



CURSO DE SQL

PROJETO FINAL

**ANA CARLA OLIVEIRA
ANA CAROLINA FORTUNATO**

DEZEMBRO 2022

1 INTRODUÇÃO

Com a evolução tecnológica ao longo dos anos, o mundo está cheio de dados. Através do uso de celulares, computadores, tablets, relógios inteligentes, dentre tanto outros dispositivos eletrônicos disponíveis no mercado, dados são gerados a todo momento. Logo, estas informações precisam ser armazenadas em locais seguros para serem acessadas sempre que necessário. Estes locais são conhecidos como banco de dados.

Nos bancos de dados, além de armazenar, também fornecem funcionalidades para manipular os dados (inserir, alterar, deletar, consultar) de uma maneira mais ágil. Ressaltamos também que os bancos de dados estão disponíveis em vários formatos, a exemplo de: XML, relacional, não relacional, grafo etc. No desenvolvimento do nosso projeto, o formato utilizado é o comumente mais praticado no mercado, o banco de dados relacional.

Em um banco de dados relacional, pode-se armazenar todos os tipos de dados, no estilo mais comum das tabelas. Em que uma linha representa um item e, a coluna representa a propriedade deste item. No banco relacional torna-se muito fácil formar relações entre tabelas.

A grande maioria dos bancos de dados, tem uma linguagem de consulta para interagir com o banco. A linguagem SQL (structured query language) foi desenvolvida inteiramente para acessar banco de dados, e é a mais popular de todas. É através desta linguagem que conseguimos inserir, alterar, consultar ou deletar dados, usando os comandos insert, truncate, delete, select, dentre outros para trazer insights de negócios.

2 OBJETIVO

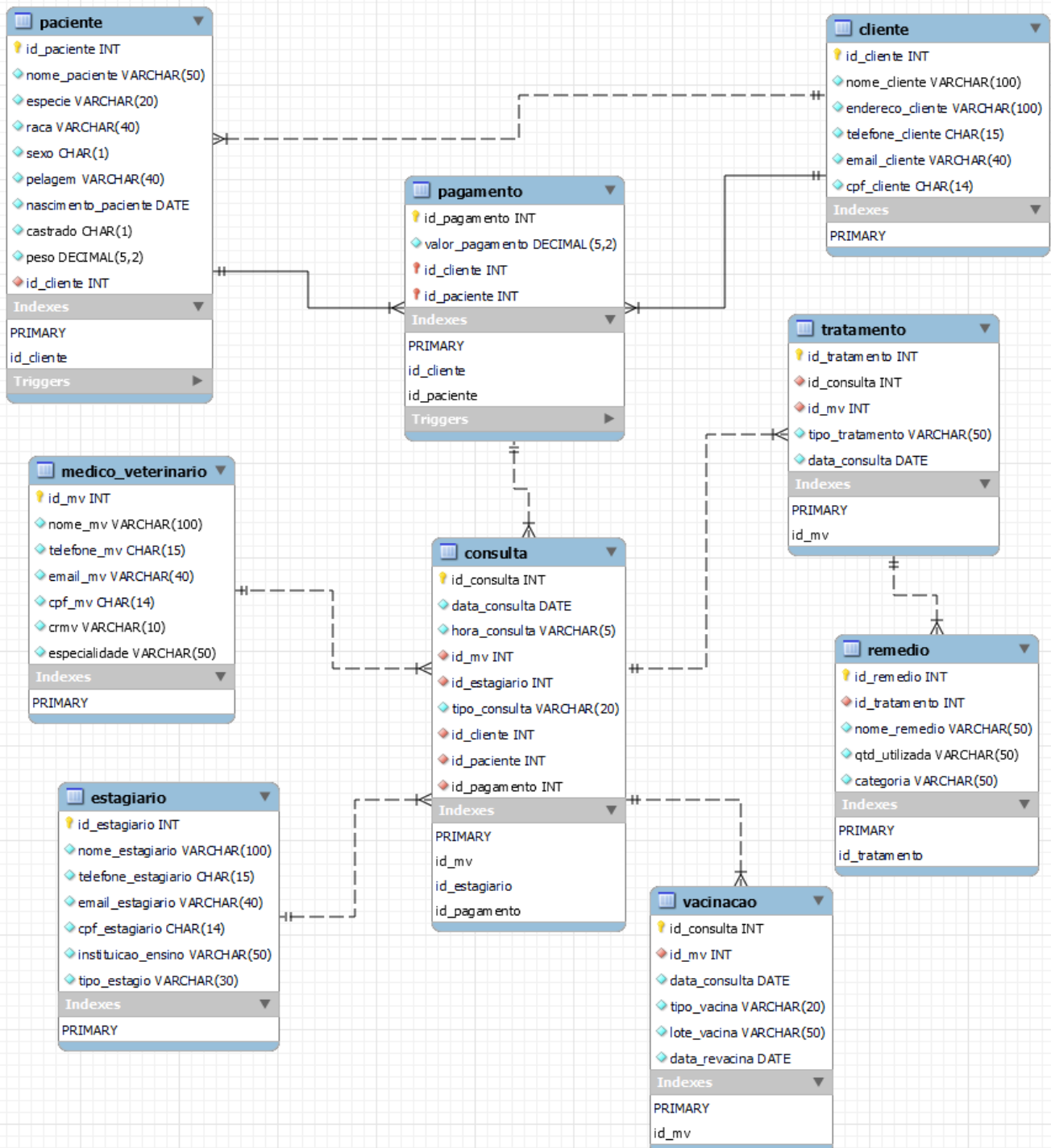
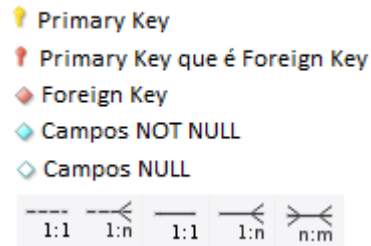
Criar um banco de dados relacional e administrar ações sobre ele a partir da temática escolhida.

3 SITUAÇÃO-PROBLEMA e MODELO DE NEGÓCIO

O tema do banco do nosso banco de dados é **Clínica Veterinária**, que consiste num banco de dados que armazena as informações de uma clínica veterinária de pequenos animais (cachorro e gato), que necessita de um controle específico sobre seus clientes e pacientes, que passam, ao pagarem, em consultas específicas de clínica, dermatologia, ortopedia e vacinação nesse esquema. Nessas consultas também possuem tratamentos prescritos e remédios que serão usados para cada caso. Tudo isso sendo feito por médicos veterinários e com ajuda dos estagiários devidamente cadastrados na clínica.

5 DIAGRAMA E-R

LEGENDA:



6 LISTAGEM DE TABELAS COM DESCRIÇÃO DE SUA ESTRUTURA (COLUNA, DESCRIÇÃO, TIPO DE DADO, TIPO DE CHAVE)

Tabela **cliente** onde contém cadastro com informações pessoais do tutor dos animais da clínica.

Chaves	Campos	Tipos de Dados
PK	id_cliente	int
	nome_cliente	varchar (100)
	endereço_cliente	varchar (100)
	telefone_cliente	char (15)
	email_cliente	varchar (40)
	cpf_cliente	char (14)

Tabela **paciente** onde contém cadastro com informações pessoais do animal que será atendido.

Chaves	Campos	Tipos de Dados
PK	id_paciente	int
	nome_paciente	varchar (50)
	especie	varchar (20)
	raca	varchar (40)
	sexo	char (1)
	pelagem	varchar (40)
	nascimento_paciente	date
	castrado	char (1)
	peso	decimal (5,2)
FK	cliente.id_cliente	int

Tabela **pagamento** que registra o pagamento de determinada consulta.

Chaves	Campos	Tipos de Dados
PK	id_pagamento	int
	valor_pagamento	decimal (5,2)
PK	cliente.id_cliente	int
PK	paciente.id_paciente	int

Tabela **medico_veterinario** onde contém informações pessoas e profissionais sobre os médicos veterinários responsáveis pelos atendimentos na clínica.

Chaves	Campos	Tipos de Dados
PK	id_mv	int
	nome_mv	varchar (100)
	telefone_mv	char (15)
	email_mv	varchar (40)
	cpf_mv	char (14)
	crm_mv	varchar (10)
	especialidade	varchar (50)

Tabela **estagiario** onde contém informações pessoas e sobre o estágio dos estagiários da clínica, eles auxiliam o médico veterinário nos atendimentos.

Chaves	Campos	Tipos de Dados
PK	id_estagiario	int
	nome_estagiario	varchar (100)
	telefone_estagiario	char (15)
	email_estagiario	varchar (40)
	cpf_estagiario	char (14)
	instituicao_ensino	varchar (50)
	tipo_estagio	varchar (30)

Tabela **consulta** que contém os registros das consultas realizadas na clínica.

Chaves	Campos	Tipos de Dados
	id_consulta	int
PK	data_consulta	date
PK	hora_consulta	varchar (5)
FK	medico_veterinario.id_mv	int
FK	estagiario.id_estagiario	int
	tipo_consulta	varchar (20)
FK	pagamento.cliente.id_cliente	int
FK	pagamento.paciente.id_paciente	int
FK	pagamento.id_pagamento	int

Tabela **tratamento** que registra tratamento prescrito através de receituário pelo médico veterinário de acordo com a queixa presente no animal que passou em consulta.

Chaves	Campos	Tipos de Dados
PK	id_tratamento	int
FK	consulta.id_consulta	int
FK	consulta.medico_veterinario.id_mv	int
	tipo_tratamento	varchar (50)
FK	consulta.data_consulta	date

Tabela **remedio** que registra os remédios prescritos em tratamento descrito como “medicamentoso”.

Chaves	Campos	Tipos de Dados
PK	id_remedio	int
FK	tratamento.id_tratamento	int
	nome_remedio	varchar (50)
	qtd_utilizada	varchar (50)
	categoria	varchar (50)

Tabela **vacinacao** armazena um tipo de consulta específica que é a vacinação, pois para esse caso temos a necessidade do registro de datas de vacinação e revacinação, assim como as informações sobre a vacina aplicada.

Chaves	Campos	Tipos de Dados
PK	consulta.id_consulta	int
FK	consulta.medico_veterinario.id_mv	int
FK	consulta.data_consulta	date
	tipo_vacina	varchar (20)
	lote_vacina	varchar (50)
	data_revacina	date

7 SCRIPTS DE CRIAÇÃO DE CADA OBJETO DO BANCO DE DADOS

Toda estrutura do banco de dados relacional criado para esse projeto está armazenada em um arquivo no GitHub intitulado 'DB_clinica_vet.sql', seu acesso pode ser feito através do link a seguir:

https://github.com/anaalucca/SQL/blob/main/DB_clinica_vet.sql

8 SCRIPTS DE INSERÇÃO DE DADOS

Os dados inseridos no banco de dados presente estão salvos em um arquivo armazenado no GitHub intitulado 'clinica_vet_dados.sql', seu acesso pode ser feito por meio do link abaixo:

https://github.com/anaalucca/SQL/blob/main/clinica_vet_dados.sql

9 RELATÓRIOS GERADOS COM BASE NA INFORMAÇÃO ARMAZENADA NAS TABELAS

O schema clinica_vet contém 9 **tabelas** principais, sendo elas cliente, consulta, estagiario, medico_veterinario, paciente, pagamento, remedio, tratamento e vaccinacao.

Ao usar SELECT * FROM em cada tabela temos os resultados a seguir:

1. Tabela **CLIENTE**:

```
SELECT * FROM cliente;
```

id_cliente	nome_cliente	endereco_cliente	telefone_cliente	email_cliente	cpf_cliente
1	Patricia Johnson	Workhaven Lane 23	11951009653	patricia.johnson@gmail.com	19574179113
2	Linda Williams	Lillydale Drive 1411	11924455222	linda.williams@gmail.com	12938923155
3	Barbara Jones	Hanoi Way 1913	11982181037	barbara.jones@gmail.com	44947611200
4	Matthew Mahan	Joliet Street 1121	11906367268	matthew.mahan@hotmail.com	38047827859
5	Timothy Bunn	Santiago de Compostela Parkway 1795	11915525745	timothy.bunn@hotmail.com	23055982286

2. Tabela **CONSULTA**:

```
SELECT * FROM consulta;
```

id_consulta	data_consulta	hora_consulta	id_mv	id_estagiario	tipo_consulta	id_cliente	id_paciente	id_pagamento
1	2022-10-10	08:30	4	2	dermatologia	1	1	1
2	2022-10-10	08:30	1	4	clínica	2	2	2
3	2022-10-10	09:00	1	4	vacinação	3	3	3
4	2022-10-10	10:00	4	2	retorno dermatologia	4	4	4
5	2022-10-10	10:30	1	4	vacinação	5	5	5

3. Tabela **ESTAGIARIO**:

SELECT * FROM estagiario;

id_estagiario	nome_estagiario	telefone_estagiario	email_estagiario	cpf_estagiario	instituicao_ensino	tipo_estagio
1	Lorena Grecco	11990872660	loogreco@gmail.com	38342632627	Universidade Anhembí Morumbi	obrigatório
2	Gabriela Ayumi	11975007954	ayumi@hotmail.com	44931804490	Faculdade Anhaguera	obrigatório
3	Bianca Rossi	11914007459	biarossi99@gmail.com	42661466650	Faculdades Metropolitanas Unidas	extracurricular
4	Gabriel Moraes	11979644830	gab_moraes@gmail.com	40759769238	Universidade Metodista de São Paulo	obrigatório
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Tabela **MEDICO_VETERINARIO**:

SELECT * FROM medico_veterinario;

id_mv	nome_mv	telefone_mv	email_mv	cpf_mv	crm	especialidade
1	Thalita de Brito	11999106258	thalita.britovet@gmail.com	32242663055	49933SP	clínica de pequenos animais
2	Vitória Gillioli	11923800583	vitoria.gillioli@hotmail.com	46530529606	57007SP	clínica de pequenos animais
3	Thiago Chagas	21941530148	thiago_orto@hotmail.com	14174590809	40524RJ	ortopedia
4	Tânia Fernandes	11988480476	tania.fernandes@hotmail.com	15265058449	1848SP	dermatologia
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Tabela **PACIENTE**

SELECT * FROM paciente;

id_paciente	nome_paciente	especie	raca	sexo	pelagem	nascimento_paciente	castrado	peso	id_cliente
1	Belinha	canino	poodle	F	branco	2010-08-23	S	8.50	1
2	Thor	canino	pitbull	M	cinza	2020-09-22	N	18.60	2
3	Leona	canino	SRD	F	caramelo	2021-05-05	S	20.10	3
4	Bob	canino	shih tzu	M	bege	2008-10-10	S	4.30	4
5	Oscar	felino	SRD	M	tigrado com branco	2021-05-08	S	5.50	5

6. Tabela **PAGAMENTO**:

SELECT * FROM pagamento;

id_pagamento	valor_pagamento	id_cliente	id_paciente
1	260.00	1	1
2	70.00	2	2
3	115.00	3	3
4	0.00	4	4
5	120.00	5	5

7. Tabela **REMEDIO**:

SELECT * FROM remedio;

id_remedio	id_tratamento	nome_remedio	qtd_utilizada	categoria
1	1	Meticorten 5mg	1 cp VO SID - 5 dias	antialérgico
2	2	Gaviz 10mg	1 cp VO BID - 10 dias	protetor gástrico
3	3	Cronidor 12mg	1 cp VO BID - 5 dias	analgésico
4	3	Prediderm 5mg	1 cp VO SID - 5 dias	antiinflamatório
5	4	Dipirona	5 gotas VO TID - 5 dias	analgésico

8. Tabela **TRATAMENTO**:

```
SELECT * FROM tratamento;
```

id_tratamento	id_consulta	id_mv	tipo_tratamento	data_consulta
1	1	4	medicamentoso	2022-10-10
2	2	1	medicamentoso	2022-10-10
3	7	1	medicamentoso	2022-10-10
4	8	1	medicamentoso	2022-10-10
5	20	1	medicamentoso	2022-10-11

9. Tabela **VACINACAO**:

```
SELECT * FROM vacinacao;
```

id_consulta	id_mv	data_consulta	tipo_vacina	lote_vacina	data_revacina
3	1	2022-10-10	V10 e raiva	320/11 e 508/22	2023-10-10
5	1	2022-10-10	V5 e raiva	123/22 e 508/22	2023-10-10
6	1	2022-10-10	V5 e raiva	123/22 e 508/23	2023-10-11
11	2	2022-10-10	giardíase e gripe	456/22 e 378/22	2022-11-10
12	2	2022-10-10	V8 e raiva	245/22 e 508/22	2023-10-10

Durante as entregas foram criadas 5 **views** para facilitar o acesso as informações das tabelas de forma específica, sendo elas: histórico de retornos; histórico de todos os gatos vacinados; histórico de todos atendimentos clínicos em outubro de 2022; histórico de todos os atendimentos feitos pela médica veterinária Thalita de Brito em outubro de 2022; e histórico de pacientes que tiveram tratamento medicamentoso receitado.

Abaixo podemos ver como essas views foram criadas e o que elas retornam:

1. View **retornos**:

```
CREATE OR REPLACE VIEW retornos AS (  
    SELECT data_consulta, nome_cliente, nome_paciente, valor_pagamento, tipo_consulta  
    FROM consulta c JOIN pagamento p ON (c.id_pagamento = p.id_pagamento)  
    JOIN cliente l ON (c.id_cliente = l.id_cliente)  
    JOIN paciente pa ON (pa.id_cliente = l.id_cliente)  
    WHERE valor_pagamento = 0  
    ORDER BY data_consulta, nome_cliente ASC  
);
```

```
SELECT * FROM retornos;
```

data_consulta	nome_cliente	nome_paciente	valor_pagamento	tipo_consulta
2022-10-10	Alberto Henning	Batata	0.00	retorno clínica
2022-10-10	Alfredo Mcadams	Lulu	0.00	retorno clínica
2022-10-10	Matthew Mahan	Bob	0.00	retorno dermatologia
2022-10-11	Alice Stewart	Lady	0.00	retorno clínica
2022-10-11	Pamela Baker	Gatoso	0.00	retorno clínica

2. View **gatos_vacinados**:

```
CREATE OR REPLACE VIEW gatos_vacinados AS (  
    SELECT nome_cliente, nome_paciente, tipo_vacina FROM consulta c  
    JOIN paciente p ON (p.id_paciente = c.id_paciente)  
    JOIN cliente l ON (l.id_cliente = p.id_cliente)  
    JOIN vaccinacao v ON (v.id_consulta = c.id_consulta)  
    WHERE tipo_consulta = "vacinação" and especie = "felino"  
    ORDER BY nome_cliente);
```

```
SELECT * FROM gatos_vacinados;
```

nome_cliente	nome_paciente	tipo_vacina
Carol Garcia	Pérola	V4
Jennifer Davis	Zeus	V4 e raiva
Leonard Schofield	Gaia	V4 e raiva
Luis Yanez	Tadeu	V5
Maria Miller	Gege	V5 e raiva

3. View **historico_clinica**:

```
CREATE OR REPLACE VIEW historico_clinica AS (  
    SELECT data_consulta, nome_mv, tipo_consulta  
    FROM consulta c JOIN medico_veterinario mv ON (c.id_mv = mv.id_mv)  
    WHERE tipo_consulta IN ('clínica', 'retorno clínica') and MONTH(data_consulta) = 10 and YEAR(data_consulta) = 2022  
);
```

```
SELECT * FROM historico_clinica;
```

data_consulta	nome_mv	tipo_consulta
2022-10-10	Thalita de Brito	clínica
2022-10-10	Thalita de Brito	clínica
2022-10-10	Thalita de Brito	retorno clínica
2022-10-10	Thalita de Brito	retorno clínica
2022-10-11	Thalita de Brito	clínica

4. View **historico_thalita**:

```
CREATE OR REPLACE VIEW historico_thalita AS (  
    SELECT data_consulta, nome_mv, tipo_consulta, valor_pagamento  
    FROM consulta c JOIN medico_veterinario mv ON (c.id_mv = mv.id_mv)  
    JOIN pagamento p ON (c.id_pagamento = p.id_pagamento)  
    WHERE nome_mv LIKE '%Thalita de Brito%' and MONTH(data_consulta) = 10 and YEAR(data_consulta) = 2022  
);
```

```
SELECT * FROM historico_thalita;
```

data_consulta	nome_mv	tipo_consulta	valor_pagamento
2022-10-10	Thalita de Brito	clínica	70.00
2022-10-10	Thalita de Brito	vacinação	115.00
2022-10-10	Thalita de Brito	vacinação	120.00
2022-10-10	Thalita de Brito	vacinação	120.00
2022-10-10	Thalita de Brito	clínica	170.00

5. View **paciente_tratamento**:

```
CREATE OR REPLACE VIEW paciente_tratamento AS (  
    SELECT nome_cliente, nome_paciente, peso, nome_remedio, qtd_utilizada FROM consulta c  
        JOIN paciente p ON (p.id_paciente = c.id_paciente)  
        JOIN cliente l ON (l.id_cliente = p.id_cliente)  
        JOIN tratamento t ON (t.id_consulta = c.id_consulta)  
        JOIN remedio r ON (r.id_tratamento = t.id_tratamento)  
        WHERE tipo_tratamento = "medicamentoso"  
        ORDER BY nome_cliente  
);  
  
SELECT * FROM paciente_tratamento;
```

nome_cliente	nome_paciente	peso	nome_remedio	qtd_utilizada
Alexander Fennell	Princesa	3.80	Hemolitan Pet 30 ml	1 ml VO BID - 10 dias
Alfredo Mcadams	Lulu	4.80	Dipirona	5 gotas VO TID - 5 dias
Alfredo Mcadams	Lulu	4.80	Agemoxi 50mg	1/4 cp VO BID - 7 dias
Ana Bradley	Lua	5.00	Vetaglós 50g	1 medida da espátula BID - 10 dias
Bernard Colby	Leona	8.30	Drontal Puppy	3 doses de 2 ml VO SID - 3 dias seguidos

Foram criadas 2 **funções**, sendo uma consulta marcada que ao inserir data, hora e id do médico veterinário ela retorna tipo de consulta, nome do paciente e cliente, e outra de faturamento que retorna o total de faturamento da clínica até a última entrada.

A seguir podemos ver como as funções foram criadas e o que elas retornam ao usar o SELECT:

1. Função **consulta_marcada**:

```
CREATE FUNCTION `consulta_marcada` (data DATE, hora VARCHAR (5), id INT)  
RETURNS VARCHAR (100)  
READS SQL DATA  
BEGIN  
    RETURN (SELECT CONCAT('', tipo_consulta, ', paciente ', nome_paciente, ', cliente ', nome_cliente)  
        FROM consulta c JOIN paciente p ON (c.id_paciente = p.id_paciente)  
        JOIN cliente l ON (p.id_cliente = l.id_cliente)  
        WHERE data_consulta = data and hora_consulta = hora and id_mv = id);  
END  
  
SELECT consulta_marcada ('2022-10-10', '18:00', 2);
```

consulta_marcada ('2022-10-10', '18:00', 2)
vacinação, paciente Arya, cliente Nicholas Barfield

2. Função **faturamento**:

```
CREATE FUNCTION `faturamento` ()  
RETURNS FLOAT  
DETERMINISTIC  
BEGIN  
    DECLARE total FLOAT;  
    SET total = (SELECT SUM(valor_pagamento) FROM pagamento);  
    RETURN total;  
END  
  
SELECT faturamento ();
```

faturamento ()
5975

No projeto foram criados 2 **stored procedures**, sendo um consultas_vet que retorna todas as consultas de determinado veterinário com seu id_mv, mas caso seja inserido um id diferente dos quais há na tabela medico_veterinario/consulta ele retorna mensagem de erro. O outro chamado cliente_paciente retorna de acordo com o nome do cliente qual paciente a ele pertence.

Abaixo podemos ver como eles foram criados e o que retornam ao usar o CALL:

1. Stored Procedure **consultas_vet**:

```
CREATE PROCEDURE `consultas_vet` (IN id INT)  
BEGIN  
    IF id IN (1,2,3,4) THEN  
        SELECT * FROM consulta WHERE consulta.id_mv = id;  
    ELSE  
        SELECT 'ERRO: não existe medico veterinário com esse id' AS aviso;  
    END IF;  
END  
  
CALL consultas_vet (1);
```

id_consulta	data_consulta	hora_consulta	id_mv	id_estagiario	tipo_consulta	id_cliente	id_paciente	id_pagamento
2	2022-10-10	08:30	1	4	clínica	2	2	2
3	2022-10-10	09:00	1	4	vacinação	3	3	3
5	2022-10-10	10:30	1	4	vacinação	5	5	5
6	2022-10-10	10:50	1	4	vacinação	5	6	6
7	2022-10-10	11:30	1	4	clínica	6	7	7

```
CALL consultas_vet (7);
```

aviso
ERRO: não existe medico veterinário com esse id

2. Stored Procedure **cliente_paciente**:

```
1 • CREATE PROCEDURE `cliente_paciente`(IN cliente VARCHAR(100))
2 BEGIN
3     SELECT nome_cliente, nome_paciente, especie, raca, sexo, pelagem, nascimento_paciente, castrado, peso
4     FROM paciente p JOIN cliente c ON (p.id_cliente = c.id_cliente)
5     WHERE nome_cliente = cliente;
6 END
```

```
CALL cliente_paciente('Patricia Johnson');
```

nome_cliente	nome_paciente	especie	raca	sexo	pelagem	nascimento_paciente	castrado	peso
Patricia Johnson	Belinha	canino	poodle	F	branco	2010-08-23	S	8.50

E por último, foi feito criação de **triggers**, sendo um de antes (BEFORE) que armazena valor de pagamento novo antes de inseri-lo na tabela pagamento e um de depois (AFTER) que armazena o peso anterior do animal depois dele ser alterado na tabela paciente.

A seguir podemos ver a criação de uma tabela onde será armazenadas as informações de antes e depois, a criação do trigger e um exemplo de como ele funciona.

1. Trigger AFTER **tr_alter_peso** referente a tabela **peso_anterior**:

```
CREATE TABLE peso_anterior (
    id_alteracao INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    id_paciente INT NOT NULL,
    nome_paciente VARCHAR (50) NOT NULL,
    id_cliente INT NOT NULL,
    peso DECIMAL (5,2) NOT NULL,
    data_alteracao DATE NOT NULL,
    hora_alteracao TIME NOT NULL,
    usuario_alteracao VARCHAR(50) NOT NULL
);

CREATE TRIGGER `tr_alter_peso`
AFTER UPDATE ON `paciente`
FOR EACH ROW
INSERT INTO `peso_anterior` (id_paciente, nome_paciente, id_cliente, peso,
                             data_alteracao, hora_alteracao, usuario_alteracao)
VALUES (OLD.id_paciente, OLD.nome_paciente, OLD.id_cliente, OLD.peso,
        CURRENT_DATE(), CURRENT_TIME(), USER());
```

Exemplo da primeira entrega:

```
UPDATE paciente SET peso = 8.5 WHERE id_paciente = 1;
```

Exemplo feito para essa entrega final:

```
UPDATE paciente SET peso = 12.8 WHERE id_paciente = 24;
```

```
SELECT * FROM peso_anterior;
```

id_alteracao	id_paciente	nome_paciente	id_cliente	peso	data_alteracao	hora_alteracao	usuario_alteracao
1	1	Belinha	1	8.30	2022-11-01	20:21:21	root@localhost
2	24	Princesa	20	12.50	2022-12-07	17:04:40	root@localhost
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Trigger BEFORE **tr_add_pagamento** referente a tabela **novos_pagamentos**:

```
CREATE TABLE novos_pagamentos (
    id_novo INT NOT NULL AUTO_INCREMENT,
    valor_pagamento DECIMAL (5,2) NOT NULL,
    data_pagamento DATE NOT NULL,
    hora_pagamento TIME NOT NULL,
    usuario_pagamento VARCHAR (50) NOT NULL,
    PRIMARY KEY (id_novo)
);

CREATE TRIGGER `tr_add_pagamento`
BEFORE INSERT ON `pagamento`
FOR EACH ROW

INSERT INTO `novos_pagamentos` (valor_pagamento, data_pagamento,
                                hora_pagamento, usuario_pagamento)
VALUES (NEW.valor_pagamento, CURRENT_DATE(), CURRENT_TIME(), USER());
```

```
INSERT INTO pagamento (valor_pagamento, id_cliente, id_paciente) VALUES (70, 2, 2);
```

```
SELECT * FROM novos_pagamentos;
```

id_novo	valor_pagamento	data_pagamento	hora_pagamento	usuario_pagamento
1	70.00	2022-12-07	17:09:30	root@localhost
NULL	NULL	NULL	NULL	NULL

Todos os scripts de criação do que foi mostrado acima e também outros relacionados aos desafios durante o curso estão salvos num repositório no GitHub que pode ser acessado pelo link: <https://github.com/anaalucca/SQL>

10 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Foi usado durante toda execução do projeto o programa **MySQL** para confecção do banco de dados e aplicações dos desafios e entregas durante o curso. Também para geração de alguns dados inseridos nas tabelas, como CPF e números de telefone, que precisávamos de números aleatórios sem ser repetidos, utilizamos o **Excel**.