CREATE DATABASE constraints;

USE constraints;

# 1 - NOT NULL

CREATE TABLE pessoas(

nome VARCHAR(100) NOT NULL,

idade INT

);

INSERT INTO pessoas (nome) VALUES ("Matheus");

SELECT \* FROM pessoas;

/\*Teste em banco do dados - tento inserir um valor

"NULL" em um atributo que não aceita NULL\*/

INSERT INTO pessoas (nome, idade) VALUES (NULL, 30);

INSERT INTO pessoas (nome, idade) VALUES ("João",NULL);

INSERT INTO pessoas (nome) VALUES ("Maria");

INSERT INTO pessoas (idade) VALUES (18);

#2 - UNIQUE

**ALTER TABLE pessoas ADD COLUMN email VARCHAR(255) UNIQUE;**

INSERT INTO pessoas (nome, idade, email) VALUES ("João", 30, "joao@gmail.com");

INSERT INTO pessoas (nome, idade, email) VALUES ("João Paulo", 25, "joao@gmail.com");

INSERT INTO pessoas (nome, idade, email) VALUES ("João Paulo", 25, "joaop@gmail.com");

# 3 - PRIMARY KEY

CREATE TABLE produtos(

id INT NOT NULL,

nome VARCHAR(255),

sku VARCHAR(10),

PRIMARY KEY (id)

);

DESC produtos;

SELECT \* FROM produtos;

INSERT INTO produtos (id, nome, sku) VALUES (1, 'Batedeira', '123asd');

INSERT INTO produtos (id, nome, sku) VALUES (NULL, 'Fogão', '1244');

INSERT INTO produtos (id, nome, sku) VALUES (1, 'TV', '555');

INSERT INTO produtos (id, nome, sku) VALUES (2, 'TV', '555');

INSERT INTO produtos (id, nome, sku) VALUES (2, 'TV', '555');

INSERT INTO produtos (id, nome, sku) VALUES (3, 'TV 20', '555');

/\*O SKU é uma codigo que não poderia repetir\*/

**ALTER TABLE produtos MODIFY COLUMN sku VARCHAR(10) UNIQUE;**

/\*Se tentar alterar a tabela que já tem registros repetidos

irá retornar um erro ao tentar alterar a tabela\*/

DELETE FROM produtos WHERE id=3;

/\*Apos apagar o registro que repete o SKU, podemos alterar a tabela colocando SKU como UNIQUE\*/

ALTER TABLE produtos MODIFY COLUMN sku VARCHAR(10) UNIQUE;

# 4 - AUTO INCREMENT

CREATE TABLE frutas (

id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nome VARCHAR(200)

);

DESC frutas;

INSERT INTO frutas (nome) VALUES ("Maçã");

INSERT INTO frutas (nome) VALUES ("Mamão");

INSERT INTO frutas (nome) VALUES ("Morango");

INSERT INTO frutas (nome) VALUES ("Laranja");

DELETE FROM frutas WHERE id = 5;

SELECT \* FROM frutas;

# 5 - FOREIGN KEY

DROP TABLE pessoas;

CREATE TABLE pessoas(

id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nome VARCHAR(100) NOT NULL,

idade INT

);

INSERT INTO pessoas(nome,idade) VALUES ("juliano", 33);

ALTER TABLE pessoas ADD COLUMN email VARCHAR(40) NOT NULL UNIQUE;

ALTER TABLE pessoas MODIFY COLUMN email VARCHAR(255) NOT NULL UNIQUE;

DESC pessoas;

SELECT \* FROM pessoas;

UPDATE pessoas SET email = "julianoqm@gmail.com"

WHERE id=1;

INSERT INTO pessoas(nome,idade) VALUES ("Juliano", 33,"julianoqm@gmail.com");

/\*UPDATE altera o registro, ALTER TABLE altera a tabela\*/

UPDATE pessoas SET nome = "Juliano", idade=30, email="julianoqm2@gmail.com"

WHERE id=1;

CREATE TABLE enderecos (

id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

rua VARCHAR(100),

numero VARCHAR(10),

pessoa\_id INT NOT NULL,

FOREIGN KEY (pessoa\_id) REFERENCES pessoas(id)

);

ALTER TABLE enderecos MODIFY COLUMN pessoa\_id INT;

DESC enderecos;

INSERT INTO enderecos (rua, numero, pessoa\_id) VALUES ("Rua Passado", "25", 2);

INSERT INTO pessoas (nome, idade) VALUES ("Pedro", 32);

INSERT INTO enderecos (rua, numero) VALUES ("Rua São Paulo", "1352");

SELECT \* FROM pessoas;

SELECT \* FROM enderecos;

DELETE FROM endereco WHERE id = 9;

DELETE FROM enderecos WHERE id>=1;

# 6 - INDEX

CREATE INDEX index\_title

ON titles(title);

SHOW INDEX FROM titles;

# 7 - DELETE INDEX

DROP INDEX index\_nome ON pessoas;

# EXERCICIO 21

CREATE DATABASE banco;

USE banco;

CREATE TABLE contas (

id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nome VARCHAR(100),

sobrenome VARCHAR(100),

saldo DOUBLE(10,2),

data\_nascimento DATE

);

ALTER TABLE contas ADD COLUMN cpf VARCHAR(11) NOT NULL UNIQUE;

ALTER TABLE contas MODIFY COLUMN cpf VARCHAR(12) NOT NULL UNIQUE;

ALTER TABLE contas DROP COLUMN cpf;

DESC contas;

CREATE INDEX index\_nome

ON contas (nome);

INSERT INTO contas (nome, sobrenome, saldo, data\_nascimento) VALUES

("Juliano", "Fernando", 10000, "1989-06-05"),

("Bruna", "Sousa", 100000, "19821215");

SELECT \* FROM contas;

#JOIN

SELECT \* FROM employees;

SELECT \* FROM salaries;

/\*Note que, em ambas tabelas os valores de emp\_no repetem\*/

DESC employees;

DESC salaries;

-- Comando --

/\*

SELECT tabela.coluna, tabela.coluna,tabela.coluna

FROM tabela1

INNER JOIN tabela2

ON tabela1.colunaChave = tabela2.colunaChave;

\*/

SELECT employees.first\_name AS Nome,employees.last\_name AS Sobrenome, salaries.salary AS "Salário", employees.birth\_date AS "Aniversário"

FROM employees

INNER JOIN salaries

ON employees.emp\_no = salaries.emp\_no

WHERE salaries.salary >= 120000

ORDER BY employees.first\_name ASC;

# Exercício 22

SELECT employees.first\_name AS Nome, employees.gender AS Genero, titles.title AS Cargo

FROM employees

INNER JOIN titles

ON employees.emp\_no = titles.emp\_no;

#LEFT JOIN

USE constraints;

SELECT \* FROM pessoas;

SELECT \* FROM enderecos;

SELECT pessoas.nome,pessoas.id, enderecos.\*

FROM pessoas

LEFT JOIN enderecos

ON pessoas.id = enderecos.pessoa\_id;

-- Retorna todos os registros de pessoas, mesmo que o campo pessoa\_id esteja NULL

INSERT INTO pessoas(nome,email)

VALUES ("Julia","julia@gmail.com");

SELECT \* FROM enderecos; -- Não inseri o ID da JULIA na tabela endereco

SELECT pessoas.nome,pessoas.id, enderecos.\*

FROM pessoas

LEFT JOIN enderecos

ON pessoas.id = enderecos.pessoa\_id;

-- LEFT JOIN na base EMPLOYEES

USE employees;

SELECT employees.first\_name, employees.last\_name, salaries.salary AS salario

FROM employees

LEFT JOIN salaries

ON employees.emp\_no = salaries.emp\_no;

#RIGHT JOIN

SELECT \* FROM pessoas;

SELECT \* FROM enderecos;

SELECT pessoas.nome, enderecos.rua

FROM enderecos

RIGHT JOIN pessoas

ON pessoas.id = enderecos.pessoa\_id;

#INNER JOIN EM VARIAS TABELAS

USE employees;

SELECT employees.first\_name AS NOME, salaries.salary AS SALARIO, titles.title AS Cargo

FROM employees

INNER JOIN salaries

ON employees.emp\_no = salaries.emp\_no

INNER JOIN titles

ON employees.emp\_no = titles.emp\_no;