**Aula II**

**Objetivo**

Definir a padronização a ser utilizada no desenvolvimento do Banco de Dados, no nome das tabelas, campos e chaves primárias e chaves estrangeiras. Apresentar os comandos mais utilizados de DDL e DML

**Padronização**

* Evitar o uso de abreviação para definir nome de tabela e/ou campos;
* Nome de tabela e/ou campos deve ser grafado em minúsculo;
* Nome da tabela: palavra simples ou composta no singular, quando composta, deve ser separada por *underline* (\_);
* A chave primária gerada automaticamente pelo BD deve conter o termo cod\_NOME\_TABELA;
* A chave estrangeira terá o mesmo nome da chave primária da tabela de origem;
* Comandos SQL serão escrito em maiúsculo e o nome de objetos em minúsculo;
* No relacionamento 1:N a tabela que tiver N receberá o atributo(chave primária) da tabela 1, a qual se torna chave estrangeira;

**Structured Query Language , ou Linguagem de Consulta Estruturada ou SQL**

DDL - Data Definition Language: são usadas para definir a estrutura de banco de dados ou esquema. Alguns exemplos:

* ALTER – altera a estrutura da base de dados
* CREATE- para criar objetos no banco de dados
* DROP – remover um objeto
* RENAME – para renomear um objeto
* TRUNCATE – remover todos os registros de uma tabela, incluindo todos os espaços alocados para os registros são removidos

DML - Data Manipulation Language: são utilizados para o gerenciamento de dados dentro de objetos do banco. Alguns exemplos:

* SELECT- recuperar dados do banco de dados
* INSERT – inserir dados em uma tabela
* UPDATE – atualiza os dados existentes em uma tabela
* DELETE – exclui registros de uma tabela

**Aula**

**Objetivo**

Introduzir os comandos CREATE DATABASE, CREATE TABLE, INSERT E SELECT. Para tanto será utilizado: psql -d nome\_db –U postgres para conexão de linha de comando.

**Comandos**

-- Conexão na **base de dados postgres** com as credenciais do usuário postgres

psql -d postgres -U postgres

-- Criar o banco de dados aula

**-- Comando DDL**

CREATE DATABASE aula;

-- Fecha conexão

\q

-- Conexão na **base de dados aula** com as credenciais do usuário postgres

psql -d aula -U postgres

-- Executar comandos na Base de Dados aula

-- Criar tabela Cliente

**-- Comando DDL**

CREATE TABLE cliente(

cod\_cliente integer,

 nome varchar(80),

cpf varchar(11),

data\_nascimento date);

-- Criar tabela Veiculo

**-- Comando DDL**

CREATE TABLE veiculo(

 cod\_veiculo integer,

 placa varchar(8),

 marca varchar(20),

 modelo varchar(30));

-- Inserir dados na tabela Cliente

**-- Comando DML**

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(22, ‘Maria’, ‘11111111111’;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(22, ‘João, ‘22222222222’;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(33, ‘Bruno, ‘33333333333;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(44, ‘Francisco, ‘44444444444;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(55, ‘Andre, ‘5555555555;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(66, ‘Marcos, ‘6666666666;

-- Consultar dados da tabela cliente

**-- Comando DML**

SELECT nome

FROM cliente;

SELECT nome, cpf

FROM cliente;

SELECT cpf, nome

FROM cliente;

SELECT nome, cpf

FROM cliente

ORDER by nome;

SELECT nome, cpf

FROM cliente

WHERE cpf= ‘44444444444’;

**Aula IV**

**Objetivo**

Introduzir os comandos DELETE, DROP TABLE, DROP DATABASE, ALTER TABLE. Utilizar os comandos vistos na aula anterior CREATE DATABASE, CREATE TABLE, INSERT E SELECT. Para tanto será utilizado o PSQL e o pgAdminIII

**Comandos**

-- Conexão na **base de dados aula** com as credenciais do usuário postgres

psql -d aula -U postgres

SELECT \*

FROM veiculo;

SELECT \*

FROM cliente;

DELETE FROM cliente;

DROP TABLE cliente;

DROP TABLE veiculo;

DROP DATABASE AULA;

DROP DATABASE AULA1;

DROP DATABASE AULA2;

-- Conexão na **base de dados postgres** com as credenciais do usuário postgres

psql -d postgres -U postgres

-- Criar o banco de dados aula

**-- Comando DDL**

CREATE DATABASE aula;

-- Fecha conexão

\q

-- Conexão na **base de dados aula** com as credenciais do usuário postgres

psql -d aula -U postgres

-- Fecha conexão

\q

CREATE SCHEMA estacionamento;

* CRIAR as tabelas Cliente e Veiculo da aula anterior, no schema estacionamento
* INSERIR os valores nas tabelas conforme aula anterior, no schema estacionamento

-- Altera a tabela veiculo adicionando o campo cod\_cliente

ALTER TABLE veiculo

 ADD COLUMN cod\_cliente integer;

-- Inserir dados na tabela Cliente

**-- Comando DML**

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(22, ‘Maria’, ‘11111111111’;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(22, ‘João, ‘22222222222’;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(33, ‘Bruno, ‘33333333333;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(44, ‘Francisco, ‘44444444444;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(55, ‘Andre, ‘5555555555;

INSERT INTO cliente(cod\_cliente, nome,cpf) VALUES(66, ‘Marcos, ‘6666666666;

INSERT INTO veiculo (cod\_veiculo,placa,marca,modelo, cod\_cliente) VALUES(1, ‘nda-1111’, ‘fiat’, ‘uno’, 11);

INSERT INTO veiculo (cod\_veiculo,placa,marca,modelo, cod\_cliente) VALUES(2, ‘ndz-2222’, ‘toyota’, ‘corolla’, 22);

INSERT INTO veiculo (cod\_veiculo,placa,marca,modelo, cod\_cliente) VALUES(3, ‘ndw-3333’, ‘fiat’, ‘palio’, 33);

INSERT INTO veiculo (cod\_veiculo,placa,marca,modelo, cod\_cliente) VALUES(5, ‘axz-5555’, ‘fiat’, palio, 55);

INSERT INTO veiculo (cod\_veiculo,placa,marca,modelo, cod\_cliente) VALUES(6, ‘axz-6666’, ‘fiat’, siena, 77);

ALTER TABLE veiculo

 ADD CONSTRAINT pk\_veiculo PRIMARY KEY(cod\_veiculo);

ALTER TABLE veiculo

 ADD CONSTRAINT fk\_cliente FOREIGN KEY (cod\_cliente)

 REFERENCES cliente (cod\_cliente);