
-- 1. TABELAS AUXILIARES

-- Tabela para gerenciar Espécies de Pet (ex: Cão, Gato)

```
CREATE TABLE IF NOT EXISTS `especie` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(50) NOT NULL UNIQUE,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela para gerenciar Raças de Pet (ex: Labrador, Persa)

```
CREATE TABLE IF NOT EXISTS `raca` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(50) NOT NULL,  
  `especie_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`especie_id`) REFERENCES `especie`(`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela para gerenciar Categorias de Produtos (ex: Ração, Brinquedos)

```
CREATE TABLE IF NOT EXISTS `categoria` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL UNIQUE,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- 2. TABELAS PRINCIPAIS

-- Tabela: Fornecedor (Referenciada por procedure_registrar_fornecedor)

```
CREATE TABLE IF NOT EXISTS `fornecedor` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome_fantasia` VARCHAR(100) NOT NULL,  
  `razao_social` VARCHAR(150) NOT NULL,  
  `cnpj` VARCHAR(18) NOT NULL UNIQUE,  
  `telefone` VARCHAR(15),  
  `email` VARCHAR(100),  
  `contato` VARCHAR(100),  
  `cep` VARCHAR(10),  
  `rua` VARCHAR(150),  
  `bairro` VARCHAR(100),  
  `numero` VARCHAR(20),  
  `complemento` VARCHAR(100),  
  `observacoes` TEXT,  
  `data_cadastro` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Cliente (Referenciada por procedure_cadastrar_cliente)

```
CREATE TABLE IF NOT EXISTS `cliente` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `cpf` VARCHAR(14) NOT NULL UNIQUE,  
  `data_nascimento` DATE,  
  `telefone` VARCHAR(15),  
  `email` VARCHAR(100),  
  `cep` VARCHAR(10),  
  `rua` VARCHAR(150),  
  `bairro` VARCHAR(100),  
  `numero` VARCHAR(20),  
  `complemento` VARCHAR(100),  
  `estado` VARCHAR(20),  
  `observacoes` TEXT,  
  `data_cadastro` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Produto (Referenciada por procedure_cadastrar_produto)

```
CREATE TABLE IF NOT EXISTS `produto` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `descricao` TEXT,  
  `categoria_id` INT,  
  `fornecedor_padrao_id` INT,  
  `preco_custo` DECIMAL(10,2) NOT NULL,  
  `preco_venda` DECIMAL(10,2) NOT NULL,  
  `quantidade_estoque` INT DEFAULT 0,  
  `estoque_minimo` INT DEFAULT 0,  
  `codigo_barras` VARCHAR(50) UNIQUE,  
  `data_cadastro` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`categoria_id`) REFERENCES `categoria`(`id`) ON DELETE SET NULL,  
  FOREIGN KEY (`fornecedor_padrao_id`) REFERENCES `fornecedor`(`id`) ON DELETE SET  
  NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Pet (Referenciada por procedure_cadastrar_pet e procedure_registrar_vacina)

```
CREATE TABLE IF NOT EXISTS `pet` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(50) NOT NULL,  
  `cliente_id` INT NOT NULL, -- Chave estrangeira para o dono  
  `especie_id` INT NOT NULL,  
  `raca_id` INT,  
  `data_nascimento` DATE,  
  `peso` DECIMAL(5,2),  
  `cor` VARCHAR(30),  
  `castrado` TINYINT(1) DEFAULT 0,  
  `vacinado` TINYINT(1) DEFAULT 0, -- Atualizado pela procedure_registrar_vacina  
  `porte` ENUM('Pequeno','Medio','Grande') NOT NULL,  
  `observacoes` TEXT,  
  `foto` VARCHAR(255),  
  `data_cadastro` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`cliente_id`) REFERENCES `cliente`(`id`) ON DELETE CASCADE,  
  FOREIGN KEY (`especie_id`) REFERENCES `especie`(`id`) ON DELETE RESTRICT,  
  FOREIGN KEY (`raca_id`) REFERENCES `raca`(`id`) ON DELETE SET NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Carteira_Vacina (Referenciada por procedure_registrar_vacina e function proxima_vacina_pet)

```
CREATE TABLE IF NOT EXISTS `carteira_vacina` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `pet_id` INT NOT NULL,  
  `nome_vacina` VARCHAR(100) NOT NULL,  
  `data_aplicacao` DATE NOT NULL,  
  `data_proxima` DATE,  
  `veterinario` VARCHAR(100),  
  `observacoes` TEXT,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`pet_id`) REFERENCES `pet`(`id`) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Agendamento (Inferida pela function total_servicos_agendamento)

```
CREATE TABLE IF NOT EXISTS `agendamento` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `pet_id` INT NOT NULL,  
  `data_hora_agendada` DATETIME NOT NULL,  
  `tipo_servico` VARCHAR(100) NOT NULL,  
  `status` ENUM('Agendado','Realizado','Cancelado') DEFAULT 'Agendado',  
  `observacoes` TEXT,  
  `data_criacao` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`pet_id`) REFERENCES `pet`(`id`) ON DELETE RESTRICT  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Tabela: Servico_Realizado (Referenciada pela function total_servicos_agendamento)

```
CREATE TABLE IF NOT EXISTS `servico_realizado` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `agendamento_id` INT NOT NULL,  
  `descricao_servico` VARCHAR(255) NOT NULL,  
  `valor` DECIMAL(10,2) NOT NULL,  
  `data_realizacao` DATETIME DEFAULT CURRENT_TIMESTAMP,  
  `funcionario_id` INT, -- Assumido  
  `observacoes` TEXT,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`agendamento_id`) REFERENCES `agendamento`(`id`) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

DELIMITER \$\$

```
/* -----  
  PROCEDURES  
  ----- */
```

```
PROCEDURE `procedure_cadastrar_cliente` (IN `p_nome` VARCHAR(100), IN `p_cpf`  
VARCHAR(14), IN `p_data_nascimento` DATE, IN `p_telefone` VARCHAR(15), IN `p_email`  
VARCHAR(100), IN `p_cep` VARCHAR(10), IN `p_rua` VARCHAR(150), IN `p_bairro`  
VARCHAR(100), IN `p_numero` VARCHAR(20), IN `p_complemento` VARCHAR(100), IN  
`p_estado` VARCHAR(20), IN `p_observacoes` TEXT) BEGIN  
  INSERT INTO `cliente` (  
    `nome`, `cpf`, `data_nascimento`, `telefone`, `email`, `cep`, `rua`, `bairro`, `numero`,  
    `complemento`, `estado`, `observacoes`  
  )  
  VALUES (  
    p_nome, p_cpf, p_data_nascimento, p_telefone, p_email, p_cep, p_rua, p_bairro, p_numero,  
    p_complemento, p_estado, p_observacoes  
  );  
END$$
```

```
PROCEDURE `procedure_cadastrar_pet` (IN `p_nome` VARCHAR(50), IN `p_cliente_id` INT,  
IN `p_especie_id` INT, IN `p_raca_id` INT, IN `p_data_nascimento` DATE, IN `p_peso`  
DECIMAL(5,2), IN `p_cor` VARCHAR(30), IN `p_castrado` TINYINT, IN `p_vacinado`  
TINYINT, IN `p_porte` ENUM('Pequeno','Medio','Grande'), IN `p_observacoes` TEXT, IN `p_foto`  
VARCHAR(255)) BEGIN  
  INSERT INTO `pet` (  
    `nome`, `cliente_id`, `especie_id`, `raca_id`, `data_nascimento`, `peso`, `cor`, `castrado`,  
    `vacinado`, `porte`, `observacoes`, `foto`  
  )  
  VALUES (  
    p_nome, p_cliente_id, p_especie_id, p_raca_id, p_data_nascimento, p_peso, p_cor, p_castrado,  
    p_vacinado, p_porte, p_observacoes, p_foto  
  );  
  SELECT LAST_INSERT_ID() AS id_pet_cadastrado;  
END$$
```

```

PROCEDURE `procedure_cadastrar_produto` (IN `p_nome` VARCHAR(100), IN `p_descricao`
TEXT, IN `p_categoria_id` INT, IN `p_fornecedor_padrao_id` INT, IN `p_preco_custo`
DECIMAL(10,2), IN `p_preco_venda` DECIMAL(10,2), IN `p_quantidade_estoque` INT, IN
`p_estoque_minimo` INT, IN `p_codigo_barras` VARCHAR(50)) BEGIN
INSERT INTO `produto` (
    `nome`, `descricao`, `categoria_id`, `fornecedor_padrao_id`, `preco_custo`, `preco_venda`,
    `quantidade_estoque`, `estoque_minimo`, `codigo_barras`
)
VALUES (
    p_nome, p_descricao, p_categoria_id, p_fornecedor_padrao_id, p_preco_custo, p_preco_venda,
    p_quantidade_estoque, p_estoque_minimo, p_codigo_barras
);
SELECT LAST_INSERT_ID() AS id_produto_cadastrado;
END$$

```

```

PROCEDURE `procedure_registrar_fornecedor` (IN `p_nome_fantasia` VARCHAR(100), IN
`p_razao_social` VARCHAR(150), IN `p_cnpj` VARCHAR(18), IN `p_telefone` VARCHAR(15),
IN `p_email` VARCHAR(100), IN `p_contato` VARCHAR(100), IN `p_cep` VARCHAR(10), IN
`p_rua` VARCHAR(150), IN `p_bairro` VARCHAR(100), IN `p_numero` VARCHAR(20), IN
`p_complemento` VARCHAR(100), IN `p_observacoes` TEXT) BEGIN
INSERT INTO `fornecedor` (
    `nome_fantasia`, `razao_social`, `cnpj`, `telefone`, `email`,
    `contato`, `cep`, `rua`, `bairro`, `numero`, `complemento`, `observacoes`
) VALUES (
    p_nome_fantasia, p_razao_social, p_cnpj, p_telefone, p_email,
    p_contato, p_cep, p_rua, p_bairro, p_numero, p_complemento, p_observacoes
);
END$$

```

```

PROCEDURE `procedure_registrar_vacina` (IN `p_pet_id` INT, IN `p_nome_vacina`
VARCHAR(100), IN `p_data_aplicacao` DATE, IN `p_data_proxima` DATE, IN `p_veterinario`
VARCHAR(100), IN `p_observacoes` TEXT) BEGIN
-- INSCRIÇÃO na tabela CARTEIRA_VACINA
INSERT INTO `carteira_vacina` (
    `pet_id`, `nome_vacina`, `data_aplicacao`, `data_proxima`, `veterinario`, `observacoes`
) VALUES (
    p_pet_id, p_nome_vacina, p_data_aplicacao, p_data_proxima, p_veterinario, p_observacoes
);

-- Atualiza o indicador 'vacinado' na tabela PET
UPDATE `pet`
SET `vacinado` = 1
WHERE `id` = p_pet_id;
END$$

```

```

/* -----
FUNCTIONS
----- */

FUNCTION `proxima_vacina_pet` (`p_pet_id` INT) RETURNS DATE READS SQL DATA
BEGIN
    DECLARE v_proxima_data DATE;

    -- Busca a menor data futura de próxima dose na carteira de vacinação
    SELECT MIN(data_proxima)
    INTO v_proxima_data
    FROM carteira_vacina
    WHERE pet_id = p_pet_id
    AND data_proxima IS NOT NULL
    AND data_proxima >= CURDATE();

    RETURN v_proxima_data;
END$$

FUNCTION `total_servicos_agendamento` (`p_agendamento_id` INT) RETURNS
DECIMAL(10,2) READS SQL DATA BEGIN
    DECLARE v_total DECIMAL(10,2);

    -- Soma o valor de todos os serviços realizados que referenciam este agendamento
    SELECT SUM(valor)
    INTO v_total
    FROM servico_realizado
    WHERE agendamento_id = p_agendamento_id;

    -- Retorna 0.00 se a soma for nula
    RETURN IFNULL(v_total, 0.00);
END$$

DELIMITER ;
COMMIT;

```