

Entrega del Proyecto Final

Realizado por Daniela Soler Collao
Curso SQL | Coderhouse

Introducción

En este proyecto se plantea modelar una base de datos para una veterinaria, con el fin de facilitar el manejo interno de esta, mediante el almacenamiento de la siguiente información:

- Ficha informativa de la mascota
- Datos del dueño de la mascota
- Carné médico de antiparasitarios y vacunas
- Agenda de consultas médicas

Esto es con el objetivo de poder mandar un mail de recordatorio de los próximos pendientes de la mascota, ya sea su próxima antiparasitación, vacunación o consulta. A su vez, es importante que el personal veterinario pueda tener fácil acceso a la información de la mascota y su historial de consultas para brindar una atención más completa y consistente.

Esta base de datos puede ser utilizada para posteriormente crear un software de veterinaria, facilitando el acceso de la información tanto a los veterinarios como a sus clientes (dueños de las mascotas).

Ejecución de los Scripts

La herramienta utilizada para realizar la base de datos es MySQL Workbench, mediante el sistema de gestión MySQL.

En el GitHub, se encuentran los archivos con formato .sql de la información descrita y trabajada con diferentes consultas y subconsultas relacionadas al tópico de la base de datos (<https://github.com/Daniela1221/EntregaFinalSQL-SolerCollao>).

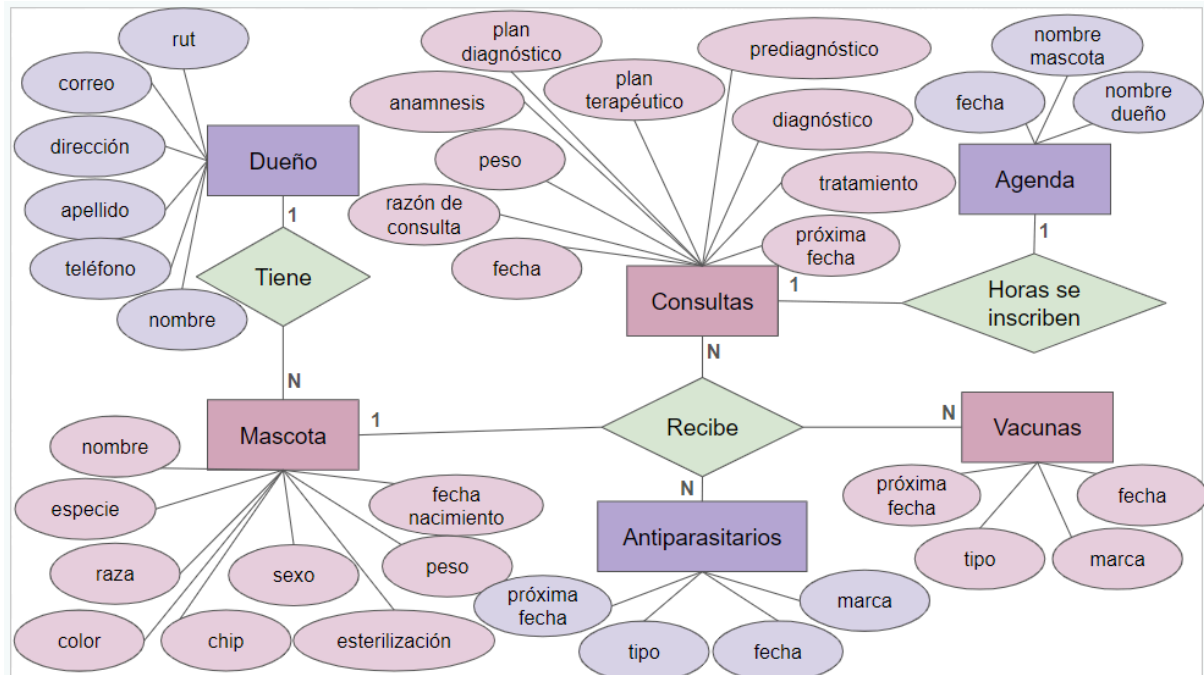
Los scripts utilizados para generar la base de datos son 4, los cuales deben ejecutarse en el siguiente orden:

- Primero, ejecutar el script '1-Tablas.sql', el cual crea el schema 'veterinaria' a trabajar y las tablas asociadas al proyecto.
- Segundo, ejecutar el script '2-Funciones_Triggers_StoredProcedures.sql', el cual crea las funciones, triggers y stored procedures a ocupar en el schema de veterinaria.
- Tercero, ejecutar el script '3-Registros.sql', el cual realiza la inserción manual de registros a la base de datos asociada al schema de veterinaria. Los datos no contienen información de clientes reales, ya que el fin es utilizar esta información a modo de ejemplo.
- Por último, ejecutar el script '4-Consultas_Subconsultas_Vistas.sql', el cual realiza las consultas necesarias que necesita la veterinaria para facilitar el manejo de esta.

Estas se encuentran descritas dentro del script, junto con una descripción de lo que facilita cada consulta, subconsulta y vista realizada dentro de la base de datos asociada al schema de la veterinaria.

Diagrama de Entidad - Relación

Para realizar la base de datos, se presenta el siguiente Diagrama de Entidad-Relación, para tener en consideración los campos y variables que se manejan en la base de datos.



En base al Diagrama, se realizan e implementan las tablas descritas a continuación.

Tablas

1. Dueño

Esta tabla cumple la función de almacenar los datos de los dueños de las mascotas que ingresan a la veterinaria, con el fin de contactar al dueño frente a cualquier situación que se presente con su mascota, ya sea recordatorios o emergencias.

Columnas	Tipo de dato	Tipo de clave
id	int	clave primaria
nombre	varchar(100)	-
apellido	varchar(100)	-
rut	varchar(20)	clave candidata
telefono	varchar(15)	clave candidata
direccion	varchar(100)	-
correo	varchar(100)	clave candidata

2. Historial dueños

Esta tabla cumple el objetivo de respaldar la información de la tabla Dueño, ya que es información fundamental para poder contactar al dueño para cualquier situación que se pueda generar.

Columnas	Tipo de dato	Tipo de clave
id	int	clave primaria
nombre	varchar(100)	-
apellido	varchar(100)	-
rut	varchar(100)	clave candidata
telefono	varchar(100)	clave candidata
direccion	varchar(100)	-
correo	varchar(100)	clave candidata

3. Mascota

Esta tabla cumple el objetivo de almacenar los datos asociados a una mascota, asociando esta tabla con la tabla Dueño. Esta se realiza con el fin de implementar la ficha informativa de la mascota.

Columnas	Tipo de dato	Tipo de clave
id	int	clave primaria
dueño_id	int	clave foránea
nombre	varchar(100)	-
especie	varchar(100)	-
raza	varchar(100)	-
color	varchar(100)	-
sexo	varchar(100)	-
fecha_nac	date	-
esterilizado	boolean	-
chip	boolean	-
peso	float	-

4. Historial mascotas

Esta tabla cumple el objetivo de respaldar la información de la tabla Mascota, ya que es información fundamental para hacer el seguimiento a la mascota y asociar su ficha con los antiparasitarios, vacunas y consultas que esta tenga asociada; facilitando el seguimiento de su historial médico.

Columnas	Tipo de dato	Tipo de clave
id	int	clave primaria
dueño_id	int	-
nombre	varchar(100)	-
especie	varchar(100)	-
raza	varchar(100)	-
color	varchar(100)	-
sexo	varchar(100)	-

fecha_nac	date	-
esterilizado	boolean	-
chip	boolean	-
peso	float	-

5. Antiparasitarios

Esta tabla cumple la función de almacenar los datos asociados a la antiparasitación de una mascota, con el fin de generar recordatorios al tener que aplicarse con determinada periodicidad. Esta contiene relación con las tablas de marca antiparasitario, tipo antiparasitario y mascota, con el fin de optimizar y organizar la base de datos.

Columnas	Tipo de dato	Tipo de clave
antiparasitario_id	int	clave primaria
tipo	int	clave foránea
marca	int	clave foránea
fecha	date	-
proxima_fecha	date	-
mascota_id	int	clave foránea

6. Marca antiparasitario

Esta tabla cumple la función de almacenar las marcas asociadas a los antiparasitarios, facilitando el manejo de información y optimizando la eficiencia de la base de datos.

Columnas	Tipo de dato	Tipo de clave
marca_id	int	clave primaria
nombre	varchar(20)	clave candidata

7. Tipo antiparasitario

Esta tabla cumple la función de almacenar los tipos asociados a los antiparasitarios, facilitando el manejo de información y optimizando la eficiencia de la base de datos.

Columnas	Tipo de dato	Tipo de clave
tipo_id	int	clave primaria
nombre	varchar(20)	clave candidata

8. Historial antiparasitarios

Esta tabla cumple la función de generar un respaldo de la información almacenada en la base de datos con respecto a los antiparasitarios, con el fin de no perder el historial médico de la mascota.

Columnas	Tipo de dato	Tipo de clave
antiparasitario_id	int	clave primaria
tipo	int	-
marca	int	-
fecha	date	-
proxima_fecha	date	-
mascota_id	int	-

9. Vacunas

Esta tabla cumple la función de registrar las características de las vacunas administradas a una mascota, teniendo relación con la tabla de mascotas para asociar la vacunación a una mascota en particular. También tiene una relación con las tablas marca vacunas y tipo vacunas para facilitar la optimización de la base de datos.

Columnas	Tipo de dato	Tipo de clave
vacuna_id	int	clave primaria
tipo	int	clave foránea
marca	int	clave foránea
fecha	date	-
fecha_proxima	date	-

mascota_id	int	clave foránea
------------	-----	---------------

10. *Marca vacuna*

Esta tabla cumple la función de facilitar la optimización de la base de datos, almacenando los nombres de las vacunas.

Columnas	Tipo de dato	Tipo de clave
marca_id	int	clave primaria
nombre	varchar(20)	clave candidata

11. *Tipo vacuna*

Esta tabla cumple la función de facilitar la optimización de la base de datos, almacenando los tipos de vacunas.

Columnas	Tipo de dato	Tipo de clave
tipo_id	int	clave primaria
nombre	varchar(20)	clave candidata

12. *Historial vacunas*

Esta tabla cumple la función de generar un respaldo de la información almacenada en la base de datos con respecto a las vacunas, con el fin de no perder el historial médico de la mascota.

Columnas	Tipo de dato	Tipo de clave
vacuna_id	int	clave primaria
tipo	int	-
marca	int	-
fecha	date	-
fecha_proxima	date	-
mascota_id	int	-

13. Agenda

Esta tabla cumple la función de almacenar la agenda de las consultas de las mascotas que toman una hora en la veterinaria para un día en particular, facilitando la organización de las consultas.

Columnas	Tipo de dato	Tipo de clave
id	int	clave primaria
fecha	date	-
nombre_dueño	varchar(100)	-
nombre_mascota	varchar(100)	-

14. Consulta

Esta tabla cumple la función de almacenar las consultas de una mascota en particular, por lo que se encuentra relacionada con la tabla de mascotas. También se encuentra relacionada con la tabla agenda, ya que el día de la consulta realizada se agenda previamente en la tabla de agenda.

Columnas	Tipo de dato	Tipo de clave
consulta_id	int	clave primaria
fecha_id	int	clave foránea
razon_consulta	varchar(200)	-
peso	float	-
anamnesis	varchar(1000)	-
plan_diagnostico	varchar(1000)	-
plan_terapeutico	varchar(1000)	-
prediagnostico	varchar(300)	-
diagnostico	varchar(300)	-
tratamiento	varchar(2000)	-
fecha_proxima	date	-
mascota_id	int	clave foránea

15. Historial consultas

Esta tabla cumple la función de generar un respaldo de la información almacenada en la base de datos con respecto a las consultas, con el fin de no perder el historial médico de la mascota.

Columnas	Tipo de dato	Tipo de clave
consulta_id	int	clave primaria
fecha_id	int	-
razon_consulta	varchar(200)	-
peso	float	-
anamnesis	varchar(1000)	-
plan_diagnostico	varchar(1000)	-
plan_terapeutico	varchar(1000)	-
prediagnostico	varchar(300)	-
diagnostico	varchar(300)	-
tratamiento	varchar(2000)	-
fecha_proxima	date	-
mascota_id	int	-

Además de la creación de tablas y registros insertados de forma manual a modo de ejemplo, se encuentran distintos tipos de cláusulas y sentencias para facilitar el uso de la información, donde las utilizadas son: Vistas, Funciones, Stored Procedures y Triggers. A continuación se detalla cada una con su objetivo, descripción y tablas relacionadas a cada uno.

Vistas

Para este apartado se generaron 8 vistas en la base de datos, con los nombres y objetivos descritos a continuación.

1. *agenda_proxima*

Descripción: Muestra la agenda de las consultas agendadas en los próximos días.

Objetivo: Tener en cuenta las próximas consultas agendadas para facilitar la organización administrativa de las mascotas que necesitan ser atendidas.

Tablas: agenda.

2. *mascotas_por_dueño*

Descripción: Muestra los registros de mascotas que posee un dueño, siendo ordenadas por dueño_id, mostrando todos los campos de la tabla mascota.

Objetivo: Es útil para visualizar la cantidad de mascotas que posee un dueño y ver la manera en que la cantidad de mascotas puede influir en una en particular a la hora de la socialización.

Tablas: mascota.

3. *pendientes_vacunas*

Descripción: Muestra los registros de mascotas a las cuáles se les venció alguna vacuna en particular, con todo el detalle de la tabla vacunas.

Objetivo: Sirve para administrar y recordar al dueño de la mascota para que se ponga al día con el calendario de vacunación de la mascota.

Tablas: vacunas, mascota, dueño.

4. *pendientes_antiparasitarios*

Descripción: Muestra los registros de mascotas a las cuáles se les venció algún antiparasitario en particular, con todo el detalle de la tabla antiparasitarios.

Objetivo: Sirve para administrar y recordar al dueño de la mascota para que se ponga al día con el calendario de antiparasitarios de la mascota.

Tablas: antiparasitarios, mascota, dueño.

5. *mascotas_de_un_dueño*

Descripción: Muestra de manera simple el nombre del dueño y el nombre de la mascota asociada a cada dueño, ordenados por este último.

Objetivo: Implementado a modo de visualización rápida para identificar las mascotas asociadas al dueño, en vez de números de id.

Tablas: dueño, mascota.

6. *numero_mascotas_por_dueño*

Descripción: Muestra el número de mascotas que posee un dueño, identificado por el nombre de este último.

Objetivo: Visualizar de manera rápida la cantidad de mascotas que posee un dueño, para tenerlo en consideración a la hora de realizar una consulta y recomendación de medicamentos.

Tablas: dueño, mascota.

7. *chip_puestos*

Descripción: Muestra el número de mascotas que poseen chip, ordenadas en base al dueño.

Objetivo: Sirve para ver la tendencia del dueño de la mascota y su grado de compromiso que tiene este con su mascota, y en caso de tener otra mascota sin chip, permite ver que se puede dialogar más sencillamente a que este le ponga chip de manera más simple.

Tablas: dueño, mascota.

8. *registro_edad_mascota*

Descripción: Muestra los datos principales de una mascota (nombre, especie, raza, color) con su respectivo cálculo de la edad en base a su fecha de nacimiento respectiva.

Objetivo: Permite analizar qué tipo de vacunas o antiparasitarios necesita la mascota, o que tipo de exámenes recomendar al dueño en caso de ser considerado una mascota de edad avanzada.

Tablas: mascota.

Funciones

Se crearon dos funciones para facilitar el manejo de información de la base de datos, descrito a continuación.

1. *calculo_edad*

Descripción: Mediante una fecha específica, retorna la diferencia en años de entre la fecha otorgada a la función respecto a la fecha actual.

Objetivo: Implementada para realizar el cálculo de edad de una mascota, la cual es utilizada para generar la vista registro_edad_mascota.

Tablas/Datos: Recibe un valor de tipo fecha, para retornar un valor de tipo entero.

2. *nombre_y_apellido*

Descripción: Mediante dos cadenas de texto, las concatena en una cadena de texto, separadas por un espacio vacío.

Objetivo: Se utiliza en varias vistas para concatenar el nombre y apellido de un dueño y facilitar la visualización de los datos.

Tablas/Datos: Recibe dos cadenas de texto y retorna una cadena de texto.

Stored procedures

En este apartado se generaron 4 stored procedures

1. *nombre_mascota_repetido*

Descripción: Muestra las mascotas asociadas a un mismo nombre, con información detallada de la tabla mascota.

Objetivo: Visualizar de manera rápida si hay un nombre de mascota repetido, para asociar el registro que se quiera ingresar respecto a esta, de manera correcta y asociarlo al id indicado, para evitar errores de asignación.

Tabla/Datos: Tabla mascota, ingresando un valor de entrada de una cadena de texto, asociada al nombre de una mascota.

2. *contar_chips*

Descripción: Muestra la cantidad de mascotas que poseen o no poseen chip registradas en la base de datos.

Objetivo: Visualizar de manera rápida la cantidad de mascotas que no poseen chip, con el fin de realizar una futura búsqueda exhaustiva y ver qué mascotas no poseen chip y así poder hablar con el respectivo dueño del tema.

Tablas/Datos: Tabla mascota, ingresando un valor de entrada de tipo entero, asociado al valor del chip.

3. *mascota_con_o_sin_chip*

Descripción: Muestra la información detallada de las mascotas dependiendo si la búsqueda va asociada a si tiene o no chip.

Objetivo: Permite ver la información de las mascotas sin chip, para poder identificarlas y hacer una búsqueda posterior al dueño que está asociada la mascota.

Tablas/Datos: Tabla mascota, ingresando un valor de entrada de tipo entero, asociado al valor del chip.

4. *historial*

Descripción: Muestra un historial generado en base a cualquier tabla que se quiera revisar en base a una mascota en particular.

Objetivo: Visualizar el historial de una mascota para los campos de antiparasitarios y vacunas, y así poder tener un análisis más exhaustivo a la hora de hacer recomendaciones al dueño en futuras consultas.

Tablas/Datos: Tabla vacunas o antiparasitarios, recibiendo como parámetros de entrada el nombre de la tabla a ver el historial como cadena de texto y el nombre el id asociado a una mascota como valor entero.

Triggers

Se realizan 5 triggers a modo de respaldo de información fundamental para la veterinaria. Para esto, se crearon las tablas historial_dueños, historial_mascotas, historial_antiparasitarios, historial_vacunas, historial_consultas.

1. *respaldo_dueño*

Descripción: Al momento de insertar un registro en la tabla dueño, genera una copia de todos los campos asociados a este registro nuevo.

Objetivo: Generar un historial de información de todos los dueños que han pasado por la veterinaria, para tener un mejor seguimiento de estos.

Tablas: dueño, historial_dueños.

2. *respaldo_mascota*

Descripción: Al momento de insertar un registro en la tabla mascota, se genera una copia de todos los campos asociados a este registro nuevo.

Objetivo: Generar un historial de información de todas las mascotas que han pasado por la veterinaria, para tener un mejor seguimiento de estos.

Tablas: mascota, historial_mascotas.

3. respaldo_antiparasitario

Descripción: Al momento de insertar un registro en la tabla antiparasitarios, se genera una copia de todos los campos asociados a este registro nuevo.

Objetivo: Generar un historial de información de todos los antiparasitarios que han pasado por la veterinaria, para tener un mejor seguimiento de estos.

Tablas: antiparasitarios, historial_antiparasitarios.

4. respaldo_vacuna

Descripción: Al momento de insertar un registro en la tabla vacunas, se genera una copia de todos los campos asociados a este registro nuevo.

Objetivo: Generar un historial de información de todas las vacunas que han pasado por la veterinaria, para tener un mejor seguimiento de estas.

Tablas: vacunas, historial_vacunas.

5. respaldo_consulta

Descripción: Al momento de insertar un registro en la tabla consulta, se genera una copia de todos los campos asociados a este registro nuevo.

Objetivo: Generar un historial de información de todas las consultas que han pasado por la veterinaria, para tener un mejor seguimiento de estos.

Tablas: consulta, historial_consultas.

Futuras líneas

- Implementar el formato datetime en la tabla de agenda para facilitar la toma de horas para consultas de las mascotas, con el fin de facilitar la generación de un calendario, lo cual facilita el manejo de este por cualquier usuario que requiera manipular la agenda de consultas.
- Generar una dependencia en la tabla de agenda dependiendo de si el paciente es nuevo o antiguo. Esto es con el fin de optimizar el manejo de información de la veterinaria, generando una dependencia entre las tablas de agenda, dueño y mascota. Es decir, si un paciente es antiguo, relacione el nombre del dueño y el nombre de la mascota como claves foráneas a las tablas dueño y mascota respectivamente, y si el paciente es nuevo, llenar los campos a modo de cadena de texto, para llenar sus datos una vez se realice la consulta.