =
MEDICAMENTO_X_ANIMAL.Cod_Medicamento
JOIN ANIMAL ON MEDICAMENTO_X_ANIMAL.Cod_Animal = ANIMAL.Cod_Animal;

```

4.3. 5 Consultas multitaslas composición externa

----- III Consultas multitaslas con composición externa: -----

--Mostrar todas las estancias y, si hay animales en ellas, mostrar el código y nombre de la raza de los animales.

```

SELECT ESTANCIA.Cod_Estancia, ESTANCIA.Nombre, ANIMAL.Cod_Raza, RAZA.Nom_Raza
FROM ESTANCIA
LEFT JOIN ANIMAL ON ESTANCIA.Cod_Estancia = ANIMAL.Cod_Estancia
LEFT JOIN RAZA ON ANIMAL.Cod_Raza = RAZA.Cod_Raza;

```

--Obtener todas las enfermedades y, si hay animales afectados por ellas, mostrar el código de los animales y junto con el código y nombre de la estancia en la que se encuentran.

```

SELECT ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad,
ENFERMEDAD_X_ANIMAL.Cod_Animal, ESTANCIA.Cod_Estancia, ESTANCIA.Nombre AS
NombreEstancia
FROM ENFERMEDAD
LEFT JOIN ENFERMEDAD_X_ANIMAL ON ENFERMEDAD.Cod_Enfermedad =
ENFERMEDAD_X_ANIMAL.Cod_Enfermedad

```

```

LEFT JOIN ANIMAL ON ENFERMEDAD_X_ANIMAL.Cod_Animal = ANIMAL.Cod_Animal
LEFT JOIN ESTANCIA ON ANIMAL.Cod_Estancia = ESTANCIA.Cod_Estancia;

```

--Listar todos los medicamentos y, si hay enfermedades asociadas a ellos, mostrar el código y nombre de las enfermedades.

```

SELECT MEDICAMENTO.Cod_Medicamento, MEDICAMENTO.Nom_Medicamento,
MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad
FROM MEDICAMENTO
LEFT JOIN MEDICAMENTO_X_ENFERMEDAD ON MEDICAMENTO.Cod_Medicamento =
MEDICAMENTO_X_ENFERMEDAD.Cod_Medicamento
LEFT JOIN ENFERMEDAD ON MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad =
ENFERMEDAD.Cod_Enfermedad;

```

--Mostrar todas las estancias que tienen animales que han sufrido enfermedades, junto con el código y nombre de las enfermedades que han afectado a esos animales. Incluir también las estancias que no tienen animales.

```

SELECT MEDICAMENTO.Cod_Medicamento, MEDICAMENTO.Nom_Medicamento,
MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad
FROM MEDICAMENTO
LEFT JOIN MEDICAMENTO_X_ENFERMEDAD ON MEDICAMENTO.Cod_Medicamento =
MEDICAMENTO_X_ENFERMEDAD.Cod_Medicamento
LEFT JOIN ENFERMEDAD ON MEDICAMENTO_X_ENFERMEDAD.Cod_Enfermedad =
ENFERMEDAD.Cod_Enfermedad;

```

--Obtener todos los medicamentos y, si hay animales a los que se les ha aplicado cada medicamento, mostrar el código de los animales.

```

SELECT MEDICAMENTO.Cod_Medicamento, MEDICAMENTO.Nom_Medicamento,
MEDICAMENTO_X_ANIMAL.Cod_Animal
FROM MEDICAMENTO
LEFT JOIN MEDICAMENTO_X_ANIMAL ON MEDICAMENTO.Cod_Medicamento =
MEDICAMENTO_X_ANIMAL.Cod_Medicamento
LEFT JOIN ANIMAL ON MEDICAMENTO_X_ANIMAL.Cod_Animal = ANIMAL.Cod_Animal;

```

4.4. 5 Consultas con anidadas o subconsultas

----- IV Consultas con subconsultas o anidadas-----

--Mostrar los nombres de todas las enfermedades que han afectado a animales que se encuentran en la estancia con código 'San Mateo' 'Cod_Estancia'.

```

SELECT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM ENFERMEDAD_X_ANIMAL
    WHERE Cod_Animal IN (
        SELECT Cod_Animal
        FROM ANIMAL
        WHERE Cod_Estancia = '1'
    )
);

```

--Obtener los nombres de todas las enfermedades que han afectado a animales de sexo 'H' y que nacieron antes de la fecha '2023-12-15'.

```
SELECT DISTINCT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM ENFERMEDAD_X_ANIMAL
    WHERE Cod_Animal IN (
        SELECT Cod_Animal
        FROM ANIMAL
        WHERE Sexo = 'H' AND Fecha_Nac < '2023-12-15'
    )
);
```

--Listar los nombres de todas las enfermedades que han afectado a animales con un peso superior al peso promedio de todos los animales.

```
SELECT DISTINCT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM ENFERMEDAD_X_ANIMAL
    WHERE Cod_Animal IN (
        SELECT Cod_Animal
        FROM ANIMAL
        WHERE Peso > (SELECT AVG(Peso) FROM ANIMAL)
    )
);
```

--Mostrar los nombres de todas las enfermedades que han afectado a animales que han sido tratados con el medicamento 'Vacuna Aftogen' de código 'Cod_Medicamento'.

```
SELECT DISTINCT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM MEDICAMENTO_X_ENFERMEDAD
    WHERE Cod_Medicamento = '28'
);
```

--Obtener los nombres de todas las enfermedades que han afectado a animales que se encuentran en la estancia con código 'Cod_Estancia' y que han sido tratados con el medicamento de código 'Cod_Medicamento'.

```
SELECT DISTINCT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM ENFERMEDAD_X_ANIMAL
    WHERE Cod_Animal IN (
        SELECT Cod_Animal
        FROM ANIMAL
    )
);
```



```

        WHERE Cod_Estancia = '1'
    )
    AND Cod_Enfermedad IN (
        SELECT Cod_Enfermedad
        FROM MEDICAMENTO_X_ENFERMEDAD
        WHERE Cod_Medicamento = '31'
    )
);

```

4.5. 5 Consultas con operadores de conjuntos

----- V Consultas con operadores de conjuntos-----

-- 1 Mostrar el código y nombre de todas las estancias que tienen animales y el código y nombre de todas las estancias que no tienen animales.

```

-- Estancias con animales
SELECT Cod_Estancia, Nombre
FROM ESTANCIA
WHERE Cod_Estancia IN (SELECT Cod_Estancia FROM ANIMAL);

```

```

-- Estancias sin animales
SELECT Cod_Estancia, Nombre
FROM ESTANCIA
WHERE Cod_Estancia NOT IN (SELECT Cod_Estancia FROM ANIMAL);

```

-- 2 Obtener el código y nombre de todas las enfermedades que han afectado a animales y el código y nombre de todas las enfermedades que no han afectado a ningún animal.

```

-- Enfermedades que han afectado a animales
SELECT Cod_Enfermedad, Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (SELECT Cod_Enfermedad FROM ENFERMEDAD_X_ANIMAL);

```

```

-- Enfermedades que no han afectado a ningún animal
SELECT Cod_Enfermedad, Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad NOT IN (SELECT Cod_Enfermedad FROM ENFERMEDAD_X_ANIMAL);

```

-- 3 Listar el código y nombre de todos los medicamentos que han sido aplicados a animales y el código y nombre de todos los medicamentos que no han sido aplicados a ningún animal.

```

-- Medicamentos aplicados a animales
SELECT Cod_Medicamento, Nom_Medicamento
FROM MEDICAMENTO
WHERE Cod_Medicamento IN (SELECT Cod_Medicamento FROM MEDICAMENTO_X_ENFERMEDAD);

```

```

-- Medicamentos que no han sido aplicados a ningún animal
SELECT Cod_Medicamento, Nom_Medicamento
FROM MEDICAMENTO
WHERE Cod_Medicamento NOT IN (SELECT Cod_Medicamento FROM MEDICAMENTO_X_ENFERMEDAD);

```

-- 4 Mostrar el código y nombre de todas las estancias que tienen animales que han sufrido enfermedades y el código y nombre de todas las estancias que no tienen animales o cuyos animales no han sufrido enfermedades.

```
-- Estancias con animales que han sufrido enfermedades
SELECT Cod_Estancia, Nombre
FROM ESTANCIA
WHERE Cod_Estancia IN (
    SELECT Cod_Estancia
    FROM ANIMAL
    WHERE Cod_Animal IN (SELECT Cod_Animal FROM ENFERMEDAD_X_ANIMAL)
);
```

```
-- Estancias sin animales o cuyos animales no han sufrido enfermedades
SELECT Cod_Estancia, Nombre
FROM ESTANCIA
WHERE Cod_Estancia NOT IN (
    SELECT Cod_Estancia
    FROM ANIMAL
    WHERE Cod_Animal IN (SELECT Cod_Animal FROM ENFERMEDAD_X_ANIMAL)
) OR Cod_Estancia NOT IN (SELECT Cod_Estancia FROM ANIMAL);
```

-- 5 Obtener el código y nombre de todos los medicamentos que han sido aplicados a animales y el código y nombre de todos los medicamentos que no han sido aplicados a ningún animal, excepto aquellos que han vencido antes de la fecha actual.

```
-- Medicamentos aplicados a animales
SELECT Cod_Medicamento, Nom_Medicamento
FROM MEDICAMENTO
WHERE Cod_Medicamento IN (SELECT Cod_Medicamento FROM MEDICAMENTO_X_ENFERMEDAD);

-- Medicamentos que no han sido aplicados a ningún animal y no han vencido antes de la fecha actual
SELECT Cod_Medicamento, Nom_Medicamento
FROM MEDICAMENTO
WHERE Cod_Medicamento NOT IN (SELECT Cod_Medicamento FROM MEDICAMENTO_X_ENFERMEDAD)
AND Fecha_Vencimiento > GETDATE();
```

4.6. 5 Consultas con funciones agregadas

----- VI Consultas con funciones agregadas-----

-- 1 Mostrar el número total de animales por estancia junto con el código y nombre de cada estancia.

```
SELECT E.Cod_Estancia, E.Nombre AS Nombre_Estancia, COUNT(A.Cod_Animal) AS
Total_Animales
FROM ESTANCIA E
LEFT JOIN ANIMAL A ON E.Cod_Estancia = A.Cod_Estancia
GROUP BY E.Cod_Estancia, E.Nombre;
```

-- 2 Obtener la edad promedio de los animales por raza junto con el código y nombre de cada raza.

```

SELECT R.Cod_Raza, R.Nom_Raza, AVG(DATEDIFF(YEAR, A.Fecha_Nac, GETDATE())) AS
Edad_Promedio
FROM RAZA R
JOIN ANIMAL A ON R.Cod_Raza = A.Cod_Raza
GROUP BY R.Cod_Raza, R.Nom_Raza;

```

-- 3 Listar la cantidad total de enfermedades por animal junto con el código y sexo de cada animal.

```

SELECT A.Cod_Animal, A.Sexo, COUNT(EXA.Cod_Enfermedad) AS Total_Enfermedades
FROM ANIMAL A
LEFT JOIN ENFERMEDAD_X_ANIMAL EXA ON A.Cod_Animal = EXA.Cod_Animal
GROUP BY A.Cod_Animal, A.Sexo;

```

-- 4 Mostrar la fecha mínima y máxima de aplicación de medicamentos por animal junto con el código y sexo de cada animal.

```

SELECT A.Cod_Animal, A.Sexo, MIN(MXA.Fecha_Aplicacion) AS Min_Fecha,
MAX(MXA.Fecha_Aplicacion) AS Max_Fecha
FROM ANIMAL A
LEFT JOIN MEDICAMENTO_X_ANIMAL MXA ON A.Cod_Animal = MXA.Cod_Animal
GROUP BY A.Cod_Animal, A.Sexo;

```

-- 5 Obtener el peso promedio de los animales por estancia junto con el código y nombre de cada estancia.

```

SELECT E.Cod_Estancia, E.Nombre AS Nombre_Estancia, AVG(A.Peso) AS Peso_Promedio
FROM ESTANCIA E
JOIN ANIMAL A ON E.Cod_Estancia = A.Cod_Estancia
GROUP BY E.Cod_Estancia, E.Nombre;

```

V. Pruebas y validación del sistema

5.1. Realización de pruebas

--- 4 devuelve un listado DE los animales
 -- que han recibido algún medicamento con fecha de aplicación entre '2023-01-01' y '2024-01-01'

```

SELECT ANIMAL.Cod_Animal, ANIMAL.Sexo, ANIMAL.Fecha_Nac, ANIMAL.Peso, ANIMAL.Cod_Estancia, ANIMAL.Cod_Raza
FROM ANIMAL
JOIN MEDICAMENTO_X_ANIMAL ON ANIMAL.Cod_Animal = MEDICAMENTO_X_ANIMAL.Cod_Animal
WHERE Fecha_Aplicacion BETWEEN '2023-01-01' AND '2024-01-01';

```

100 % <

Resultados Mensajes

	Cod_Animal	Sexo	Fecha_Nac	Peso	Cod_Estancia	Cod_Raza
1	5	M	2022-05-05	700.00	1	5
2	2	H	2022-02-15	450.50	2	2
3	6	H	2022-06-15	480.75	2	6
4	9	M	2022-09-15	610.75	1	3

```
SELECT ESTANCIA.Cod_Estancia, ESTANCIA.Nombre, ANIMAL.Cod_Raza, RAZA.Nom_Raza
FROM ESTANCIA
JOIN ANIMAL ON ESTANCIA.Cod_Estancia = ANIMAL.Cod_Estancia
JOIN RAZA ON ANIMAL.Cod_Raza = RAZA.Cod_Raza;
```

10 % <

Resultados Mensajes

	Cod_Estancia	Nombre	Cod_Raza	Nom_Raza
4	2	Santa Maria	4	Holstein
5	1	San Mateo	5	Pardo Suizo
3	2	Santa Maria	6	Brahman
7	1	San Mateo	1	Criollo
3	2	Santa Maria	2	Nelore
3	1	San Mateo	3	Ganado Mestizo
10	2	Santa Maria	4	Holstein
11	1	San Mateo	5	Pardo Suizo
12	2	Santa Maria	6	Brahman
13	1	San Mateo	1	Criollo

```

SELECT ENFERMEDAD.Cod_Enfermedad, ENFERMEDAD.Nom_Enfermedad, ENFERMEDAD_X_ANIMAL.Cod_Animal,
ESTANCIA.Cod_Estancia, ESTANCIA.Nombre AS NombreEstancia
FROM ENFERMEDAD
LEFT JOIN ENFERMEDAD_X_ANIMAL ON ENFERMEDAD.Cod_Enfermedad = ENFERMEDAD_X_ANIMAL.Cod_Enfermedad
LEFT JOIN ANIMAL ON ENFERMEDAD_X_ANIMAL.Cod_Animal = ANIMAL.Cod_Animal
LEFT JOIN ESTANCIA ON ANIMAL.Cod_Estancia = ESTANCIA.Cod_Estancia;

```

00 %

Resultados Mensajes

	Cod_Enfermedad	Nom_Enfermedad	Cod_Animal	Cod_Estancia	NombreEstancia
1	1	Fiebre aftosa	1	1	San Mateo
2	1	Fiebre aftosa	2	2	Santa Maria
3	1	Fiebre aftosa	7	1	San Mateo
4	1	Fiebre aftosa	8	2	Santa Maria
5	1	Fiebre aftosa	14	2	Santa Maria
6	1	Fiebre aftosa	20	2	Santa Maria
7	1	Fiebre aftosa	26	2	Santa Maria
8	2	Brucelosis	3	1	San Mateo
9	2	Brucelosis	4	2	Santa Maria
10	2	Brucelosis	9	1	San Mateo

```

SELECT DISTINCT Nom_Enfermedad
FROM ENFERMEDAD
WHERE Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM ENFERMEDAD_X_ANIMAL
    WHERE Cod_Animal IN (
        SELECT Cod_Animal
        FROM ANIMAL
        WHERE Cod_Estancia = '1'
    )
)
AND Cod_Enfermedad IN (
    SELECT Cod_Enfermedad
    FROM MEDICAMENTO_X_ENFERMEDAD
    WHERE Cod_Medicamento = '31'
);

```

0 %

Resultados Mensajes

Nom_Enfermedad
Leptospirosis

VI. Conclusiones y recomendaciones

La conclusión del proyecto es que se ha desarrollado una solución integral basada en una base de datos relacional para mejorar la gestión del sector agropecuario en el Departamento del Beni, Bolivia. Esta solución aborda los desafíos existentes en la gestión eficiente del ganado bovino, proporcionando herramientas para optimizar las operaciones relacionadas con la salud y el seguimiento de cada bovino en la región.

La implementación de esta solución brinda beneficios significativos para el sector ganadero, ya que facilita la toma de decisiones informadas, optimiza el seguimiento de la salud de cada animal y maximiza la productividad en el sector agropecuario. Al contar con un sistema centralizado y eficiente, los productores y profesionales del sector pueden acceder a información relevante y tomar medidas preventivas para prevenir enfermedades y mantener la salud del ganado.

El diseño de la base de datos relacional y la implementación de consultas SQL permiten obtener información valiosa sobre las estancias, razas, animales, enfermedades y medicamentos relacionados con el ganado bovino en el Departamento del Beni. Estos datos ayudan en la toma de decisiones, la identificación de patrones y la aplicación de tratamientos adecuados.

VII. Referencias bibliográficas

Bibliografía

ED_TEAM. (2023). *ed.team*. Obtenido de

<https://ed.team/blog/normalizacion-de-bases-de-datos>

Oscar Durán, m. v. (2021). *veterquimica*. Obtenido de [veterquimica.com](https://www.veterquimica.com):

<https://www.veterquimica.cl/el-potencial-de-la-ganaderia-de-bolivia-una-genetica-para-el-mundo/>

<https://totalpec.com/blog/162/vacune-a-su-ganado-bovino-para-prevenir-enfermedades-reproductivas>

