



Trabalho 01
Documento de Trabalho4-BD1-PET Shop – Pedro
Henrique Silva Santana.pdf

Sistema de Gestão de PET Shop

Equipe:

Guilherme Dias Cardoso – 12011BSI254
Guilherme Rimoldi Kameoka – 12011BSI200
Nicolas de Paula Vieira Pinto – 12011BSI206
Pedro Henrique Silva Santana – 12011BSI218
Rafael Dutra Basso – 11911BSI234
Victor Hugo Martins Alves – 12011BSI217

Líder da Equipe: Pedro Henrique Silva Santana
pedro.santana@ufu.br



Histórico da Participação dos Membros da Equipe

Membro	Data	Descrição
Guilherme Dias Cardoso	14/03/2022 a 18/03/2022	Construção de consultas e participação nas reuniões.
Guilherme Rimoldi Kameoka	14/03/2022 a 18/03/2022	Construção de consultas e participação nas reuniões.
Nicolas de Paula Vieira Pinto	14/03/2022 a 18/03/2022	Construção de consultas e participação nas reuniões.
Pedro Henrique Silva Santana	14/03/2022 a 18/03/2022	Construção de consultas e participação nas reuniões.
Rafael Dutra Basso	14/03/2022 a 18/03/2022	Construção de consultas e participação nas reuniões.
Victor Hugo Martins Alves	14/03/2022 a 18/03/2022	Construção e correção de consultas e participação em reuniões.



1. Descrição do Trabalho

Nosso projeto é um sistema de banco de dados de um Pet Shop com as entidades, entre as principais, Cliente, Pet, Agenda, Produto, Compra e Estoque. No projeto foi destacado as relações entre as entidades e suas cardinalidades. Na segunda etapa do projeto, as chaves das entidades e chaves estrangeiras foram definidas, assim como os tipos de cada atributo e as relações de cada modelo. A terceira etapa foi definida pelos scripts de criação e população do banco. A quarta etapa foi designado ao grupo a criação de consultas para manipulação do nosso banco de dados.

2. Script de Comandos de Manipulação do Banco

```
/* Criar 2 comandos novos de inserção nas principais tabelas do modelo; */

--insercao na tabela cliente
insert into cliente (idcliente, nome, cpf, telefone, dataNascimento, cep,
logradouro, bairro, numeroEndereco, estado) values
(default, 'Tiago Azevedo', '165.854.793-43', '9814-2777', '1980-07-20',
'38408-266', 'Rua João Limão-rio dos Anjos', 'Segismundo Pereira', 1000,
'MG');

--insercao na tabela profissional
insert into profissional (idprofissional, nome, funcao, cpf, telefone,
dataNascimento, cep, logradouro, bairro, numeroEndereco, estado) values
(default, 'Melissa Ribeiro Sousa', 'Veterinário', '770.036.164-91',
'9933-4197', '1985-01-03', '38400-100', 'Avenida João Naves', 'Saraiva',
300, 'MG');

/* Criar 2 comandos de remoção nas principais tabelas, sendo que pelo menos
1 deve exigir remoção em mais de uma tabela; */

--excluir a tupla de estoque com id = 10
delete from estoque where idestoque = 10;

--exclui a vacina da tabela muitos para muitos e depois exclui a propria
vacina
delete from pet_vacina where vacina_idvacina = 10;
delete from vacina where idvacina = 10;

/* Criar 3 comandos de atualização de dados nas principais tabelas do
modelo, sendo que pelo menos 1 delas deve obter o(s) novo(s) valor(es) a
serem armazenados a partir de consulta(s) a outra(s) tabela(s).*/

--update do telefone na tabela cliente
update cliente
set telefone = '9987-7727'
where idcliente = 1;

--update do peso na tabela pet
update pet
set peso = 25
where idpet = 1;
```



```
--update do cartavacina na tabela Pet
update pet
  set cartavacina = 'A, B, C, D'
  where idpet = 1;

--update do preco do produto com base na compra
UPDATE produto SET preco =
  (SELECT preco FROM produto_has_compra WHERE produto_idproduto = 1 and
  compra_idcompra = 1)
  where idproduto = 1;

/* Criar 5 consultas simples nas principais tabelas do modelo usando as
cláusulas WHERE, DISTINCT, BETWEEN, LIKE e ORDER BY; */

--devolve todas as tuplas da tabela pet que tem a especie = 'Cachorro'
select * from pet
  where especie = 'Cachorro';

--devolve todos os tipos distintos de produto
select distinct tipo
  from produto;

--devolve o valor total e a data da venda de vendas entre determinada data
select valortotal, datavenda from ordemservico o
  where datavenda between '2022-02-01' and '2022-02-28';

--devolve os clientes que comecam com a letra L
select nome from cliente
  where upper(nome) like 'L%';

--devolve o nome e a data de nascimento dos profissional em ordem decrescente
pela data de nascimento
select nome, datanascimento from profissional
  order by datanascimento desc;

/* Criar 3 consultas aninhadas (sub consultas) usando IN e NOT IN; */

--todos os produtos que a quantidade da compra foi maior que 3
select *
  from produto p2
  where p2.idproduto
    in(SELECT phc.produto_idproduto
      FROM produto_has_compra phc
      where phc.qtd >=3 );

--todos os pets que não se vacinaram nessa data
select * from pet p where p.idpet not in
  (SELECT pv.pet_idpet
   FROM pet_vacina pv
   WHERE pv.dataaplicacao>'2022/01/10');

--seleciona o nome e o id dos produtos da tabela produtos, onde o preco da
compra = 99.9 ou 78.9
SELECT nome, idproduto
FROM produto
WHERE preco IN (
  SELECT preco
  FROM compra
```



```
WHERE preco = 99.9 OR preco = 78.9  
);
```

```
/* Criar 5 consultas que envolvem funções de agregação (COUNT, SUM, MIN,  
MAX, AVG) com agrupamento (GROUP BY) e HAVING; */
```

```
--conta quantos profissionais existem por funcao  
select funcao, count(*) from profissional  
group by funcao;
```

```
--mostra os tipos de produtos que possuem mais de 2 produtos cadastrados  
select tipo, count(*) as ntipos  
from produto  
group by tipo  
having count(*) > 2;
```

```
--lista todos os tipos de itens diferentes de ração e quantos diferentes  
produtos de cada tipo  
SELECT tipo, count(tipo) as variacoes  
FROM produto p  
group by tipo HAVING p.tipo != 'Ração' ;
```

```
--devolve os pets mais pesados de cada sexo  
SELECT p.sexo, MAX(p.peso)  
FROM pet p  
GROUP BY p.sexo;
```

```
--os produtos mais baratos de cada tipo  
SELECT p.tipo, MIN(p.preco)  
FROM produto p  
GROUP BY p.tipo;
```

```
--media do preco de cada tipo de produtos  
SELECT p.tipo, AVG(p.preco)  
FROM produto p  
GROUP BY p.tipo;
```

```
/* Criar 5 consultas avançadas usando INNER JOIN e (INNER, RIGHT, LEFT,  
FULL) OUTER JOIN; */
```

```
--devolve a junção interna de cliente e pet  
select c.nome, p.nome, p.especie, p.raca  
from cliente c inner join pet p  
on c.idcliente = p.cliente_idcliente;
```

```
--devolve a junção interna de profissional com as OS que ele realizou  
select p.idprofissional, p.nome, p.funcao, ohs.ordemservico_idordemservico,  
ohs.servico_idservico  
from profissional p inner join ordemservico_has_servico ohs  
on p.idprofissional = ohs.profissional_idprofissional;
```

```
--devolve a junção interna de agenda com AHS, mostrando os serviços que estão  
agendados  
select a.idagenda, a.pet_idpet, a.datainicio, ahs.hora, ahs.servico_idservico  
from agenda a inner join agenda_has_servico ahs  
on a.idagenda = ahs.agenda_idagenda;
```

```
--retorna todos os pet's que não estão agendados.
```



```
select p.*
```

```
  from pet p left join agenda a
```

```
  on a.cliente_idcliente = p.idpet WHERE a.cliente_idcliente IS NULL;
```

```
--seleciona todos os produtos e seus preços e os relaciona com sua  
quantidade e validade
```

```
select p.nome, p.preco, e.qtdestoque, e.produto_idproduto, e.validade
```

```
  from produto p inner join estoque e
```

```
  on p.idproduto = e.idestoque;
```

```
--seleciona todos os fornecedores e compras
```

```
select f.nome,f.cnpj,c.idcompra,c.datacompra
```

```
  from Fornecedor f full OUTER JOIN compra c
```

```
    on c.fornecedor_idfornecedor = f.idfornecedor;
```