¡Logotipo

Descripción generada automáticamente

Universidad Técnica Particular de Loja

**Integrantes:**

* Miguel Ángel Caraguay Correa
* Adriana Sofía Jaramillo Ochoa
* Mateo Sebastián Martínez Velázquez

**Tema:**

Proyecto Camaronera

**Grupo:**

2

## Caso de estudio

Se desea implementar una base de datos para llevar el control de la información de las mascotas atendidas en una veterinaria.

Para lo cual se requiere archivar los datos de las mascotas y sus dueños con los tratamientos prescritos de la veterinaria los cuales incluyen una serie de medicamentos.

También se requiere conocer los datos del veterinario dueño de la veterinaria, y sus pasantes.

* La veterinaria y la mascota pueden tener un solo dueño.
* Se debe registrar la frecuencia mensual con la que el cliente visita la veterinaria, si el cliente visita la veterinaria más de 5 veces por mes se aplica un descuento del 10% en su consulta, si el cliente visita 3 o más veces por mes se aplica un descuento del 5% y si su frecuencia es menor a 3 no se aplica descuentos.

Este atributo descuento debe ser calculado dependiendo de la frecuencia mensual del cliente.

* Cada veterinaria va a tener máximo 3 estudiantes haciendo practicas preprofesionales en ella y estos estudiantes deben cursar el noveno y decimo ciclo de universidad.
* Los valores de peso deben ser ingresados en libras.
* Los valoras de tamaño deben ser ingresados en centímetros.
* La edad de la mascota debe registrarse en meses.

|  |
| --- |
| Diseño conceptual |

### Diagrama E/R

Diagrama

Descripción generada automáticamente

### Diccionario de datos

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo Entidad/Relacion** | **Atributo** | **Restriccion de Dominio Adicional** | **Valor por defecto** |
| Cliente | descuento | Los valores solo pueden ser:  5% y 10% |  |
| Estudiante | Ciclo | Solo valores : 9 , 10 |  |
| Tratamiento | fecha\_fin | Debe ser mayor a fecha\_inicio |  |
| Mascota | genero | Solo valores:  Macho “m”  Hembra “h” |  |

### Restricciones de integridad

* + - Los valores de peso deben ser ingresados en libras.
    - Los valoras de tamaño deben ser ingresados en centímetros.
    - La edad de la mascota debe registrase en meses.
    - Este atributo descuento debe ser calculado dependiendo de la frecuencia mensual del cliente.
    - Cada veterinaria solo puede tener 3 estudiantes como pasantes.

|  |
| --- |
| Diseño lógico |

### Diagrama Relacional

Diagrama

Descripción generada automáticamente

### 2.2. Diccionario de datos

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo Entidad/Relacion** | **Atributo** | **Restriccion de Dominio Adicional** | **Valor por defecto** |
| Cliente | descuento | Los valores solo pueden ser:  5% y 10% |  |
| Estudiante | Ciclo | Solo valores : 9 , 10 |  |
| Tratamiento | fecha\_fin | Debe ser mayor a fecha\_inicio |  |
| Mascota | genero | Solo valores:  Macho “m”  Hembra “h” |  |

### Restricciones de integridad

* + - Los valores de peso deben ser ingresados en libras.
    - Los valoras de tamaño deben ser ingresados en centímetros.
    - La edad de la mascota debe registrase en meses.
    - Este atributo descuento debe ser calculado dependiendo de la frecuencia mensual del cliente.
    - Cada veterinaria solo puede tener 3 estudiantes como pasantes.

|  |
| --- |
| Diseño físico para Oracle |

### 3.1 Traducción de modelo lógico a DDL y creación de la base de datos

|  |
| --- |
| DROP DATABASE IF EXISTS veterinaria;  CREATE DATABASE veterinaria;  USE veterinaria;  CREATE TABLE personas (  cedula VARCHAR(20) NOT NULL,  nombres VARCHAR(50) NOT NULL,  apellidos VARCHAR(50) NOT NULL,  telefono\_1 VARCHAR(10) NOT NULL,  telefono\_2 VARCHAR(10),  telefono\_3 VARCHAR(10),  email VARCHAR(70) UNIQUE,  edad INT NOT NULL,  CONSTRAINT personas\_pk PRIMARY KEY (cedula)  );  CREATE TABLE veterinarias (  ruc VARCHAR(20) NOT NULL,  nombre VARCHAR(50) NOT NULL,  direccion VARCHAR(50) NOT NULL,  telefono VARCHAR(10) NOT NULL,  CONSTRAINT veterinaria\_pk PRIMARY KEY (ruc)  );  CREATE TABLE veterinarios(  cedula VARCHAR(20) NOT NULL,  titulo VARCHAR(50) NOT NULL,  ruc VARCHAR(20) NOT NULL,  CONSTRAINT veterinarios\_pk PRIMARY KEY (cedula),  CONSTRAINT veterinarios\_personas\_fk FOREIGN KEY (cedula) REFERENCES personas (cedula),  CONSTRAINT veterios\_veterias\_fk FOREIGN KEY (ruc) REFERENCES veterinarias (ruc)  );  CREATE TABLE clientes(  cedula VARCHAR(20) NOT NULL,  frecuencia\_mensual INT NOT NULL,  descuento DOUBLE,  CONSTRAINT clientes\_pk PRIMARY KEY (cedula),  CONSTRAINT clientes\_personas\_fk FOREIGN KEY (cedula) REFERENCES personas (cedula),  CONSTRAINT clientes\_frecuencia\_ck CHECK (descuento IN (0.05 , 0.10))  );  CREATE TABLE estudiantes(  cedula VARCHAR(20) NOT NULL,  ciclo INT NOT NULL,  ruc VARCHAR(20) NOT NULL,  CONSTRAINT estudiantes\_pk PRIMARY KEY (cedula),  CONSTRAINT estudiantes\_personas\_fk FOREIGN KEY (cedula) REFERENCES personas (cedula),  CONSTRAINT estudiantes\_verterianrias\_fk FOREIGN KEY (ruc) REFERENCES veterinarias(ruc),  CONSTRAINT estudiantes\_ck CHECK (ciclo IN (9 , 10))  );  CREATE TABLE mascotas(  id INT NOT NULL,  nombre VARCHAR(50) NOT NULL,  especie VARCHAR(50) NOT NULL,  raza VARCHAR(50) NOT NULL,  edad INT NOT NULL,  genero CHAR NOT NULL,  peso DOUBLE NOT NULL,  tamanio DOUBLE NOT NULL,  cedula\_cliente VARCHAR(20) NOT NULL,  ruc\_veterinaria VARCHAR(20) NOT NULL,  CONSTRAINT mascotas\_pk PRIMARY KEY (id),  CONSTRAINT mascotas\_genero\_ck CHECK (genero IN ('m' , 'h')),  CONSTRAINT mascotas\_clientes\_fk FOREIGN KEY (cedula\_cliente) REFERENCES clientes (cedula),  CONSTRAINT mascotas\_veterinarias\_fk FOREIGN KEY (ruc\_veterinaria) REFERENCES veterinarias (ruc)  );  CREATE TABLE tratamientos (  id INT NOT NULL,  descripcion VARCHAR(2000) NOT NULL,  fecha\_inicio DATE NOT NULL,  fecha\_fin DATE NOT NULL,  CONSTRAINT tratamientos\_pk PRIMARY KEY (id),  CONSTRAINT tratamientos\_fecha\_fin\_ck CHECK(fecha\_inicio < fecha\_fin)  );  CREATE TABLE medicamentos (  id INT NOT NULL,  nombre VARCHAR(50) NOT NULL,  presentacion VARCHAR(100) NOT NULL,  cantidad CHAR(8) NOT NULL,  CONSTRAINT medicamentos\_pk PRIMARY KEY (id)  );  CREATE TABLE tratamientos\_mascotas (  id\_tratamiento INT NOT NULL,  id\_mascota INT NOT NULL,  CONSTRAINT tratamientos\_mascotas\_pk PRIMARY KEY (id\_tratamiento, id\_mascota),  CONSTRAINT tra\_mas\_tratamientos\_fk FOREIGN KEY (id\_tratamiento) REFERENCES tratamientos (id),  CONSTRAINT tra\_mas\_mascotas\_fk FOREIGN KEY (id\_mascota) REFERENCES mascotas (id)  );  CREATE TABLE medicamentos\_tratamientos (  id\_medicamento INT NOT NULL,  id\_tratamiento INT NOT NULL,  CONSTRAINT medicamentos\_tratamientos\_pk PRIMARY KEY (id\_medicamento, id\_tratamiento),  CONSTRAINT med\_tra\_tratamientos\_fk FOREIGN KEY (id\_tratamiento) REFERENCES tratamientos (id),  CONSTRAINT med\_tra\_medicamentos\_fk FOREIGN KEY (id\_medicamento) REFERENCES medicamentos (id)  ); |

### 3.2 Restricciones o requerimientos que no se pueden implementar con DDL

|  |  |
| --- | --- |
| **Restricción / Requerimiento** | **¿Cómo se debería controlar?** |
| Los valores de peso deben ser ingresados en libras. | Controlar a nivel de programación antes de ingresar el peso |
| Los valoras de tamaño deben ser ingresados en centímetros. | Controlar a nivel de programación antes de ingresar el tamaño |
| La edad de la mascota debe registrase en meses. | Controlar a nivel de programación antes de ingresar la edad |
| Este atributo descuento debe ser calculado dependiendo de la frecuencia mensual del cliente. | Realizar un procedimiento que calcule el descuento. |
| Cada veterinaria solo puede tener 3 estudiantes como pasantes. | Controlar con un trigger antes de ingresar un estudiante como pasante de una veterinaria |

|  |
| --- |
| Procedimientos almacenados y Triggers |
| -- Función  -- Mostrar los nombres de las mascotas que sigue en tratamiento de cada veterinaria  DROP FUNCTION IF EXISTS MascotasTratamiento;  DELIMITER //  CREATE FUNCTION MascotasTratamiento (ruc\_vet VARCHAR(20)) RETURNS VARCHAR(1000)  BEGIN  DECLARE resultado VARCHAR(1000) DEFAULT '';  DECLARE nom VARCHAR(20);  DECLARE fin INTEGER DEFAULT 0;  DECLARE nom\_cursor CURSOR FOR  SELECT m.nombre  FROM mascotas m  JOIN tratamientos\_mascotas tm on tm.id\_mascota = m.id  JOIN tratamientos t on t.id = tm.id\_tratamiento  WHERE m.ruc\_veterinaria = ruc\_vet  AND t.fecha\_fin > CURDATE();    DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;  OPEN nom\_cursor;  get\_nom:LOOP  FETCH nom\_cursor INTO nom;  IF fin = 1 THEN  LEAVE get\_nom;  END IF;  SET resultado := CONCAT(resultado,',', nom);  END LOOP get\_nom;  CLOSE nom\_cursor;  RETURN substr(resultado, 2);  END  //  DELIMITER;  -- Procedimiento  DROP PROCEDURE IF EXISTS CalcularDescuento;  DELIMITER //  CREATE PROCEDURE CalcularDescuento(identf VARCHAR(20))  BEGIN  DECLARE desct DOUBLE DEFAULT 0.0;  SELECT  CASE  WHEN frecuencia\_mensual > 5 THEN 0.10  WHEN frecuencia\_mensual > 3 THEN 0.05  ELSE 0.0  END INTO desct  FROM clientes  WHERE cedula = identf;  UPDATE clientes SET descuento = desct WHERE cedula = identf;  END//  DELIMITER  -- Trigger  DROP TRIGGER IF EXISTS MaxEstudiantes;  DELIMITER //  CREATE TRIGGER MaxEstudiantes  BEFORE INSERT ON estudiantes  FOR EACH ROW  BEGIN  DECLARE cant\_est INT;  SELECT count(\*) INTO cant\_est  FROM estudiantes  WHERE ruc = new.ruc;  IF cant\_est >= 3 THEN  signal sqlstate '45000' set message\_text = 'La veterinaria ya cuenta con el maximo de estudiantes';  END IF;  END// |

|  |
| --- |
| Reportes estadísticos |
| -- 1. Buscar todos las mascotas que son perros mayores a 18 meses  SELECT m.nombre, m.edad  FROM mascotas m  WHERE m.edad > 20  ORDER BY m.edad DESC;  -- 2. Contar el numero de mascotas que tiene cada veterinaria ordenado de mayor a menor  SELECT v.nombre, COUNT(\*) "mascotas ingresadas"  FROM mascotas m  JOIN veterinarias v ON m.ruc\_veterinaria = v.ruc  GROUP BY 1  ORDER BY 2;  -- 3. Consultar todos los estudiantes que su nombre comienze con "R" y sean de decimo ciclo  SELECT p.cedula, p.nombres, p.apellidos, e.ciclo  FROM personas p  JOIN estudiantes e ON e.cedula = p.cedula  WHERE e.ciclo = 10 AND p.nombres LIKE "R%";  -- 4. Mostrat cuantas mascotas existen de acuerdo a su especie  SELECT especie, count(\*)  FROM mascotas  GROUP BY 1;  -- 5. Numero de tratamientos dados ordenados segun su mes por fecha de inicio  SELECT DATE\_FORMAT(fecha\_inicio, "%M") "MES DE TRATAMIENTO", count(\*)"NUMERO DE TRATAMIENTOS"  FROM tratamientos  GROUP BY 1;  -- 6. Mostrar el nombre, especie y tamaño de las mascotas las cuales tengan  -- un tamañoo mayor a 25 cm  SELECT m.nombre, m.especie, m.tamanio  FROM mascotas m  WHERE m.tamanio > 25;  -- 7. Contar el numero de medicamentos que toman las mascotas que siguen un tratamiento  SELECT m.nombre, count(\*) "Medicamentos tomados"  FROM mascotas m  JOIN tratamientos\_mascotas tm ON tm.id\_mascota = m.id  JOIN medicamentos\_tratamientos mt ON mt.id\_tratamiento = tm.id\_tratamiento  GROUP BY 1;  -- 8. los clientes que han superado la media de frecuencia mensual  SELECT c.cedula, p.nombres, p.apellidos, c.frecuencia\_mensual  FROM clientes c  JOIN personas p on p.cedula = c.cedula  WHERE c.frecuencia\_mensual > (SELECT AVG(frecuencia\_mensual)  FROM clientes);  -- 9. Indicar el nombre de los veterinarios que tienen 3 o mas mascotas en su veterianria  SELECT p.nombres, count(\*)  FROM veterinarios vts  RIGHT JOIN personas p ON p.cedula = vts.cedula  JOIN veterinarias vta ON vta.ruc = vts.ruc  JOIN mascotas m ON m.ruc\_veterinaria = vta.ruc  GROUP BY 1  HAVING count(\*) >= 3;  -- 10. Mostrar los datos del cliente que la mascota más pequeña  SELECT p.cedula, p.nombres, p.apellidos, p.telefono\_1, m.nombre, m.tamanio  FROM personas p  JOIN clientes c ON p.cedula = c.cedula  JOIN mascotas m ON c.cedula = m.cedula\_cliente  WHERE m.tamanio = (SELECT MIN(tamanio) FROM mascotas); |

|  |
| --- |
| Implementación de la base de datos en Oracle |

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente