



Faculdade de Informática e Administração Paulista

Data Application & Data Science

ENTREGÁVEL DA DISCIPLINA

INTEGRANTES

RM (SOMENTE NÚMEROS)	NOME COMPLEMENTO (SEM ABREVIAR)
95138	Mateus Mendes dos Santos
95484	Adolfo Kentaro Hada
93618	Luca Katsumata Spiller
94250	Matheus Gomes Campolongo
92798	Claudio Tiyoki Yamashiro

SUMÁRIO

1 – DESCRIÇÃO DO PROJETO E REGRAS DE NEGÓCIO	5
2 – Dicionário de Dados.....	7
3 – Projeto Lógico do Banco de Dados.....	9
4 – Projeto Físico do Banco de Dados.....	10
5 – Data Definition Language – DDL.....	11
6 – Data Manipulation Language – DML (INSERT).....	14
7 – DATA MANIPULATION LANGUAGE – DML (UPDATE / DELETE)	15
8 – DATA QUERY LANGUAGE – DQL (SELECT).....	16
8.1 – Relatório simples contendo apenas uma tabela com dados ordenados.	16

1 – Descrição do Projeto e Regras de Negócio

Iremos entregar uma solução que seja capaz de entregar um produto de acordo com características/gostos do usuário. Para isso, estaremos utilizando uma empresa fictícia como base, sendo ela a **Nike**.

Onde o recurso chave, será o ChatGPT que com as devidas informações passadas pelo usuário, ele que irá entregar o produto mais adequado.

REGRAS DE NEGÓCIO:

- 1 – Todas as informações de preenchimento de dados pelo usuário no seu cadastro são obrigatórias;
 - 2 – Nome e Sobrenome devem ser preenchidos em colunas diferentes;
 - 3 – E-mail deve ser único, não pode existir mais de um usuário cadastrado com o mesmo e-mail;
 - 4 – A senha deve suportar tanto String quanto números e sinais (deve ser alfanumérica);
 - 5 – Os Identificadores únicos (Id) de todas as tabelas devem ser gerados automaticamente;
 - 6 – Todas as informações do produto são obrigatórias;
 - 7 – O produto deve ter o atributo gênero, sendo ele um Enum que armazena Masculino, Feminino e Unissex;
 - 8 – O nome da categoria deve ser único, não pode existir duas categorias iguais cadastradas no banco;
 - 9 – Muitos Produtos podem pertencer a uma categoria, e várias categorias podem englobar um Produto. Relação de muitos pra muitos entre Produto e Categoria;
- POR EX:** Um tênis pode aparecer tanto na aba de tênis de corridas quanto na aba de Basquete.

10 – Deve existir uma entidade associativa entre Categoria e Produto para serem armazenados os identificadores de cada tabela.

2 – Dicionário de Dados

Tabela	NK TB_USUARIO			
Descrição	Tabela de USUÁRIO. Relaciona-se com a tabela CARRINHO			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
id_usuario	BIGINT		PK	Identificador único do usuário
ds_email	VARCHAR	30	NN	Email do usuário
nm_usuario	VARCHAR	30	NN	Nome do usuário
ds_senha	VARCHAR	30	NN	Senha do usuário
nm_sobrenome	VARCHAR	30	NN	Sobrenome do usuário

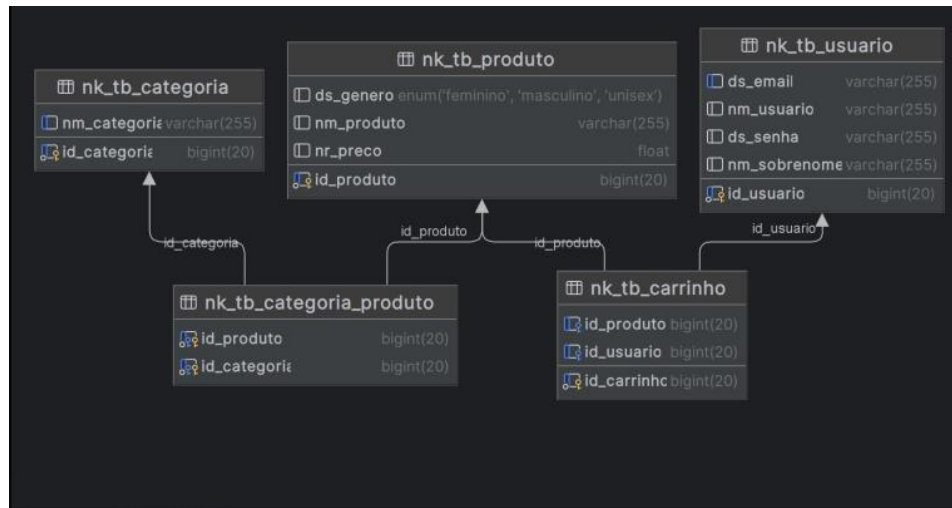
Tabela	NK TB_PRODUTO			
Descrição	Tabela de PRODUTO. Relaciona-se com a tabela CARRINHO, CATEGORIA_PRODUTO			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
id_produto	BIGINT		PK	Identificador único do produto
ds_genero	ENUM	('FEMININO', 'MASCULINO', 'UNISSEX')	NN	Gênero do produto
nm_produto	VARCHAR	50	NN	Nome do produto
nr_preco	FLOAT		NN	Preço do produto

Tabela	NK TB_CARRINHO			
Descrição	Tabela de CARRINHO. Relaciona-se com a tabela USÁRIO, PRODUTO			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
id_carrinho	BIGINT		PK	Identificador único do carrinho
Id_produto	BIGINT		NN	Identificador único do produto
id_usuario	BIGINT		NN	Identificador único do usuário

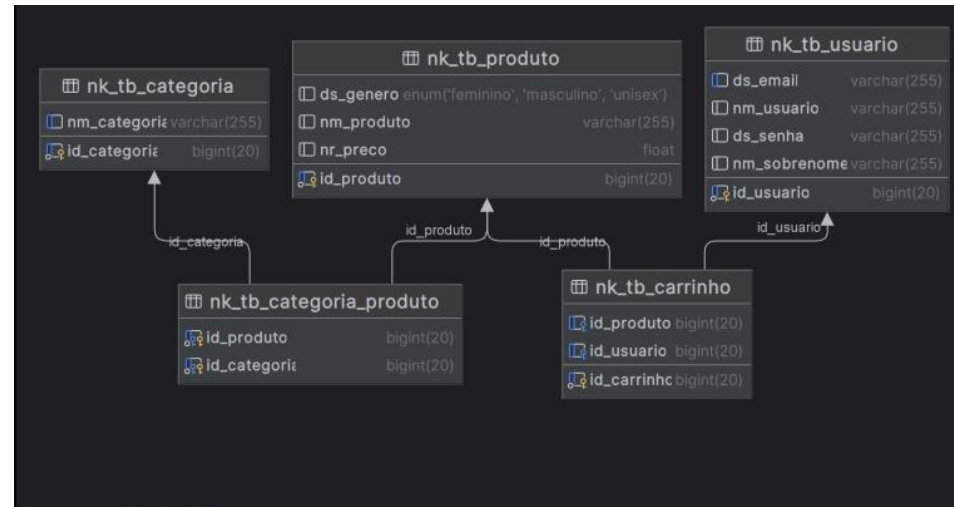
Tabela	NK TB CATEGORIA			
Descrição	Tabela de CATEGORIA. Relaciona-se com a tabela CATEGORIA_PRODUTO			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
id_categoria	BIGINT		PK	Identificador único da categoria
nm_categoria	VARCHAR	30	NN	Nome da categoria

Tabela	NK TB CATEGORIA_PRODUTO			
Descrição	Tabela de CATEGORIA do PRODUTO. Relaciona-se com a tabela CATEGORIA, PRODUTO			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
Id_categoria	BIGINT		NN	Identificador único da categoria
id_produto	BIGINT		NN	Identificador único do produto

3 – Projeto Lógico do Banco de Dados



4 – Projeto Físico do Banco de Dados



5 – Data Definition Language – DDL

```
CREATE DATABASE nike;
USE nike;

DROP TABLE NK_TB_USUARIO;
DROP TABLE NK_TB_PRODUTO;
DROP TABLE NK_TB_CATEGORIA;
DROP TABLE NK_TB_CATEGORIA_PRODUTO;
DROP TABLE NK_TB_CARRINHO;

create table nk_tb_categoria
(
    id_categoria BIGINT          not NULL          AUTO_INCREMENT
        primary key,
    nm_categoria VARCHAR(30) not null,
    constraint UN_NM_CATEGORIA
        unique (nm_categoria)
);

create table nk_tb_produto
(
    id_produto BIGINT          not NULL
        AUTO_INCREMENT
        primary key,
    ds_genero enum ('FEMININO', 'MASCULINO', 'UNISSEX') not
null,
    nm_produto VARCHAR(50)          not
null,
    nr_preco float          not
null
);

create table nk_tb_categoria_produto
(
    id_produto BIGINT not NULL,
    id_categoria bigint not null,
    primary key (id_produto, id_categoria),
    constraint FK_CATEGORIA
        foreign key (id_categoria) references nk_tb_categoria
(id_categoria),
    constraint FK_PRODUTO
        foreign key (id_produto) references nk_tb_produto
(id_produto)
);

create table nk_tb_usuario
```

```
(
    id_usuario    BIGINT        not NULL        AUTO_INCREMENT
        primary key,
    ds_email       VARCHAR(30) not null,
    nm_usuario     VARCHAR(30) not null,
    ds_senha       VARCHAR(30) not null,
    nm_sobrenome    VARCHAR(30) not null,
    constraint UK_EMAIL
        unique (ds_email)
);

create table nk_tb_carrinho
(
    id_carrinho    BIGINT not NULL                AUTO_INCREMENT
        primary key,
    id_produto     bigint not null,
    id_usuario     bigint not null,
    constraint FK_PRODUTO_CARRINHO
        foreign key (id_produto) references nk_tb_produto
(id_produto),
    constraint FK_USUARIO_CARRINHO
        foreign key (id_usuario) references nk_tb_usuario
(id_usuario)
);

create table sq_categoria
(
    next_not_cached_value bigint(21)                not null,
    minimum_value         bigint(21)                not null,
    maximum_value         bigint(21)                not null,
    start_value           bigint(21)                not null,
    increment             bigint(21)                not null,
    cache_size            bigint(21) unsigned not NULL,
    cycle_option          tinyint(1) unsigned not null,
    cycle_count           bigint(21)                not null,
);

create table sq_produto
(
    next_not_cached_value bigint(21)                not null,
    minimum_value         bigint(21)                not null,
    maximum_value         bigint(21)                not null,
    start_value           bigint(21)                not null,
    increment             bigint(21)                not null,
    cache_size            bigint(21) unsigned not null,
    cycle_option          tinyint(1) unsigned not null,
    cycle_count           bigint(21)                not null,
);

create table sq_usuario
```

```
(
  next_not_cached_value bigint(21)          not null,
  minimum_value          bigint(21)          not null,
  maximum_value          bigint(21)          not null,
  start_value            bigint(21)          not NULL,
  increment              bigint(21)          not NULL,
  cache_size             bigint(21) unsigned not null,
  cycle_option           tinyint(1) unsigned not NULL,
  cycle_count            bigint(21)          not null
);
```

Código-fonte 1 – DDL da tabela ug_empresa
Fonte: Elaborado pelos autores (2020)

6 – Data Manipulation Language – DML (INSERT)

```
INSERT INTO UG_EMPREENDEDOR  
VALUES ('Nome', '455.888.256-66', '11353535555',  
'email@email.com', 'masculino', '12/04/2000', 1);
```

Script em SQL para carga inicial do banco de dados na tabela empreendedor

7 – Data Manipulation Language – DML (UPDATE / DELETE)

```
UPDATE NK_TB_USUARIO  
SET DS_SENHA = 'P45sW0Rd'  
WHERE ID_USUARIO = 3;
```

```
DELETE FROM NK_TB_USUARIO  
WHERE ID_USUARIO = 5;
```

Comandos de Atualização e de Deleção

8 – Data Query Language – DQL (SELECT)

8.1 – Relatório simples contendo apenas uma tabela com dados ordenados.

1- Relatório com listagem de produtos com seus preços em ordem crescente;

2- Relatório que retorna o nome do cliente que comprou determinado produto e a quantidade;

3- Relatório que retorna todos os dados de acordo com as compras feitas pelos 5 clientes.

```
-- 1

SELECT  NR_PRECO,  NM_PRODUTO  FROM  NK_TB_PRODUTO  ORDER  BY
NR_PRECO  ASC;

-- 2

SELECT      NK_TB_USUARIO.nm_usuario      AS      nm_usuario,
NK_TB_PRODUTO.nm_produto, COUNT(*) AS QUANTIDADE
FROM NK_TB_CARRINHO
INNER JOIN  NK_TB_USUARIO  ON  NK_TB_CARRINHO.id_usuario  =
NK_TB_USUARIO.id_usuario
INNER JOIN  NK_TB_PRODUTO  ON  NK_TB_CARRINHO.id_produto  =
NK_TB_PRODUTO.id_produto
GROUP BY NK_TB_USUARIO.nm_usuario, NK_TB_PRODUTO.nm_produto;

-- 3

SELECT NK_TB_USUARIO.*, NK_TB_CARRINHO.*, NK_TB_PRODUTO.*
FROM NK_TB_USUARIO
INNER JOIN  NK_TB_CARRINHO  ON  NK_TB_USUARIO.id_usuario  =
NK_TB_CARRINHO.id_usuario
INNER JOIN  NK_TB_PRODUTO  ON  NK_TB_CARRINHO.id_produto  =
NK_TB_PRODUTO.id_produto;
```