Autarquia Educacional do Vale do São Francisco

Faculdade de Ciências Aplicadas e Sociais de Petrolina

Curso de Gestão de TI

**Projeto banco de dados - Mercado**

Banco de Dados 2

Pedro Eduardo

Luis Eduardo da Silva

**Professor:** Emanuella Vidal

Petrolina, 2019

**SUMÁRIO**

1. DIAGRAMA 4

2. PROJETO FÍSICO 5

2.1. CRIAÇÃO DE DOMÍNIOS 5

2.1.1. (Dizer o que cada domínio irá fazer) 5

2.2. CRIAÇÃO DE TABELAS 6

2.3. ALTERAÇÃO DE TABELAS 7

2.3.1. (Dizer o que cada alteração irá fazer) 7

2.3.2. 7

2.3.3. 7

2.3.4. 7

2.3.5. 7

2.4. INSERÇÃO DE DADOS NAS TABELAS 8

2.4.1. (Dizer o que cada comando irá fazer) 8

2.4.2. 8

2.4.3. 8

2.4.4. 8

2.4.5. 8

2.5. ATUALIZAÇÃO DE DADOS DAS TABELAS 9

2.5.1. (Dizer o que cada atualização irá fazer) 9

2.5.2. 9

2.5.3. 9

2.5.4. 9

2.5.5. 9

2.6. EXCLUSÃO DE DADOS DAS TABELAS 10

2.6.1. (Dizer o que cada exclusão irá fazer) 10

2.6.2. 10

2.6.3. 10

2.6.4. 10

2.6.5. 10

2.7. VISÕES 11

2.7.1. (Dizer o que cada visão esta fazendo) 11

2.7.2. 11

2.7.3. 11

2.7.4. 11

2.7.5. 11

2.8. Procedimentos 12

2.8.1. (Dizer o que cada procedimento irá fazer) 12

2.8.2. 12

2.8.3. 12

2.8.4. 12

2.8.5. 12

2.9. Exceções 13

2.9.1. (Dizer o que cada exceção irá fazer) 13

2.9.2. 13

2.10. Gatilhos 14

2.10.1. (Dizer o que cada gatilho irá fazer) 14

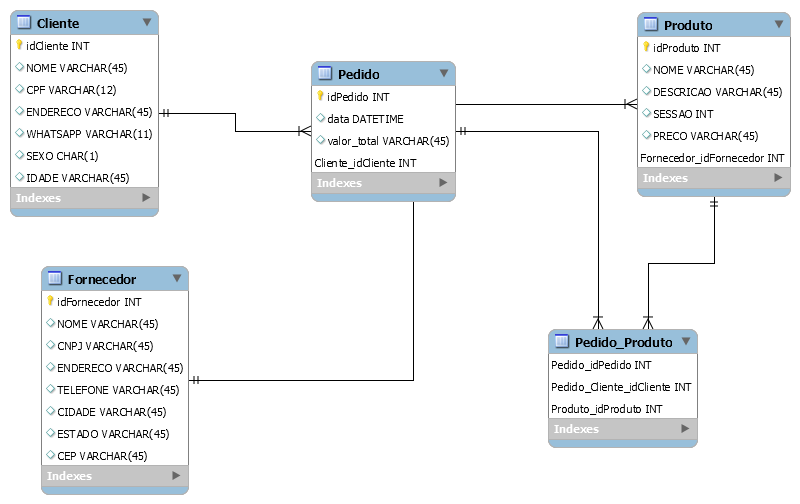
2.10.2. 14

2.10.3. 14

2.10.4. 14

2.10.5. 14

# DIAGRAMA



# PROJETO FÍSICO

### CRIAÇÃO DE DOMÍNIOS

### 2.0.1. create domain "DM\_sexo" as char (1);

2.0.2. create domain "DM\_endereco" as varchar(40)

2.0.3. create domain "DM\_Valor" as numeric(10,2);

2.0.4. create domain "DM\_nome" as varchar(40);

2.0.5. create domain "DM\_codigo" as integer not null;

### CRIAÇÃO DE TABELAS

TABELA FORNECEDORES

CREATE TABLE IF NOT EXISTS `BD\_mercado`.`Fornecedor` (

`idFornecedor` INT NOT NULL AUTO\_INCREMENT,

`NOME` VARCHAR(45) NULL,

`CNPJ` VARCHAR(45) NULL,

`ENDERECO` VARCHAR(45) NULL,

`TELEFONE` VARCHAR(45) NULL,

`CIDADE` VARCHAR(45) NULL,

`ESTADO` VARCHAR(45) NULL,

`CEP` VARCHAR(45) NULL,

PRIMARY KEY (`idFornecedor`))

ENGINE = InnoDB

TABELA CLENTES

CREATE TABLE IF NOT EXISTS `BD\_mercado`.`Cliente` (

`idCliente` INT NOT NULL AUTO\_INCREMENT,

`NOME` VARCHAR(45) NULL,

`CPF` VARCHAR(12) NULL,

`ENDERECO` VARCHAR(45) NULL,

`WHATSAPP` VARCHAR(11) NULL,

`SEXO` CHAR(1) NULL,

`IDADE` VARCHAR(45) NULL,

PRIMARY KEY (`idCliente`))

ENGINE = InnoDB

TABELA PRODUTOS

CREATE TABLE IF NOT EXISTS `BD\_mercado`.`Produto` (

`idProduto` INT NOT NULL AUTO\_INCREMENT,

`NOME` VARCHAR(45) NULL,

`DESCRICAO` VARCHAR(45) NULL,

`SESSAO` INT NULL,

`PRECO` VARCHAR(45) NULL,

`Fornecedor\_idFornecedor` INT NOT NULL,

PRIMARY KEY (`idProduto`, `Fornecedor\_idFornecedor`),

INDEX `fk\_Produto\_Fornecedor1\_idx` (`Fornecedor\_idFornecedor` ASC) VISIBLE,

CONSTRAINT `fk\_Produto\_Fornecedor1`

FOREIGN KEY (`Fornecedor\_idFornecedor`)

REFERENCES `BD\_mercado`.`Fornecedor` (`idFornecedor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

TABELA PEDIDO

CREATE TABLE IF NOT EXISTS `BD\_mercado`.`Pedido` (

`idPedido` INT NOT NULL AUTO\_INCREMENT,

`data` DATETIME NULL,

`valor\_total` VARCHAR(45) NULL,

`Cliente\_idCliente` INT NOT NULL,

PRIMARY KEY (`idPedido`, `Cliente\_idCliente`),

INDEX `fk\_Pedido\_Cliente\_idx` (`Cliente\_idCliente` ASC) VISIBLE,

CONSTRAINT `fk\_Pedido\_Cliente`

FOREIGN KEY (`Cliente\_idCliente`)

REFERENCES `BD\_mercado`.`Cliente` (`idCliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

TABELA PEDIDO\_PRODUTO

CREATE TABLE IF NOT EXISTS `BD\_mercado`.`Pedido\_Produto` (

`Pedido\_idPedido` INT NOT NULL,

`Pedido\_Cliente\_idCliente` INT NOT NULL,

`Produto\_idProduto` INT NOT NULL,

PRIMARY KEY (`Pedido\_idPedido`, `Pedido\_Cliente\_idCliente`, `Produto\_idProduto`),

INDEX `fk\_Pedido\_has\_Produto\_Produto1\_idx` (`Produto\_idProduto` ASC) VISIBLE,

INDEX `fk\_Pedido\_has\_Produto\_Pedido1\_idx` (`Pedido\_idPedido` ASC, `Pedido\_Cliente\_idCliente` ASC) VISIBLE,

CONSTRAINT `fk\_Pedido\_has\_Produto\_Pedido1`

FOREIGN KEY (`Pedido\_idPedido` , `Pedido\_Cliente\_idCliente`)

REFERENCES `BD\_mercado`.`Pedido` (`idPedido` , `Cliente\_idCliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Pedido\_has\_Produto\_Produto1`

FOREIGN KEY (`Produto\_idProduto`)

REFERENCES `BD\_mercado`.`Produto` (`idProduto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

### ALTERAÇÃO DE TABELAS

2.3.1. ALTER TABLE clientes ALTER email TO idade;

2.3.2. ALTER TABLE clientes ALTER telefone TO WhatsApp;

2.3.3. ALTER TABLE produto ALTER valor TO preco;

* + 1. ALTER TABLE fornecedores ALTER email TO codigo\_postal;
    2. ALTER TABLE funcionarios ALTER email TO instagram\_WhatsApp;

### INSERÇÃO DE DADOS NAS TABELAS

### 

2.4.1. insert into funcionarios (funcionarios.cod\_func, funcionarios.nome)values(1,'pedro');

2.4.2. insert into clientes (clientes.cod\_cliente, clientes.nome)values(2,'maria');

2.4.3. insert into clientes (clientes.cod\_cliente, clientes.nome)values(4,'jose');

2.4.4. insert into produto(produto.cod\_produto, produto.sessao)values(1,22);

2.4.5. insert into fornecedores(fornecedores.cod\_forn, fornecedores.nome)values(1,'edilson');

### ATUALIZAÇÃO DE DADOS DAS TABELAS

2.5.1. insert into funcionarios (funcionarios.cod\_func, funcionarios.nome)values(1,'pedro');

2.5.2. insert into clientes (clientes.cod\_cliente, clientes.nome)values(2,'maria');

2.5.3. insert into clientes (clientes.cod\_cliente, clientes.nome)values(4,'jose');

2.5.4. insert into produto(produto.cod\_produto, produto.sessao)values(1,22);

2.5.5. insert into fornecedores(fornecedores.cod\_forn, fornecedores.nome)values(1,'edilson');

### EXCLUSÃO DE DADOS DAS TABELAS

2.6.1. delete from clientes where clientes.nome like '%ana%';

2.6.2. delete from clientes where clientes.cod\_cliente = 4;

2.6.3. delete from funcionarios where funcionarios.cod\_func = 1;

2.6.4. delete from fornecedores where fornecedores.cod\_forn = 1;

2.6.5. delete from produto where produto.cod\_produto = 1;

### VISÕES

2.7.1 create view pedido\_cliente as

select clientes.cod\_cliente,clientes.nome, clientes.endereco, pedido.cod\_pedido, pedido.valor\_total

from clientes, pedido

where clientes.cod\_cliente = pedido.cod\_cliente;

2.7.2. create view nome as

select funcionarios.cod\_func,funcionarios. cod\_func, funcionarios.endereco

from clientes, funcionarios

where funcionarios.cod\_func = funcionarios.cod\_func;

2.7.3 create view nome as

select fornecedores.cod\_forn, fornecedores.nome,fornecedores.endereco

from fornecedores, clientes

where fornecedores.telefone = fornecedores.telefone;

2.7.4 create view endereco as

select funcionarios.cod\_func,funcionarios.nome,funcionarios.endereco

from clientes, funcionarios

where funcionarios.nome = funcionarios.nome;

2.7.5. create view endereco as

select fornecedores.cod\_forn, fornecedores.nome, fornecedores.endereco

from clientes, fornecedores

where fornecedores.nome =fornecedores.nome;