



unopar

UNIVERSIDADE PITÁGORAS UNOPAR ANHANGUERA - MARAVILHA  
ENGENHARIA DE SOFTWARE

NATAN OGLIARI - 34466876

PROGRAMAÇÃO E DESENVOLVIMENTO DE BANCO DE DADOS

Maravilha/SC

2023

NATAN OGLIARI - 34466876

## PROGRAMAÇÃO E DESENVOLVIMENTO DE BANCO DE DADOS

Trabalho de portfólio apresentado como requisito parcial para a obtenção de pontos para a média semestral.

Orientador: Anderson Emidio de Macedo Gonçalves.

Maravilha/SC  
2023

# Sumário

	<b>Páginas</b>
<b>1   Introdução</b>	<b>4</b>
<b>2   Desenvolvimento</b>	<b>5</b>
<b>3   Método</b>	<b>6</b>
<b>4   Conclusões</b>	<b>10</b>

# 1 Introdução

Figura 1. Logo SQL

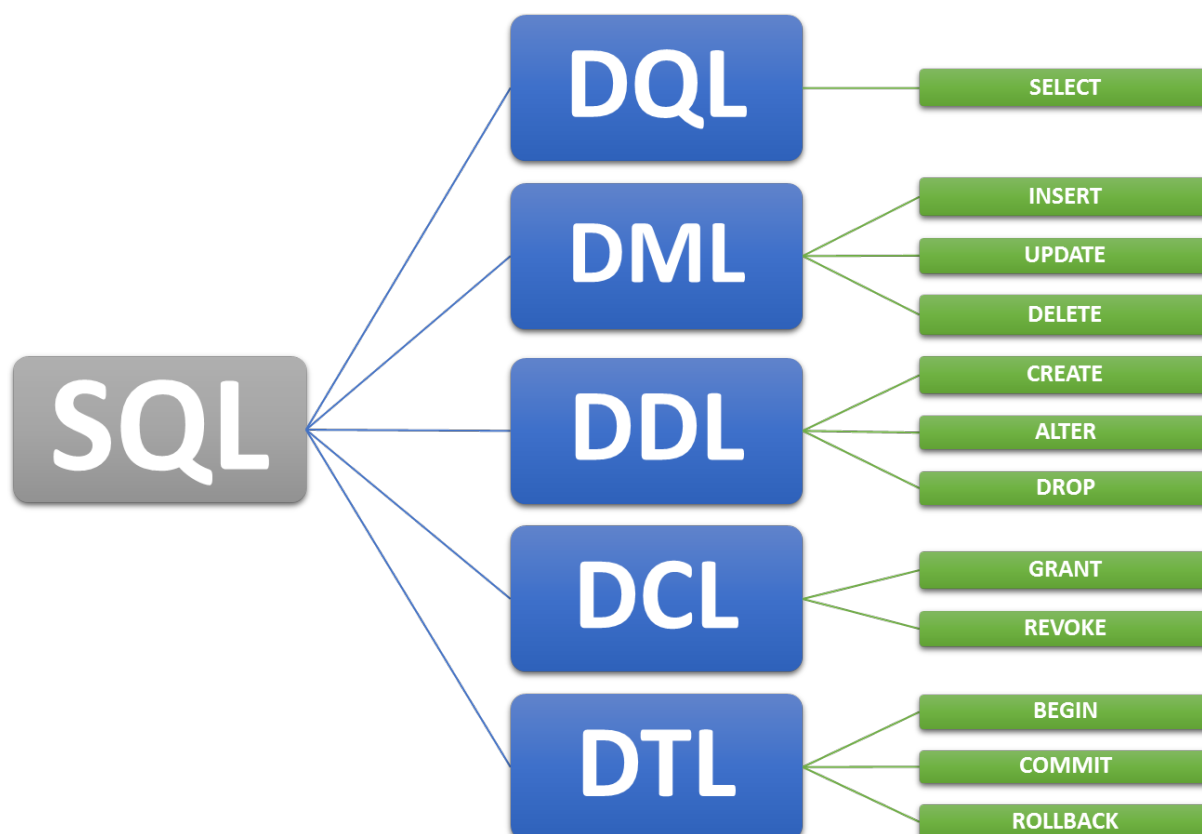
Na disciplina de programação e desenvolvimento de banco de dados é apresentado aos discentes a linguagem **SQL** *Structured Query Language*<sup>1</sup>, para a manipulação desta linguagem é proposto o software **MySQL da Workbench**



Fonte: Asnastasia (2023)

Na figura 2, expõe a divisão da linguagem **SQL**, a mesma é dividida em cinco subconjuntos, sendo eles: **DQL**, **DML**, **DDL**, **DCL** e **DTL**, cada uma com suas respectivas funções. por exemplo a **DQL**, é a linguagem de consulta de dados, definida pelo comando *SELECT*, ao qual possibilita a consulta do dados armazenados no banco de dados.

Figura 2. Subdivisões da Linguagem SQL



Fonte: DevMedia (2023)

Para esta aula prática é proposto o uso da **DDL**, Linguagem de definição de dados, a qual define os comandos *CREATE*, *ALTER* e *DROP*, sendo elas na sequência, Criação de tabelas, visualizações e índices; Alteração das estruturas e a remoção das estruturas criadas.

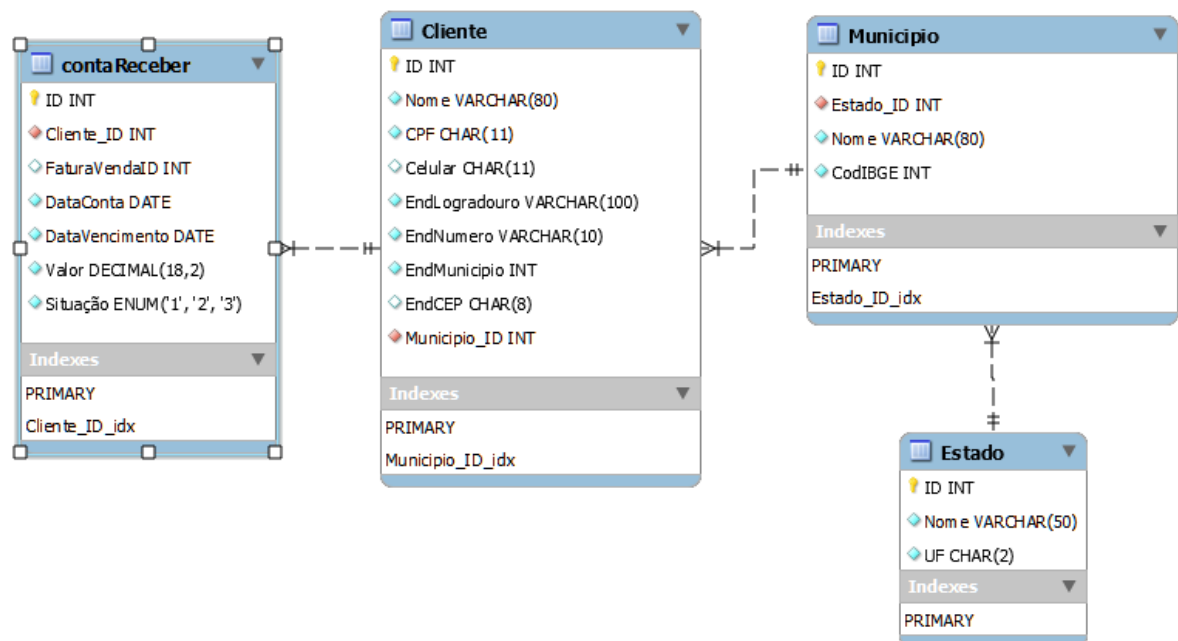
## 2 Desenvolvimento

Para implementação desta aula prática foram estabelecidas algumas regras informadas no roteiro da aula prática. sendo a atividade proposta:

- Criar uma estrutura de um banco de dados com a linguagem **SQL** por meio de uma entidade-relacionamento pré-definido;
- Inserir dados no banco de dados criado;
- Consultar os dados armazenados por meio da criação de uma visão (*View*);
- Elaborar um relatório no final da atividade;

Na atividade proposta o relatório dispõe de alguns procedimentos para a realização da atividade. Sugere a criação de uma base de dados de uma loja com o nome do banco de **Loja\_1**, com a utilização de definições de dados **DDL 1** da linguagem SQL, e respeitando o modelo definido no **DER**, porposto pela atividade conforme figura 3.

**Figura 3.** Diagrama entidade relacionamento



Fonte: O autor (2023).

Uma observação importante a qual deparei no desenvolvimento desta atividade, foi que já possuía instalado o **MySQL da Workbench**, no entanto, não estava configurado o SQL Server, portanto tive que configurar o mesmo.

### 3 Método

Após a criação do projeto através do software **MySQL da Workbench**, foi proseguido com a criação do banco de dados conforme figura 3, com o estabelecimento de chaves primárias e as indicações de elementos não nulos e auto incrementos, são quatro tabelas a tabela *contaReceber*, possui um elemento chamado de *Situação ENUM('1', '2', '3')*, sendo: 1 - conta registrada, 2 - conta cancelada e 3 - conta paga.

Deste modo é elaborado o *scripty* inserir dados conforme lista 2, por segurança utilizo o especificador de banco o *USE Loja\_1*.

Para a criação do Banco de dados **Loja\_1**, utilizei o metodo shell, utlizei o scripty??.

```
1 -- Criação do banco de dados Loja_1
2 -- Autor Natan Ogliari
3 CREATE DATABASE Loja_1
4     DEFAULT CHARSET = utf8
5     DEFAULT COLLATE = utf8_general_ci;
6
7 -- Expecifica o uso de Loja_1
8 USE Loja_1;
9
10 -- Cria a tabela 'estado', caso nao existe
11 CREATE TABLE IF NOT EXISTS Estado(
12     ID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
13     Nome VARCHAR(50) NOT NULL DEFAULT '',
14     UF CHAR(2) NOT NULL DEFAULT ''
15 );
16
17 -- Cria a tabela 'municipio', caso nao existe
18 CREATE TABLE IF NOT EXISTS Municipio(
19     ID INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
20     Estado_ID INT NOT NULL,
21     Nome VARCHAR(80) NOT NULL,
22     CodIBGE INT NOT NULL,
23     CONSTRAINT fk_Municipio_Estado1
24     FOREIGN KEY (Estado_ID)
25     REFERENCES Estado(ID)
26 );
27
28 -- Cria a tabela 'cliente', caso nao existe
29 CREATE TABLE IF NOT EXISTS cliente(
30     ID INT PRIMARY KEY AUTO_INCREMENT,
31     Nome VARCHAR(80) NOT NULL,
```

```

32     CPF CHAR(11) NOT NULL,
33     Celular CHAR(11),
34     EndLogradouro VARCHAR(100) NOT NULL,
35     EndNumero VARCHAR(10) NOT NULL,
36     EndMunicipio INT NOT NULL,
37     EndCEP CHAR(8),
38     Municipio_ID INT NOT NULL,
39     KEY fk_Cliente_Municipioid_idx_idx (Municipio_ID),
40     CONSTRAINT fk_Cliente_Municipioid_idx
41     FOREIGN KEY (Municipio_ID)
42     REFERENCES municipio('ID')
43 );
44
45 -- Cria a tabela 'ContaReceber', caso nao existe
46 CREATE TABLE IF NOT EXISTS ContaReceber(
47     ID INT PRIMARY KEY AUTO_INCREMENT,
48     FaturaVendaID INT,
49     DataConta DATE NOT NULL,
50     DataVencimento DATE NOT NULL,
51     Valor DECIMAL(18,2) NOT NULL,
52     -- Situação ENUM('1', '2', '3') NOT NULL,
53     Cliente_ID INT NOT NULL,
54     CONSTRAINT fk_ContaReceber_Cliente
55     FOREIGN KEY (Cliente_ID)
56     REFERENCES Cliente(ID)
57 );
58
59 -- Adicionando 'Situação' a tabela 'contaReceber'
60 ALTER TABLE ContaReceber
61     ADD Situação ENUM('1', '2', '3') NOT NULL
62 ;

```

**Listagem 1.** *Criação\_Loja\_1.sql*

No scripty 1, a linha **52** está em comentário pois, acabei esquecendo de adicionar a linha *Situação*, e para a correção exponho as linhas **59 a 62**, para a correção deste esquecimento. Neste ponto aproveito o aprendizado para a manipulação das tabelas com o comando **ALTER TABLE ... ADD**.

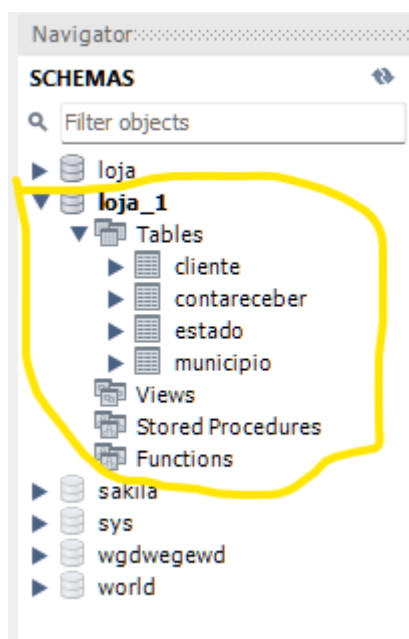
Na Figura 5, expõe a captura de tela do banco de dados criado no MySQL Workbench.

**Figura 4.** Captura de tela da criação do banco de dados.

Output			
Action Output			
#	Time	Action	Message
5	09:16:29	CREATE DATABASE Loja_1 DEFAULT CHARSET = utf8 DEFAULT COLLATE = utf8_general_ci	Error Code: 1007. Can't create database 'loja_1': database exists
6	09:16:36	CREATE TABLE IF NOT EXISTS Estado( ID INT PRIMARY KEY NOT NULL AUTO_INCREMENT, Nome VARCHAR(50) NOT NULL DEFAULT "", ...	0 row(s) affected, 1 warning(s): 1050 Table 'estado' already exists
7	09:16:38	CREATE TABLE IF NOT EXISTS Municipio( ID INT PRIMARY KEY NOT NULL AUTO_INCREMENT, Estado_ID INT NOT NULL, Nome VARCH...	0 row(s) affected, 1 warning(s): 1050 Table 'municipio' already exists
8	09:23:47	CREATE TABLE IF NOT EXISTS cliente( ID INT PRIMARY KEY AUTO_INCREMENT, Nome VARCHAR(80) NOT NULL, CPF CHAR(11) NOT N...	0 row(s) affected
9	09:28:11	CREATE TABLE IF NOT EXISTS ContaReceber( ID INT PRIMARY KEY AUTO_INCREMENT, FaturaVendaID INT, DataConta DATE NOT NUL...	Error Code: 1824. Failed to open the referenced table 'clienet'
10	09:28:30	CREATE TABLE IF NOT EXISTS ContaReceber( ID INT PRIMARY KEY AUTO_INCREMENT, FaturaVendaID INT, DataConta DATE NOT NUL...	0 row(s) affected

Fonte: O autor (2023).

**Figura 5.** Criação do banco de dados.



Fonte: O autor (2023).

```

1  -- Scripty para inserir dados no banco Loja_1
2  USE Loja_1;
3
4  -- INSERINDO VALOR NA TABELA 'estado'
5  INSERT INTO
6  estado (Nome,UF) VALUES ('Santa Catarina', 'SC');
7  INSERT INTO
8  estado (Nome,UF) VALUES ('Santa Catarina', 'SC');
9  INSERT INTO
10 estado (Nome,UF) VALUES ('Santa Catarina', 'SC');
11
12 -- INSERINDO VALOR NA TABELA 'municipio'
13 INSERT INTO
14 municipio (Estado_ID, Nome, CodIBGE) VALUES ( 1, 'Chinelo Queimado',

```



```

1254786);
15 INSERT INTO
16 municipio (Estado_ID, Nome, CodIBGE) VALUES ( 2, 'Sanga Silva',
    1934986);
17 INSERT INTO
18 municipio (Estado_ID, Nome, CodIBGE) VALUES ( 3, 'Saracura', 2834583)
    ;
19
20 -- INSERINDO VALOR NA TABELA 'cliente'
21 INSERT INTO
22 cliente (Nome, CPF, Celular, EndLogradouro,EndNumero,EndMunicipio,
    EndCEP,Municipio_ID) VALUES ('Na9
23 tan Ogliari', 85698753695, 23651225, 'Banhado seco', 120, 3,
    89870000, 3);
24 INSERT INTO
25 cliente (Nome, CPF, Celular, EndLogradouro,EndNumero,EndMunicipio,
    EndCEP,Municipio_ID) VALUES ('Francisco Setembrino', 89658653694,
    23556225, 'Jundia', 180, 2, 89870000, 2);
26 INSERT INTO
27 cliente (Nome, CPF, Celular, EndLogradouro,EndNumero,EndMunicipio,
    EndCEP,Municipio_ID) VALUES ('Nego veio', 85698753695, 23651225, '
    Pingo preto', 360, 1, 89870000, 1);
28
29 -- Inserindo valor na tabela 'ContaReceber'
30 INSERT INTO
31 contareceber(Cliente_ID, FaturaVendaID, DataConta, DataVencimento ,
    Valor, Situação) VALUES (1,1,'2023-11-10','2023-06-10',350.00,1);
32 INSERT INTO
33 contareceber(Cliente_ID, FaturaVendaID, DataConta, DataVencimento ,
    Valor, Situação) VALUES (2,2,'2023-11-10','2023-06-10',300.00,1);
34 INSERT INTO
35 contareceber(Cliente_ID, FaturaVendaID, DataConta, DataVencimento ,
    Valor, Situação) VALUES (2,2,'2023-10-10','2023-10-10',450.00,1);

```

## **Listagem 2. inserir.sql**

*Scripty para consulta de dados no banco de dados Loja\_1.*

```

1 -- Scripty Consulta
2 -- autor: Natan Ogliari
3 SELECT ContaReceber.ID,
4 cliente.Nome,
5 cliente.CPF,
6 ContaReceber.DataVencimento,

```

```

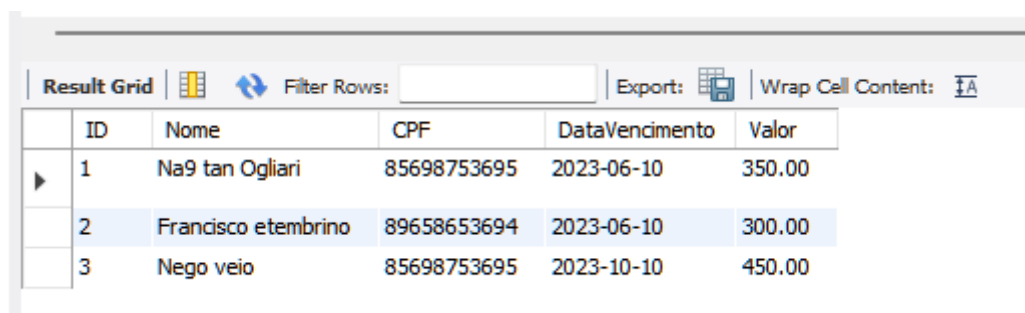
7 ContaReceber.Valor
8 FROM loja_1.ContaReceber
9 INNER JOIN
10 loja_1.cliente ON ContaReceber.ID = cliente.ID
11 WHERE
12 ContaReceber.Situação = 1;

```

**Listagem 3.** *consulta.sql*

O resultado do scripty *consulta.sql* é demonstrado na figura 6.

**Figura 6.** *Visualização dos dados consultados*



The image shows a screenshot of a database management system's query result grid. The grid has a toolbar at the top with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with six columns: ID, Nome, CPF, DataVencimento, and Valor. There are three rows of data displayed.

ID	Nome	CPF	DataVencimento	Valor
1	Na9 tan Ogliari	85698753695	2023-06-10	350.00
2	Francisco etembrino	89658653694	2023-06-10	300.00
3	Nego veio	85698753695	2023-10-10	450.00

Fonte: O autor (2023).

Desta forma é possível e necessario a criação de scripty para a manipulação dos dados armazenados.

## 4 Conclusões

A criação de um *database*, parte de princípio melhora no gerenciamento de dados de uma organização, e a criação de scripty de manipulação é necessário devido as suas progressivas e sucessivas necessidades de alterações, adicção, remoção e atualizações.

## Referências

ASNASTASIA. **ícone SQL**. 2023. Acessado em: 20 out. 2023. Disponível em: <<https://pt.dreamstime.com/ilustra%C3%A7%C3%A3o-stock-%C3%ADcone-logo-design-ui-ou-ux-app-do-base-de-dados-do-sql-image96841987>>.

DEVMEDIA. **Guia Completo de SQL**. 2023. Acessado em: 20 out. 2023. Disponível em: <<https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>>.