

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO

PROJETO:
PIZZA DELIVERY COM ENTRETENIMENTO

Participantes:

Lorena da Silva Elias
Victor Hugo Eustáquio Lopes
Yago Vinícius Ferreira de Castro

Professora:

Maria Camila Nardini Barioni

Matéria:

Sistemas de Banco de Dados

28 de junho de 2019

ÍNDICE

I.	Esquema Conceitual.....	2
	A. Descrição do Modelo Entidade-Relacionamento.....	2
	B. Diagrama do Esquema Conceitual.....	4
II.	Esquema Relacional.....	5
	A. Descrição do Modelo Relacional.....	5
	B. Diagrama do Esquema Relacional.....	7
III.	Criação do Banco de Dados.....	8
IV.	Especificação de Consultas em SQL.....	13
	A. Consultas.....	13
	B. Operações de Inserção.....	16
	C. Gatilhos e Procedimentos Armazenados.....	21

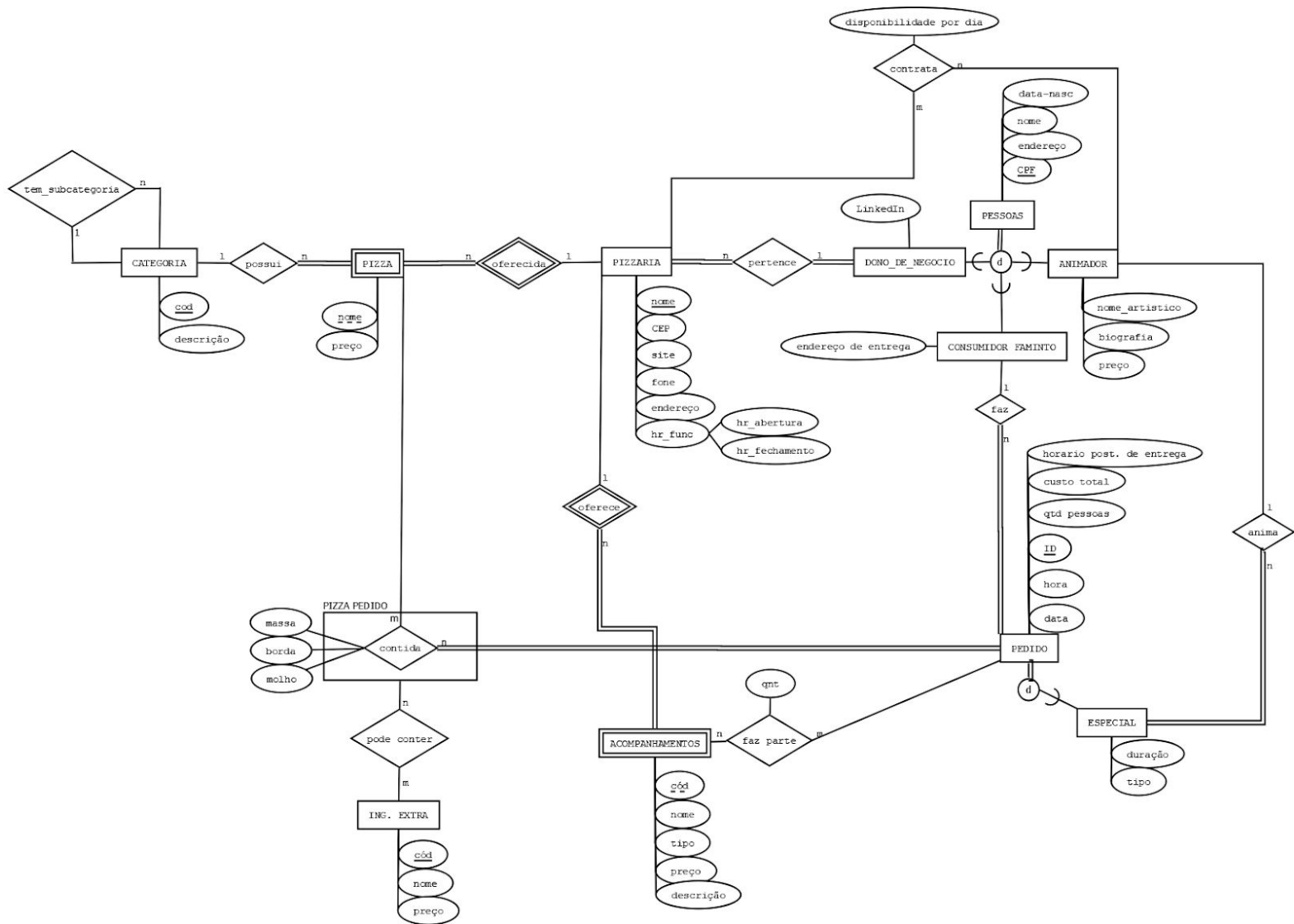
I. Esquema Conceitual

A. Descrição do Modelo Entidade-Relacionamento

- a) O item não possui descrições relevantes;
- b) Foi criada a entidade PESSOA com os atributos: data de nascimento, nome, endereço e CPF como sendo chave primária, para ser possível identificar cada instância;
- c) A partir da superclasse PESSOA foi criada a entidade DONO DE NEGÓCIO com atributo: LinkedIn e os atributos já existentes na superclasse. Tal hierarquia de especialização possui restrições de disjunção (d), pois uma entidade pode ser membro de no máximo uma das subclasses e de completude total, pois toda entidade na superclasse precisa ser membro de pelo menos uma subclasse;
- d) Foi criada uma entidade PIZZARIA e também um relacionamento entre ela e DONO DE NEGÓCIO, tendo este cardinalidade (n : 1) e participação total dos dois lados, pois a relação é obrigatória para ambos os lados, não existindo um sem o outro;
- e) Foram adicionados como atributos de PIZZARIA: CEP, endereço, número de telefone, website, horário de funcionamento sendo este tratado como composto (horário de abertura e horário de fechamento) e nome sendo sua chave primária.
- f) Foi criada a entidade PIZZA, sendo um entidade fraca, pois depende da existência de PIZZARIA e que possui os atributos: preço e nome, sendo este último tratado como chave fraca de PIZZA. Também foi criada um relacionamento entre PIZZA e PIZZARIA com cardinalidade (n : 1) e participação total da PIZZA;
- g) Foi criada uma entidade CATEGORIA tendo como atributos: código, como chave primária, e descrição composta por tipo (tradicional e especial) e sabor (doce ou salgado) e um relacionamento entre CATEGORIA e PIZZA, com cardinalidade (1 : n) e participação total da PIZZA, pois ela sempre tem que pertencer a uma CATEGORIA. Também foi criado o auto-relacionamento TEM SUBCATEGORIA na entidade CATEGORIA, que possui cardinalidade (1 : n);
- h) Foi criada a entidade fraca ACOMPANHAMENTOS, a qual possui atributos: nome, descrição, tipo de acompanhamento, preço, quantidade de acompanhamento e código, sendo este último sua chave fraca. Também foi criado um relacionamento entre PIZZARIA e ACOMPANHAMENTOS com cardinalidade (n : m) e participação total de ACOMPANHAMENTOS;
- i) A partir da superclasse PESSOA foi criada a entidade CONSUMIDOR FAMINTO que tem como atributo endereço de entrega e os já existentes na superclasse;

- j) Foi criada uma entidade PEDIDO com os seguintes atributos: ID, como sendo chave primária, data, horário do pedido, horário posterior de entrega e para quantas pessoas é o pedido;
- k) Foi criado um relacionamento entre PIZZA e PEDIDO, o qual possui cardinalidade ($n : m$) e os seguintes atributos: tipo da massa, borda e quantidade de molho. A participação da PIZZA no relacionamento é total, pois ela necessariamente está contida no PEDIDO;
- l) Foi criado um relacionamento entre ACOMPANHAMENTOS e PEDIDO cuja cardinalidade é ($n : m$) e possui o atributo quantidade;
- m) Foram criadas a entidade INGREDIENTES EXTRAS e a agregação PIZZA PEDIDO na relação CONTIDA entre PIZZA e PEDIDO. Também foi criada uma relação entre INGREDIENTES EXTRAS e a agregação recém criada, com cardinalidade ($n : m$);
- n) Foram adicionadas como atributos da entidade INGREDIENTES EXTRAS: código, como chave primária, nome e preço;
- o) Foi derivada a entidade PEDIDO em PEDIDO ESPECIAL, que possui duração e tipo de entretenimento como atributos. Tal hierarquia de especialização possui restrições de disjunção (d), pois uma entidade pode ser membro de no máximo uma das subclasses e de completude total, pois toda entidade na superclasse precisa ser membro de pelo menos uma subclasse;
- p) Foi adicionado à entidade PEDIDO o atributo custo total, pois ele pode ser calculado com base em informações existentes no banco de dados;
- q) Foi criada uma nova entidade derivada de PESSOA, chamada ANIMADOR o qual possui como atributos: nome artístico, biografia e preço;
- r) Foi criado um relacionamento entre ANIMADOR e PEDIDO ESPECIAL o qual possui cardinalidade ($1 : n$) e o PEDIDO ESPECIAL tem participação total no relacionamento, pois obrigatoriamente terá um ANIMADOR;
- s) Foi criado um relacionamento entre o ANIMADOR e a PIZZARIA o qual possui como atributo disponibilidade dele para trabalhar com cardinalidade ($n : m$), .

B. Diagrama do Esquema Conceitual



II. Esquema Relacional

A. Descrição do Modelo Relacional

Foram mapeadas as:

Entidades fortes

- CATEGORIA: Criamos uma tabela para CATEGORIA, que inclui código como sua chave primária e sua descrição.
- PIZZARIA: Criamos uma tabela para PIZZARIA, que inclui nome como sua chave primária, cep, site, fone, endereço, hora de abertura e hora de fechamento.
- PESSOA: Criamos uma tabela para PESSOA, que inclui CPF como sua chave primária, data de nascimento, nome e endereço.
- DONO_DE_NEGOCIO: Criamos uma tabela para DONO_DE_NEGOCIO, que é uma entidade derivada de PESSOAS, que inclui CPF como sua chave primária estrangeira (vinda de PESSOAS) e sua conta do linkedin.
- ANIMADOR: Criamos uma tabela para ANIMADOR, que é uma entidade derivada de PESSOA, que inclui CPF como sua chave primária estrangeira (vinda de PESSOA), seu nome artístico, biografia e preço.
- CONSUMIDOR FAMINTO: Criamos uma tabela para CONSUMIDOR FAMINTO, que é uma entidade derivada de PESSOA, que inclui CPF como sua chave primária estrangeira (vinda de PESSOA) e endereço de entrega.
- PEDIDO (NORMAL): Criamos uma tabela para PEDIDO (NORMAL) que inclui ID como sua chave primária.
- (PEDIDO) ESPECIAL: Criamos uma tabela para (PEDIDO) ESPECIAL, que é uma entidade derivada de PEDIDO, que inclui ID como sua chave primária estrangeira (vinda de PEDIDO), duração e tipo.
- ING. EXTRA: Criamos uma tabela para ING.EXTRA, que inclui COD como sua chave primária, nome e preço.

Agregação

- PIZZA_PEDIDO: Criamos uma tabela para PIZZA_PEDIDO, que é uma agregação baseada na opção de mapeamento 1 e, nesse caso, os atributos massa, borda e molho do relacionamento CONTIDA foram inseridos nesta tabela. Tendo como chave primária

estrangeira o ID (vindo de PEDIDO), o nome_pizza e o nome da pizzeria (vindos de PIZZA).

Entidades fracas

- PIZZA: Criamos uma tabela para PIZZA, que inclui nome_pizza como chave primária, nome (vindo de PIZZARIA) como chave primária estrangeira, cod (vindo de CATEGORIA), como chave estrangeira e preço.
- ACOMPANHAMENTO: Criamos uma tabela para ACOMPANHAMENTO, que inclui COD como chave primária, nome (vindo de PIZZARIA) como chave primária estrangeira, tipo, preço, descrição e nome.

Relacionamentos (1:N)

Decidimos usar o primeiro modelo de mapeamento para não termos que mapear os relacionamentos :

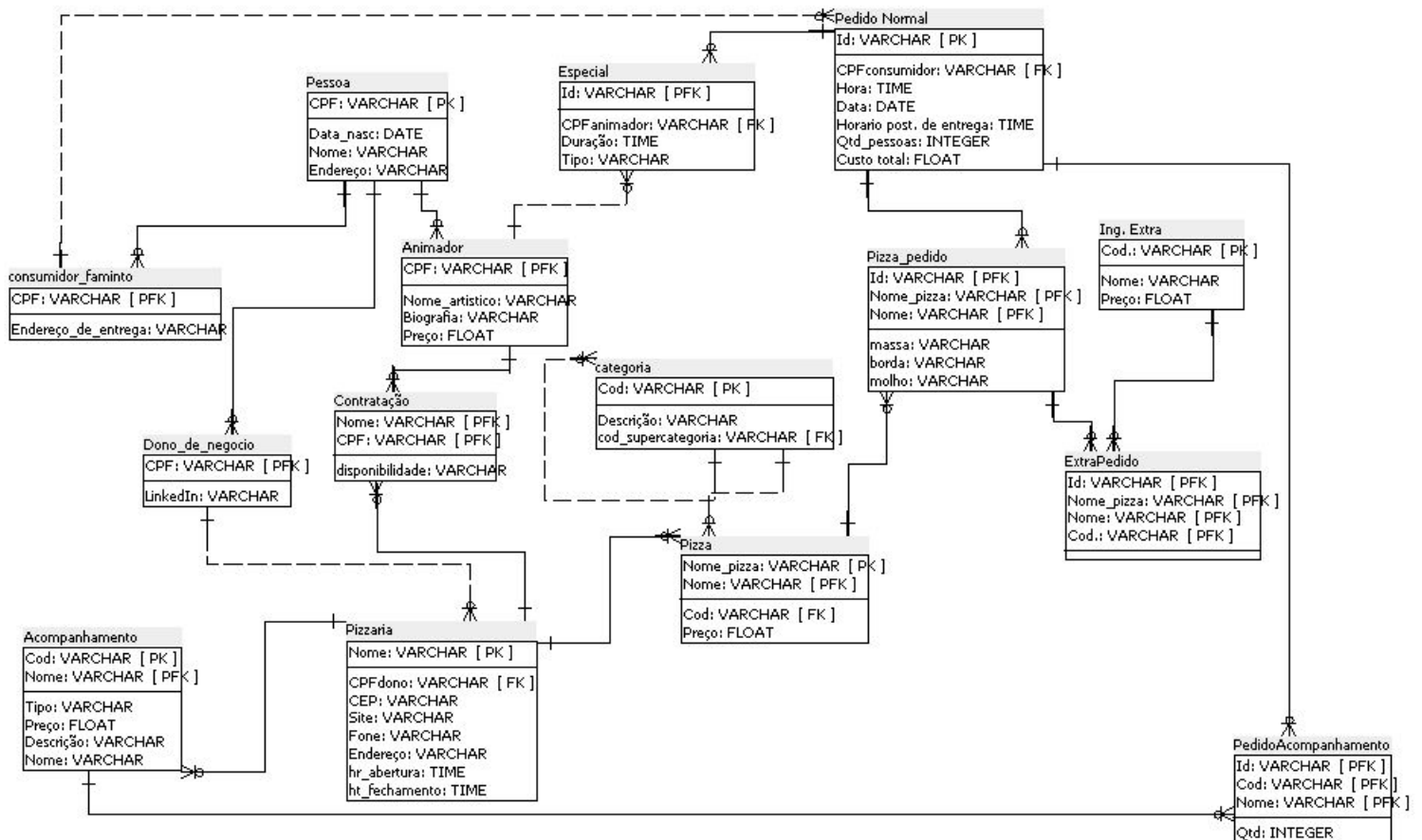
- TEM_SUBCATEGORIA: Inserimos o código da supercategoria na tabela CATEGORIA.
- POSSUI: Inserimos o código de CATEGORIA como chave estrangeira em PIZZA.
- ANIMA: Inserimos o CPF de ANIMADOR como chave estrangeira em (PEDIDO) ESPECIAL.
- PERTENCE: Inserimos o CPF do DONO_DE_NEGOCIO como chave estrangeira em PIZZARIA.

Relacionamentos (N:M)

- FORNECE: Criamos uma tabela FORNECIMENTO, que inclui CNPJ (vindo de FORNECEDOR) como chave primária estrangeira, nome (vindo de PIZZARIA) como chave primária estrangeira e data de fornecimento.
- CONTRATA: Criamos uma tabela CONTRATAÇÃO, que inclui nome (vindo de PIZZARIA) como chave primária estrangeira, CPF (vindo de ANIMADOR) como chave primária estrangeira e disponibilidade.
- FAZ PARTE: Criamos uma tabela PEDIDO_ACOMPANHAMENTO, que inclui ID (vindo de PEDIDO) como chave primária estrangeira, cod e nome (vindos de ACOMPANHAMENTO) como chave primária estrangeira.
- PODE CONTER: Criamos uma tabela EXTRA_PEDIDO, a qual representa o relacionamento PODE CONTER, que inclui ID, nome_pizza e nome (vindos de PIZZA_PEDIDO) como chave primária estrangeira e cod (vindo de ING. EXTRA) como chave primária estrangeira.

- CONTIDA: Estamos mapeando esse relacionamento junto com a agregação PIZZA_PEDIDO, ao criar a tabela PIZZA_PEDIDO que representa tudo isso.

B. Diagrama do Esquema Relacional



III. Criação do Banco de Dados

```
create schema pizzaria;  
set search_path to pizzaria;
```

```
create table Pessoa(  
    CPF varchar(15),  
    Data_Nasc date,  
    Nome varchar(50),  
    Endereco varchar(50),  
    Constraint PessoaPK primary key(CPF)  
);
```

```
create table Dono_De_Negocio(  
    CPF varchar(15),  
    LinkedIn varchar(30),  
    Constraint DonoPK primary key(CPF),  
    Constraint DonoFK foreign key (CPF)  
        References Pessoa  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
create table Consumidor_Faminto(  
    CPF varchar(15),  
    Endereco_Entrega varchar(50),  
    Constraint ConsumidorPK primary key (CPF),  
    Constraint ConsumidorFK foreign key (CPF)  
        References Pessoa(CPF)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```

create table Animador(
    CPF varchar(15),
    Nome_Artistico varchar(50),
    Biografia varchar(500),
    Preco float,
    Constraint AnimadorPK primary key (CPF),
    Constraint AnimadorFK foreign key (CPF)
        References Pessoa(CPF)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

create table Pedido_Normal(
    Ident varchar(10),
    CPF_Consumidor varchar(15),
    Hora time,
    Data_date,
    Hora_entrega time,
    Qtd_Pessoas smallint,
    Custo_Total float,
    Constraint Pedido_NormalPK primary key (ident),
    Constraint Pedido_NormalFK foreign key (CPF_Consumidor)
        References Consumidor_Faminto(CPF)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

create table Pedido_Especial(
    Ident varchar(15),
    CPF_Animador varchar(15),
    Duracao varchar(10),
    Tipo varchar(30),
    Constraint EspecialPK primary key (Ident),

    Constraint EspecialFK1 foreign key (Ident)
        REFERENCES PEDIDO_NORMAL(Ident)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    foreign key (CPF_Animador)
        References Animador(CPF)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```

create table Pizzaria(
    Nome varchar(20),
    CPF_Dono varchar(15),
    CEP varchar(15),
    Site varchar(50),
    Fone varchar(15),
    Endereco varchar(50),
    Hora_Abertura time,
    Hora_Fechamento time,
    Constraint PizzariaPK primary key (Nome),
    Constraint PizzariaFK foreign key (CPF_Dono)
        References Dono_De_Negocio(CPF)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```

create table Acompanhamento(
    Cod varchar(5),
    Nome_PIZZARIA varchar(20),
    NomeAcompanhamento varchar(20),
    Tipo varchar(10),
    Preco float,
    Descricao varchar(50),
    Constraint AcompanhaPK primary key (Cod, Nome_PIZZARIA),
    Constraint AcompanhaFK foreign key (Nome_PIZZARIA)
        References Pizzaria(Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```

create table Contratacao(
    Nome varchar(50),
    CPF varchar(15),
    Dispo varchar(15),
    Constraint ContrataPK primary key (Nome, CPF),
    Constraint ContrataFK1 foreign key (Nome)
        References Pizzaria (Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    foreign key (CPF)

```

```

        References Animador (CPF)
        ON UPDATE CASCADE
        ON DELETE CASCADE
    );

CREATE TABLE CATEGORIA(
    COD VARCHAR(20),
    DESCRICAO VARCHAR(50),
    COD_SUPER VARCHAR(20),
    CONSTRAINT CATEGPK PRIMARY KEY (COD),
    CONSTRAINT CATEGFK FOREIGN KEY (COD_SUPER)
        REFERENCES CATEGORIA (COD)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

create table PIZZA(
    Nome_pizza varchar(20),
    Nome varchar(20),
    COD varchar(20),
    preco float,
    CONSTRAINT PIZZAPK PRIMARY KEY (NOME_PIZZA,NOME),
    CONSTRAINT PIZZAFK FOREIGN KEY (NOME)
        REFERENCES PIZZARIA (NOME)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (COD)
        REFERENCES CATEGORIA (COD)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```

CREATE TABLE INGREDIENTE_EXTRA(
    COD VARCHAR(20),
    NOME VARCHAR(20),
    PRECO FLOAT,
    CONSTRAINT INGEXPK PRIMARY KEY (COD)
);

```

```

create table Pizza_Pedido(
    Ident varchar(15),
    Nome_Pizza varchar(20),

```

```

Nome varchar(20),
Massa varchar(20),
Borda varchar(20),
Molho varchar(20),
Constraint Pizzape_PK primary key (Ident, Nome_Pizza, Nome),
Constraint Pizzape_FK1 foreign key (Nome_Pizza, Nome)
    References Pizza(Nome_Pizza, Nome)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
foreign key (Ident)
    References Pedido_NORMAL(Ident)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE EXTRA_PEDIDO(
    IDENT VARCHAR(15),
    NOME_PIZZA VARCHAR(20),
    NOME VARCHAR(20),
    COD VARCHAR(20),
    CONSTRAINT EXTRAPEPK PRIMARY KEY (IDENT,NOME_PIZZA,NOME,COD),
    CONSTRAINT EXTRAPEFK1 FOREIGN KEY (IDENT,NOME_PIZZA,NOME)
        REFERENCES PIZZA_PEDIDO (IDENT,NOME_PIZZA,NOME)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (COD)
        REFERENCES INGREDIENTE_EXTRA(COD)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE PEDIDO_ACOMPANHAMENTO(
    IDENT VARCHAR(20),
    COD VARCHAR(20),
    NOME_ACOMP VARCHAR(20),
    NOME_PIZZARIA VARCHAR(20),
    QUANT INTEGER,
    CONSTRAINT PEACOMPPK PRIMARY KEY (IDENT,COD,NOME_PIZZARIA),
    CONSTRAINT PEACOMPFK1 FOREIGN KEY (COD,NOME_PIZZARIA)
        REFERENCES ACOMPANHAMENTO (COD,NOME_PIZZARIA)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Ident)

```

```

REFERENCES PEDIDO_NORMAL(Ident)
ON UPDATE CASCADE
ON DELETE CASCADE
);

```

IV. Especificação de Consultas em SQL

A. Consultas

Consulta 1 - O nome dos três usuários que mais gastam na plataforma: disponibiliza os nomes das pessoas que são mais “fiéis” ao APP, podendo ajudar o gerente a tomar alguma decisão de inserir vantagens para esses clientes fiéis do APP.

```

SELECT PESSOA.NOME
FROM PESSOA, (SELECT SUM(PEDIDO_NORMAL.CUSTO_TOTAL), PEDIDO_NORMAL.CPF_CONSUMIDOR
               FROM PIZZA INNER JOIN (PIZZA_PEDIDO INNER JOIN PEDIDO_NORMAL
ON(PEDIDO_NORMAL.IDENT = PIZZA_PEDIDO.IDENT))
               ON(PIZZA.NOME_PIZZA = PIZZA_PEDIDO.NOME_PIZZA)
               GROUP BY PEDIDO_NORMAL.CPF_CONSUMIDOR
               ORDER BY SUM(PRECO) DESC
               LIMIT 3) AS CONSULT
WHERE PESSOA.CPF = CONSULT.CPF_CONSUMIDOR

```

Consulta 2 - Média de preço das pizzas de cada pizzaria: ajudaria o gerente na tomada da decisão de seus preços, em relação à concorrência para que possa promover promoções ou um aumento no preço médio de suas pizzas.

```

SELECT COALESCE(PIZZARIA.NOME, 'Todas as pizzarias') AS Pizzarias,
       AVG(PIZZA.PRECO) AS Medias_Pizzarias
FROM PIZZA, PIZZARIA
WHERE PIZZA.NOME = PIZZARIA.NOME
GROUP BY
       ROLLUP (PIZZARIA.NOME)

```

Consulta 3 - Mostra qual a pizzaria mais lucrativa por dono: ajudaria o gerente a ver qual a pizzaria que está dando mais lucro podendo decidir em qual deverá investir mais.

```

SELECT DONO_DE_NEGOCIO.CPF,PIZZARIA.NOME,SUM(PEDIDO_NORMAL.CUSTO_TOTAL)
FROM DONO_DE_NEGOCIO INNER JOIN (PIZZARIA INNER JOIN (PIZZA_PEDIDO INNER JOIN PEDIDO_NORMAL
ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT)
ON PIZZARIA.NOME = PIZZA_PEDIDO.NOME)
ON PIZZARIA.CPF_DONO = DONO_DE_NEGOCIO.CPF
GROUP BY PIZZARIA.NOME, DONO_DE_NEGOCIO.CPF
ORDER BY SUM(PEDIDO_NORMAL.CUSTO_TOTAL) DESC;

```

Consulta 4 - Quanto cada pizzaria arrecadou no mês 06/2019: tem como intuito mostrar para o gerente de cada pizzaria qual foi o lucro total no mês de junho.

```
SELECT SUM(PEDIDO_NORMAL.CUSTO_TOTAL),PIZZA.NOME
FROM PIZZA INNER JOIN (PIZZA_PEDIDO INNER JOIN PEDIDO_NORMAL
                      ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT)
                      ON PIZZA_PEDIDO.NOME_PIZZA = PIZZA.NOME_PIZZA
WHERE PEDIDO_NORMAL.DATA_ >= '2019/07/01' AND
      PEDIDO_NORMAL.DATA_ <= '2019/07/31'
GROUP BY PIZZA.NOME
```

Consulta 5 - As três pizzas mais vendidas: dá um feedback ao gerente sobre quais as pizzas que, independente de tendências mensais, mais vendem, levando em conta todas as pizzas vendidas no banco de dados.

```
SELECT PIZZA.NOME_PIZZA,COUNT(PIZZA_PEDIDO.NOME_PIZZA)
FROM PEDIDO_NORMAL INNER JOIN (PIZZA_PEDIDO INNER JOIN PIZZA
                              ON PIZZA.NOME_PIZZA = PIZZA_PEDIDO.NOME_PIZZA
                              AND PIZZA.NOME = PIZZA_PEDIDO.NOME)
                              ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT
GROUP BY PIZZA.NOME_PIZZA
ORDER BY COUNT(PIZZA_PEDIDO.NOME_PIZZA) DESC
LIMIT 3
```

Consulta 6 - O animador que mais atendeu no mês atual: dá um feedback ao gerente sobre qual animador tentar disponibilizar por mais dias durante a semana, aumentar o preço ou até mesmo saber qual animador está mais apto a ser contratado.

```
SELECT NOME_ARTISTICO
FROM ANIMADOR INNER JOIN(PEDIDO_ESPECIAL INNER JOIN PEDIDO_NORMAL
                        ON PEDIDO_ESPECIAL.IDENT = PEDIDO_NORMAL.IDENT)
                        ON ANIMADOR.CPF = PEDIDO_ESPECIAL.CPF_ANIMADOR
WHERE PEDIDO_NORMAL.DATA_ >= '2019/07/01' AND PEDIDO_NORMAL.DATA_ < '2019/08/01'
GROUP BY NOME_ARTISTICO
HAVING COUNT(NOME_ARTISTICO) >= ALL(SELECT COUNT(NOME_ARTISTICO)
                                     FROM ANIMADOR INNER JOIN(PEDIDO_ESPECIAL INNER JOIN PEDIDO_NORMAL
                                                             ON PEDIDO_ESPECIAL.IDENT = PEDIDO_NORMAL.IDENT)
                                                             ON ANIMADOR.CPF = PEDIDO_ESPECIAL.CPF_ANIMADOR
                                     WHERE PEDIDO_NORMAL.DATA_ >= '2019/07/01' AND
                                           PEDIDO_NORMAL.DATA_ < '2019/08/01'
                                     GROUP BY NOME_ARTISTICO)
```

Consulta 7 - Quanto cada pizzaria vendeu de pizzas durante o mês atual (julho): dá um feedback ao gerente sobre quais pizzarias competidoras estão vendendo mais pizzas no mês, incentivando-o a procurar o por quê, ou, mostraria sua liderança nas vendas, incentivando-o a continuar sua estratégia de vendas.

```

SELECT COUNT(PIZZA_PEDIDO.IDENT) AS Quantidade_Pizzas,
       COALESCE(PIZZARIA.NOME, 'TODAS AS PIZZARIAS') AS PIZZARIAS
FROM PIZZARIA INNER JOIN(PIZZA_PEDIDO INNER JOIN PEDIDO_NORMAL
       ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT)
       ON PIZZARIA.NOME = PIZZA_PEDIDO.NOME
WHERE PEDIDO_NORMAL.DATA_ >= '2019/07/01' AND
       PEDIDO_NORMAL.DATA_ < '2019/08/01'
GROUP BY ROLLUP(PIZZARIA.NOME)

```

Consultas 8[a] e 8[b] - A consulta 8[a] traz como resultado o número de pessoas com idade entre 18 e 30 que pedem pedido especial. Enquanto a consulta 8[b] traz como resultado o número de pessoas com idade maior que 30: ambas as consultas ajudam o gerente a saber qual deve ser o público alvo de seus pedidos especiais.

8[a]:

```

SELECT COUNT(PESSOA.CPF) AS QUANTIDADE
FROM PESSOA INNER JOIN (PEDIDO_NORMAL INNER JOIN PEDIDO_ESPECIAL
       ON PEDIDO_NORMAL.IDENT = PEDIDO_ESPECIAL.IDENT)
       ON PESSOA.CPF = PEDIDO_NORMAL.CPF_CONSUMIDOR
WHERE PESSOA.DATA_NASC >= '1989/01/01' AND
       PESSOA.DATA_NASC < '2001/01/01'

```

8[b]:

```

SELECT COUNT(PESSOA.CPF) AS QUANTIDADE
FROM PESSOA INNER JOIN (PEDIDO_NORMAL INNER JOIN PEDIDO_ESPECIAL
       ON PEDIDO_NORMAL.IDENT = PEDIDO_ESPECIAL.IDENT)
       ON PESSOA.CPF = PEDIDO_NORMAL.CPF_CONSUMIDOR
WHERE PESSOA.DATA_NASC <= '1988/01/01'

```

Consulta 9 - Qual a pizza mais vendida no mês atual (julho): dá um feedback ao gerente sobre qual a pizza em tendência no mês, para que possa produzir mais, fazer promoções ou até mesmo aumentar o preço da pizza.

```

SELECT PIZZA.NOME, PIZZA.PRECO
FROM PEDIDO_NORMAL INNER JOIN (PIZZA_PEDIDO INNER JOIN PIZZA
       ON PIZZA.NOME_PIZZA = PIZZA_PEDIDO.NOME_PIZZA
       AND PIZZA.NOME = PIZZA_PEDIDO.NOME)
       ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT
WHERE CAST(PEDIDO_NORMAL.DATA_ AS VARCHAR) LIKE '2019-07-%'
GROUP BY PIZZA.NOME_PIZZA, PIZZA.NOME
HAVING COUNT(PIZZA_PEDIDO.NOME_PIZZA) >= ALL(SELECT COUNT(PIZZA.NOME_PIZZA)
       FROM PEDIDO_NORMAL INNER JOIN (PIZZA_PEDIDO INNER JOIN PIZZA
       ON PIZZA.NOME_PIZZA = PIZZA_PEDIDO.NOME_PIZZA
       AND PIZZA.NOME = PIZZA_PEDIDO.NOME)
       ON PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT
       WHERE CAST(PEDIDO_NORMAL.DATA_ AS VARCHAR) LIKE '2019-07-%'
       GROUP BY PIZZA.NOME_PIZZA)

```


Consulta 10 - O ingrediente extra mais vendido: dá um feedback ao gerente sobre qual ingrediente extra ele deva investir mais, aumentar o estoque ou até mesmo aumentar o preço.

```
SELECT INGREDIENTE_EXTRA.NOME, COUNT(INGREDIENTE_EXTRA.COD)
FROM INGREDIENTE_EXTRA INNER JOIN EXTRA_PEDIDO
      ON INGREDIENTE_EXTRA.COD = EXTRA_PEDIDO.COD
GROUP BY INGREDIENTE_EXTRA.NOME
HAVING COUNT(INGREDIENTE_EXTRA.COD) >= ALL(SELECT COUNT(INGREDIENTE_EXTRA.COD)
      FROM INGREDIENTE_EXTRA INNER JOIN EXTRA_PEDIDO
      ON INGREDIENTE_EXTRA.COD = EXTRA_PEDIDO.COD
      GROUP BY INGREDIENTE_EXTRA.NOME)
```

B. Operações de Inserção

```
INSERT INTO PESSOA VALUES ('66463343072','1950/07/25','LORENA ELIAS','RUA ITAGUACU, 1074');
INSERT INTO PESSOA VALUES ('30018096042','1970/03/03','YAGO DE CASTRO','RUA DAS AVENCAS, 442');
INSERT INTO PESSOA VALUES ('33004508028','1975/12/25','RICARDO ZAMBONI','RUA ACRE, 336');
INSERT INTO PESSOA VALUES ('89658221068','1965/10/11','RONALDO TEIXEIRA','RUA AMAZONA, 456');
INSERT INTO PESSOA VALUES ('82535972077','2000/01/01','JOAO NEVES','RUA JOAQUIM, 342');
INSERT INTO PESSOA VALUES ('39329543006','1998/06/27','LARA MENDES','AV JOAO NAVES, 4444');
INSERT INTO PESSOA VALUES ('17501150001','1998/11/16','VICTOR HUGO EUSTAQUIO','AV SEGISMUNDO, 1111');
INSERT INTO PESSOA VALUES ('40760174040','1989/08/06','BRENO CALDEIRA','AV DO CONTORNO, 126');
INSERT INTO PESSOA VALUES ('30565597078','1994/09/14','JULIA BIASI','RUA DOS COLIBRIS, 555');
INSERT INTO PESSOA VALUES ('76419280052','2001/09/11','ALEX GARCIA','RUA CERQUEIRA CESAR, 222');
INSERT INTO PESSOA VALUES ('80250873010','1980/06/24','BRUNO OLIVEIRA','AV CESAR FINOTTI, 550');
INSERT INTO PESSOA VALUES ('49885481044','1990/04/14','LUCAS SANTOS PORTUGAL','RUA TIRADENTES, 789');
INSERT INTO PESSOA VALUES ('38439357001','1990/08/11','LUIZ OTAVIO','RUA DAS ACACIAS, 666');
INSERT INTO PESSOA VALUES ('64732139027','1994/04/06','ALEX SILVA','RUA DA BALBURDIA, 624');
INSERT INTO PESSOA VALUES ('33442922003','1971/01/01','PEDRO MARMELO SILVA','RUA DA BALBURDIA, 532');
INSERT INTO PESSOA VALUES ('50949237060','1972/02/02','PEDRO BARBOSA','RUA DOS BOBOS, 0');
INSERT INTO PESSOA VALUES ('14511417008','1974/04/04','PEDRO MARMOTA','RUA DOS ROEDORES, 2630');
INSERT INTO PESSOA VALUES ('06459207038','1976/06/06','JOAO VITOR BARBOSA','RUA DAS BRUMAS, 2511');
INSERT INTO PESSOA VALUES ('41918144001','1978/08/08','JOAO VICTOR BARMÍ','RUA DA BALBURDIA, 6327');
INSERT INTO PESSOA VALUES ('40116068043','1980/10/10','LUCAS MENDES','RUA DAS ACACIAS, 999');
INSERT INTO PESSOA VALUES ('92157107010','1982/12/12','LUCAS MOMENTI','RUA DAS TULIPAS, 756');
INSERT INTO PESSOA VALUES ('00121707040','1973/03/03','LUCAS GABRIEL ANDRADE','RUA DA CONCORDIA, 682');
INSERT INTO PESSOA VALUES ('09486727066','1975/05/05','FABIO ANDRADE','RUA DA CONCORDIA, 682');
INSERT INTO PESSOA VALUES ('19354861016','1977/07/07','MATHEUS ARDUINO','AV THAILANDIA, 3574');
INSERT INTO PESSOA VALUES ('78165617095','1979/09/09','MATHEUS ALCANTRA','AV BRASIL, 456');
INSERT INTO PESSOA VALUES ('46110749001','1981/11/11','TARCISIO LIMA','RUA DAS TULIPAS, 251');
```

INSERT INTO PESSOA VALUES ('89938087086','1991/01/11','JORGE LUIZ','RUA DAS ORQUIDIAS, 444');
 INSERT INTO PESSOA VALUES ('71579708056','1993/03/03','LUIZ JORGE','RUA DAS ORQUIDIAS, 444');
 INSERT INTO PESSOA VALUES ('12126086070','1995/05/05','YGOR CASTRO','RUA CARAMELO, 123');
 INSERT INTO PESSOA VALUES ('77865030053','1997/07/07','MIGUEL CASPIM','RUA COLISEU, 1253');

INSERT INTO DONO_DE_NEGOCIO VALUES ('89658221068','ronaldo-teixeira-135326753');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('66463343072','lorena-elias-032339165');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('33004508028','ricardo-zamboni-123431245');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('30018096042','yago-de-castro-135357438');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('33442922003','pedro-marmelo-silva-354862579');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('50949237060','pedro-barbosa-369854721');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('14511417008','pedro-marmota-657321489');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('06459207038','joao-vitor-barbosa-842679310');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('41918144001','joao-victor-barmi-542687931');
 INSERT INTO DONO_DE_NEGOCIO VALUES ('40116068043','lucas-mendes-014785692');

INSERT INTO CONSUMIDOR_FAMINTO VALUES ('82535972077','RUA DAS ARAUCARIAS, 123');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('39329543006','AV JOAO NAVES, 555');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('40760174040','RUA DAS ACACIAS, 1432');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('30565597078','RUA DA BANDEIRA 3121');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('38439357001','RUA DAS ACACIAS, 666');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('64732139027','RUA DA BALBURDIA, 624');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('92157107010','RUA DAS TULIPAS, 756');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('00121707040','RUA DA CONCORDIA, 682');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('09486727066','RUA DA CONCORDIA, 682');
 INSERT INTO CONSUMIDOR_FAMINTO VALUES ('19354861016','AV THAILANDIA, 3574');

INSERT INTO ANIMADOR VALUES ('17501150001','PRINCE','Faço cover do cantor Prince', 100);
 INSERT INTO ANIMADOR VALUES ('76419280052','PALHACITOS','Sou engraçado haha', 70);
 INSERT INTO ANIMADOR VALUES ('80250873010','MAICOL JEQUISON','Faço cover do cantor Michael Jackson', 150);
 INSERT INTO ANIMADOR VALUES ('49885481044','MALABARISTA CALIENTE','Faço malabarismo com pinos de boliche em chamas', 200);
 INSERT INTO ANIMADOR VALUES ('78165617095','ELVIS PRESLEY','Faco cover do artistica Elvis Presley', 90);
 INSERT INTO ANIMADOR VALUES ('46110749001','POPEYE','Sou um fisiculturista que imita o personagem Popeye', 95);
 INSERT INTO ANIMADOR VALUES ('89938087086','MALABAR FAQUITO','Faco malabarismo com facas e objetos pontiagudos', 125);
 INSERT INTO ANIMADOR VALUES ('71579708056','PICASSO GENERICO','Faco uma pintura dos clientes', 50);
 INSERT INTO ANIMADOR VALUES ('12126086070','NEYMAR DO FREESTYLE','Faco truques de freestyle com uma bola de futebol', 60);
 INSERT INTO ANIMADOR VALUES ('77865030053','MARCOS PIADISTA','Faco um show de standup para os clientes', 225);

INSERT INTO PIZZARIA VALUES ('PIZZAIOLO','89658221068','38408-168','WWW.PIZZAIOLO.COM.BR','0800 940 0940','Av.Belarmino Cotta Pacheco, 645','18:00','00:00');

```

INSERT INTO PIZZARIA VALUES('DISK PIZZA','66463343072','38408-140','WWW.DISKPIZZA.COM.BR','
3231-0770','R. José Paes de Almeida, 234','17:00','23:30');
INSERT INTO PIZZARIA VALUES('PIZZA DA
VOVO','33004508028','38400-426','WWW.PIZZADAVOVO.COM.BR','3210-7079','Av. Dr. Laérte Viêira Gonçalves','
18:00','00:00');
INSERT INTO PIZZARIA VALUES('NONA','30018096042','38408-176','WWW.PIZZANONA.COM.BR','3214-9150','
Av. Dr. Laérte Viêira Gonçalves, 1346','17:00','23:30');
INSERT INTO PIZZARIA VALUES('FOME
ZERO','33442922003','30550-600','WWW.FOMEZEROPIZZAS.COM.BR','3211-9110','Av. dos Reis, 132','
18:00','00:00');
INSERT INTO PIZZARIA
VALUES('PIZZANDO','50949237060','33065-430','WWW.PIZZANDO.COM.BR','3312-4312','R. do Almirante, 1234','
18:30','00:00');
INSERT INTO PIZZARIA VALUES('UAI
PIZZAS','30018096042','35553-043','WWW.UAIPIZZAS.COM.BR','3653-2352','R. São José, 1035','18:30','23:30');
INSERT INTO PIZZARIA VALUES('LIGA LÁ','06459207038','33054-64','WWW.LIGALAPIZZA.COM.BR','3387-6534','
Av. Rio Pardo, 1362','18:00','01:00');
INSERT INTO PIZZARIA VALUES('BORDA DA
CIDADE','41918144001','37048-047','WWW.BORDADACIDADE.COM.BR','3446-3212','R. Acre, 146','
19:00','01:00');
INSERT INTO PIZZARIA VALUES('MA CHE
PIZZA','40116068043','33440-525','WWW.MACHEPIZZA.COM.BR','3313-5069','Av. Dom Pedro, 1400','
18:00','23:30');

```

```

INSERT INTO ACOMPANHAMENTO VALUES('01','PIZZAIOLO','MAIONESE','CONDIMENTO',0.50,'UMA UNIDADE
DE SACHE DE MAIONESE');
INSERT INTO ACOMPANHAMENTO VALUES('01','DISK PIZZA','MAIONESE','CONDIMENTO',0.75,'UMA UNIDADE
DE SACHE DE MAIONESE');
INSERT INTO ACOMPANHAMENTO VALUES('01','PIZZA DA VOVO','MAIONESE','CONDIMENTO',0.20,'UMA
UNIDADE DE SACHE DE MAIONESE');
INSERT INTO ACOMPANHAMENTO VALUES('01','NONA','MAIONESE','CONDIMENTO',1.00,'UMA UNIDADE DE
SACHE DE MAIONESE');
INSERT INTO ACOMPANHAMENTO VALUES('02','PIZZAIOLO','COCA-COLA','BEBIDA',8.00,'UMA UNIDADE DE
COCA-COLA DE 2 LITROS');
INSERT INTO ACOMPANHAMENTO VALUES('02','DISK PIZZA','COCA-COLA','BEBIDA',5.00,'UMA UNIDADE DE
COCA-COLA DE 2 LITROS');
INSERT INTO ACOMPANHAMENTO VALUES('02','PIZZA DA VOVO','COCA-COLA','BEBIDA',10.00,'UMA
UNIDADE DE COCA-COLA DE 2 LITROS');
INSERT INTO ACOMPANHAMENTO VALUES('02','NONA','COCA-COLA','BEBIDA',7.00,'UMA UNIDADE DE
COCA-COLA DE 2 LITROS');
INSERT INTO ACOMPANHAMENTO VALUES('03','PIZZAIOLO','KETCHUP','CONDIMENTO',0.50,'UMA UNIDADE
DE SACHE DE KETCHUP');
INSERT INTO ACOMPANHAMENTO VALUES('03','DISK PIZZA','KETCHUP','CONDIMENTO',0.75,'UMA UNIDADE
DE SACHE DE KETCHUP');
INSERT INTO ACOMPANHAMENTO VALUES('03','PIZZA DA VOVO','KETCHUP','CONDIMENTO',0.20,'UMA
UNIDADE DE SACHE DE KETCHUP');
INSERT INTO ACOMPANHAMENTO VALUES('03','NONA','KETCHUP','CONDIMENTO',1.00,'UMA UNIDADE DE
SACHE DE KETCHUP');

```

```

INSERT INTO ACOMPANHAMENTO VALUES('04','PIZZAIOLO','BRAHMA','BEBIDA',3.00,'UMA UNIDADE DE
BRAHMA 350ml');
INSERT INTO ACOMPANHAMENTO VALUES('04','DISK PIZZA','SKOL','BEBIDA',2.50,'UMA UNIDADE DE SKOL
350ml');
INSERT INTO ACOMPANHAMENTO VALUES('04','PIZZA DA VOVO','HEINEKEN','BEBIDA',4.00,'UMA UNIDADE
DE HEINEKEN 350ml');
INSERT INTO ACOMPANHAMENTO VALUES('04','NONA','BUDWEISER','BEBIDA',5.00,'UMA UNIDADE DE
BUDWEISER 350ml');

```

```

INSERT INTO CONTRATAÇÃO VALUES ('PIZZAIOLO','17501150001','SEG/TER/QUA');
INSERT INTO CONTRATAÇÃO VALUES ('PIZZAIOLO','76419280052','QUI/SEX/SAB');
INSERT INTO CONTRATAÇÃO VALUES ('DISK PIZZA','76419280052','SEG/TER/QUA');
INSERT INTO CONTRATAÇÃO VALUES ('DISK PIZZA','17501150001','QUI/SEX/SAB');
INSERT INTO CONTRATAÇÃO VALUES ('PIZZA DA VOVO','80250873010','SEG/TER/QUA');
INSERT INTO CONTRATAÇÃO VALUES ('PIZZA DA VOVO','49885481044','QUI/SEX/DOM');
INSERT INTO CONTRATAÇÃO VALUES ('NONA','80250873010','QUI/SEX/SAB');
INSERT INTO CONTRATAÇÃO VALUES ('NONA','49885481044','SEG/TER/QUA');
INSERT INTO CONTRATAÇÃO VALUES ('PIZZAIOLO','12126086070','SEG/TER/QUA');
INSERT INTO CONTRATAÇÃO VALUES ('DISK PIZZA','12126086070','QUI/SEX/SAB');

```

```

INSERT INTO CATEGORIA VALUES ('1','PIZZA SALGADA',NULL);
INSERT INTO CATEGORIA VALUES ('2','PIZZA DOCE',NULL);
INSERT INTO CATEGORIA VALUES ('3','TRADICIONAL','1');
INSERT INTO CATEGORIA VALUES ('4','ESPECIAL','1');
INSERT INTO CATEGORIA VALUES ('5','NOVAS','1');
INSERT INTO CATEGORIA VALUES ('6','NOVAS','2');
INSERT INTO CATEGORIA VALUES ('7','PROMOÇÃO','3');
INSERT INTO CATEGORIA VALUES ('8','PROMOÇÃO','4');
INSERT INTO CATEGORIA VALUES ('9','VEGETARIANA','1');
INSERT INTO CATEGORIA VALUES ('10','VEGANA','2');

```

```

INSERT INTO PIZZA VALUES ('CALABRESA','NONA','3',19.99);
INSERT INTO PIZZA VALUES ('MODA DA CASA','PIZZA DA VOVO','3',24.90);
INSERT INTO PIZZA VALUES ('PEPPERONI','PIZZAIOLO','4',49.99);
INSERT INTO PIZZA VALUES ('PEPPERONI','DISK PIZZA','4',49.99);
INSERT INTO PIZZA VALUES ('CACHORRO QUENTE','NONA','4',26.50);
INSERT INTO PIZZA VALUES ('QUATRO QUEIJOS','NONA','9',24.60);
INSERT INTO PIZZA VALUES ('QUATRO QUEIJOS','PIZZA DA VOVO','9',26.60);
INSERT INTO PIZZA VALUES ('QUATRO QUEIJOS','PIZZAIOLO','9',49.99);
INSERT INTO PIZZA VALUES ('BROCOLIS','NONA','10',25.60);
INSERT INTO PIZZA VALUES ('BROCOLIS','PIZZA DA VOVO','10',26.60);
INSERT INTO PIZZA VALUES ('BROCOLIS','DISK PIZZA','10',49.99);
INSERT INTO PIZZA VALUES ('SORVETE','NONA','2',25.90);
INSERT INTO PIZZA VALUES ('SORVETE','DISK PIZZA','2',59.99);
INSERT INTO PIZZA VALUES ('CASADINHO','PIZZA DA VOVO','2',26.99);
INSERT INTO PIZZA VALUES ('BEIJINHO','PIZZAIOLO','2',59.90);
INSERT INTO PIZZA VALUES ('BEIJINHO','NONA','2',25.90);

```

```

INSERT INTO INGREDIENTE_EXTRA VALUES('0001','CATUPIRY',3.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('0010','TOMATE SECO',2.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('0011','COGUMELOS CHAMPIGNON',4.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('0100','BATATA PALHA',1.50);
INSERT INTO INGREDIENTE_EXTRA VALUES('0101','BACON',2.55);
INSERT INTO INGREDIENTE_EXTRA VALUES('0110','CEBOLA',1.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('0111','CEBOLA ROXA',1.35);
INSERT INTO INGREDIENTE_EXTRA VALUES('1000','PALMITO',2.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('1001','MILHO',1.00);
INSERT INTO INGREDIENTE_EXTRA VALUES('1010','ERVILHA',1.00);

```

```

INSERT INTO PEDIDO_NORMAL VALUES ('111','82535972077','19:50','2019/07/01','20:30',3,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('112','39329543006','19:30','2019/07/01','20:00',4,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('113','40760174040','19:10','2019/07/01','20:00',2,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('114','30565597078','17:00','2019/07/01','17:30',3,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('115','38439357001','19:00','2019/07/01','19:30',5,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('116','64732139027','21:00','2019/07/02','21:30',1,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('117','92157107010','21:05','2019/07/02','21:30',2,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('118','00121707040','21:40','2019/07/02','22:30',3,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('119','00121707040','21:55','2019/07/02','22:30',3,NULL);
INSERT INTO PEDIDO_NORMAL VALUES ('120','09486727066','22:12','2019/07/02','23:00',2,NULL);

```

```

INSERT INTO PEDIDO_ESPECIAL VALUES ('111','49885481044','30','MALABARISMO');
INSERT INTO PEDIDO_ESPECIAL VALUES ('112','80250873010','60','MUSICAL');
INSERT INTO PEDIDO_ESPECIAL VALUES ('113','17501150001','30','MUSICAL');
INSERT INTO PEDIDO_ESPECIAL VALUES ('114','49885481044','90','MALABARISMO');
INSERT INTO PEDIDO_ESPECIAL VALUES ('115','76419280052','30','PALHAÇO');
INSERT INTO PEDIDO_ESPECIAL VALUES ('116','46110749001','30','MÍMICA');
INSERT INTO PEDIDO_ESPECIAL VALUES ('117','71579708056','120','PINTURA');
INSERT INTO PEDIDO_ESPECIAL VALUES ('118','12126086070','30','MÁGICA');
INSERT INTO PEDIDO_ESPECIAL VALUES ('119','77865030053','60','STANDUP');
INSERT INTO PEDIDO_ESPECIAL VALUES ('120','89938087086','30','MALABARISMO');

```

```

INSERT INTO PIZZA_PEDIDO VALUES ('111','CACHORRO QUENTE','NONA','MEDIA','NORMAL','NORMAL');
INSERT INTO PIZZA_PEDIDO VALUES ('112','CACHORRO QUENTE','NONA','FINA','NORMAL','NORMAL');
INSERT INTO PIZZA_PEDIDO VALUES ('113','MODA DA CASA','PIZZA DA VOVO','GROSSA','NORMAL','EXTRA');
INSERT INTO PIZZA_PEDIDO VALUES ('114','MODA DA CASA','PIZZA DA VOVO','MEDIA','NORMAL','POUCO');
INSERT INTO PIZZA_PEDIDO VALUES ('115','PEPPERONI','PIZZAIOLO','GROSSA','RECHEADA C CATUPIRY','POUCO');
INSERT INTO PIZZA_PEDIDO VALUES ('116','MODA DA CASA','PIZZA DA VOVO','MEDIA','NORMAL','EXTRA');
INSERT INTO PIZZA_PEDIDO VALUES ('117','PEPPERONI','DISK PIZZA','GROSSA','RECHEADA C CATUPIRY','NORMAL');
INSERT INTO PIZZA_PEDIDO VALUES ('118','BEIJINHO','PIZZAIOLO','FINA','NORMAL','NORMAL');
INSERT INTO PIZZA_PEDIDO VALUES ('119','BROCOLIS','DISK PIZZA','NORMAL','NORMAL','EXTRA');
INSERT INTO PIZZA_PEDIDO VALUES ('120','SORVETE','NONA','NORMAL','NORMAL','NORMAL');
INSERT INTO PIZZA_PEDIDO VALUES ('111','QUATRO QUEIJOS','NONA','GROSSA','NORMAL','EXTRA');

```

```

INSERT INTO EXTRA_PEDIDO VALUES('111','CACHORRO QUENTE','NONA','0001');
INSERT INTO EXTRA_PEDIDO VALUES('112','CACHORRO QUENTE','NONA','0010');
INSERT INTO EXTRA_PEDIDO VALUES('113','MODA DA CASA','PIZZA DA VOVO','0010');
INSERT INTO EXTRA_PEDIDO VALUES('114','MODA DA CASA','PIZZA DA VOVO','0011');
INSERT INTO EXTRA_PEDIDO VALUES('115','PEPPERONI','PIZZAIOLO','0100');
INSERT INTO EXTRA_PEDIDO VALUES('116','MODA DA CASA','PIZZA DA VOVO','0110');
INSERT INTO EXTRA_PEDIDO VALUES('117','PEPPERONI','DISK PIZZA','0111');
INSERT INTO EXTRA_PEDIDO VALUES('118','BEIJINHO','PIZZAIOLO','0100');
INSERT INTO EXTRA_PEDIDO VALUES('119','BROCOLIS','DISK PIZZA','1000');
INSERT INTO EXTRA_PEDIDO VALUES('120','SORVETE','NONA','0100');

```

```

INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('111','01','MAIONESE','NONA',4);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('112','02','COCA-COLA','NONA',1);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('113','02','COCA-COLA','PIZZA DA VOVO',2);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('114','01','MAIONESE','PIZZA DA VOVO',8);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('115','03','KETCHUP','PIZZAIOLO',10);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('116','04','HEINEKEN','PIZZA DA VOVO',6);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('117','04','SKOL','DISK PIZZA',12);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('118','02','COCA-COLA','PIZZAIOLO',2);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('119','02','COCA-COLA','DISK PIZZA',1);
INSERT INTO PEDIDO_ACOMPANHAMENTO VALUES ('120','02','COCA-COLA','NONA',5);

```

C. Gatilhos e Procedimentos Armazenados

Gatilho Especificado

Identificamos que o gatilho especificado era um gatilho para a atualização do valor do campo `custo_total` da tabela `pedido_normal`.

Gatilho - para essa implementação, nós utilizamos vários gatilhos em tabelas que ao serem povoadas, causam alguma alteração no custo total de um pedido.

```

CREATE TRIGGER CUSTOTOTAL1
AFTER INSERT OR UPDATE OR DELETE ON pizza_pedido
FOR EACH ROW
EXECUTE PROCEDURE CUSTOTOTAL();

```

```

CREATE TRIGGER CUSTOTOTAL2
AFTER INSERT OR UPDATE OR DELETE ON pedido_acompanhamento
FOR EACH ROW
EXECUTE PROCEDURE CUSTOTOTAL();

```

```

CREATE TRIGGER CUSTOTOTAL3
AFTER INSERT OR UPDATE OR DELETE ON extra_pedido
FOR EACH ROW
EXECUTE PROCEDURE CUSTOTOTAL();

```

```

CREATE TRIGGER CUSTOTOTAL4
AFTER INSERT OR UPDATE OR DELETE ON pedido_especial
FOR EACH ROW
EXECUTE PROCEDURE CUSTOTOTAL();

```

Procedimento armazenado usado pelos gatilhos - Nessa implementação, nós calculamos cada parte que pode alterar o valor do custo_total, somamos todas essas partes e no fim atribuímos essa soma total ao custo_total do pedido que foi modificado.

```

CREATE OR REPLACE FUNCTION CUSTOTOTAL()
RETURNS TRIGGER AS $CUSTOTOTAL$
DECLARE
    PIZZATOTAL FLOAT;
    ACOMPANHAMENTOTOTAL FLOAT;
    TOTAL FLOAT;
    CUSTOANIMACAO FLOAT;
    DURACAOANIMACAO INTEGER;
    QUANTIDADEACOMPANHAMENTO INTEGER;
    CUSTOACOMPANHAMENTO FLOAT;
    PRECO_ING_EXTRA FLOAT;
    TOTALPEDIDOESP INTEGER;
    PRECO_TOTAL_ING_EXTRA FLOAT;

BEGIN
    PIZZATOTAL:= 0;
    ACOMPANHAMENTOTOTAL := 0;
    CUSTOANIMACAO := 0;
    QUANTIDADEACOMPANHAMENTO := 0;
    CUSTOACOMPANHAMENTO := 0;
    TOTAL := 0;
    PRECO_ING_EXTRA := 0;
    TOTALPEDIDOESP := 0;
    PRECO_TOTAL_ING_EXTRA := 0;

    --TESTAR SE PEDIDO COM ENTRETENIMENTO
    SELECT COUNT(PEDIDO_ESPECIAL.IDENT) INTO TOTALPEDIDOESP
    FROM PEDIDO_ESPECIAL
    WHERE PEDIDO_ESPECIAL.IDENT = NEW.IDENT;

    IF (TOTALPEDIDOESP = 1)
    --CALCULA O PRECO DA ANIMACAO CASO SEJA PEDIDO ESPECIAL
    THEN
        SELECT (ANIMADOR.PRECO * ((CAST(PEDIDO_ESPECIAL.DURACAO AS INTEGER))/30))
        INTO CUSTOANIMACAO
        FROM ANIMADOR INNER JOIN PEDIDO_ESPECIAL
        ON (PEDIDO_ESPECIAL.CPF_ANIMADOR = ANIMADOR.CPF)
        WHERE PEDIDO_ESPECIAL.IDENT = NEW.IDENT;
    END IF;

```

```

--CALCULA O PRECO TOTAL DOS ACOMPANHAMENTOS
SELECT SUM(PEDIDO_ACOMPANHAMENTO.QUANT * ACOMPANHAMENTO.PRECO)
      INTO ACOMPANHAMENTOTOTAL
FROM ACOMPANHAMENTO, PEDIDO_ACOMPANHAMENTO, PEDIDO_NORMAL
WHERE ACOMPANHAMENTO.COD = PEDIDO_ACOMPANHAMENTO.COD AND
      PEDIDO_NORMAL.IDENT = PEDIDO_ACOMPANHAMENTO.IDENT AND
      ACOMPANHAMENTO.NOME_PIZZARIA = PEDIDO_ACOMPANHAMENTO.NOME_PIZZARIA AND
      PEDIDO_ACOMPANHAMENTO.IDENT = NEW.IDENT
GROUP BY PEDIDO_ACOMPANHAMENTO.IDENT;

--CALCULA O PREÇO TOTAL DAS PIZZAS
SELECT SUM(PIZZA.PRECO) INTO PIZZATOTAL
FROM PIZZA INNER JOIN (PIZZA_PEDIDO INNER JOIN PEDIDO_NORMAL
                      ON (PIZZA_PEDIDO.IDENT = PEDIDO_NORMAL.IDENT))
                      ON PIZZA_PEDIDO.NOME_PIZZA = PIZZA.NOME_PIZZA
WHERE PIZZA_PEDIDO.IDENT = NEW.IDENT;

--CALCULA O PREÇO TOTAL DOS EXTRAS
SELECT SUM(INGREDIENTE_EXTRA.PRECO) INTO PRECO_TOTAL_ING_EXTRA
FROM INGREDIENTE_EXTRA INNER JOIN (EXTRA_PEDIDO INNER JOIN PIZZA_PEDIDO
                                ON EXTRA_PEDIDO.IDENT = PIZZA_PEDIDO.IDENT)
                                ON INGREDIENTE_EXTRA.COD = EXTRA_PEDIDO.COD
WHERE PIZZA_PEDIDO.IDENT = NEW.IDENT;

IF(CUSTOANIMACAO = NULL) THEN
    CUSTOANIMACAO :=0;
END IF;

IF(ACOMPANHAMENTOTOTAL = NULL) THEN
    ACOMPANHAMENTOTOTAL := 0;
END IF;

IF(PRECO_TOTAL_ING_EXTRA = NULL) THEN
    PRECO_TOTAL_ING_EXTRA := 0;
END IF;

IF(PIZZATOTAL = NULL) THEN
    PIZZATOTAL := 0;
END IF;

TOTAL = CUSTOANIMACAO + ACOMPANHAMENTOTOTAL + PRECO_TOTAL_ING_EXTRA + PIZZATOTAL;
RAISE NOTICE 'CUSTOS COMPUTADOS PARA PEDIDO %: ANIMACAO = %, ACOMPANHAMENTO = %,
EXTRA = %, PIZZA = %', NEW.IDENT, CUSTOANIMACAO, ACOMPANHAMENTOTOTAL,
PRECO_TOTAL_ING_EXTRA, PIZZATOTAL;

UPDATE PEDIDO_NORMAL SET CUSTO_TOTAL = TOTAL WHERE PEDIDO_NORMAL.IDENT = NEW.IDENT;

```



```
RETURN NEW;  
END;  
$CUSTOTOTAL$ language 'plpgsql';
```

```
CREATE TRIGGER CUSTOTOTAL1  
AFTER INSERT OR UPDATE OR DELETE ON pizza_pedido  
FOR EACH ROW  
EXECUTE PROCEDURE CUSTOTOTAL();
```

```
CREATE TRIGGER CUSTOTOTAL2  
AFTER INSERT OR UPDATE OR DELETE ON pedido_acompanhamento  
FOR EACH ROW  
EXECUTE PROCEDURE CUSTOTOTAL();
```

```
CREATE TRIGGER CUSTOTOTAL3  
AFTER INSERT OR UPDATE OR DELETE ON extra_pedido  
FOR EACH ROW  
EXECUTE PROCEDURE CUSTOTOTAL();
```

```
CREATE TRIGGER CUSTOTOTAL4  
AFTER INSERT OR UPDATE OR DELETE ON pedido_especial  
FOR EACH ROW  
EXECUTE PROCEDURE CUSTOTOTAL();
```

Gatilho Definido pelo Grupo

Escolhemos implementar um gatilho para validação de CPF dos usuários. Ajuda o gerente a ter mais segurança sobre os clientes do APP.

Gatilho - Ao inserir ou atualizar o campo do CPF, aciona-se o gatilho para determinar se aquele CPF é válido e pode ser inserido normalmente, ou se ele não é, lançando uma exceção.

```
CREATE TRIGGER CPFVALIDO  
BEFORE INSERT OR UPDATE OF CPF ON PESSOA  
FOR EACH ROW  
EXECUTE PROCEDURE validarCPF();
```

Procedimento armazenado usado pelo gatilho - é baseado no algoritmo de verificação de CPF. Calcula-se os dois últimos dígitos para decidir se ele é válido ou não.

```
CREATE OR REPLACE FUNCTION validarCPF() RETURNS trigger AS $$  
DECLARE  
    cpfdado varchar(11);  
    temp1 real;  
    temp2 real;  
    soma integer;  
    digito1 integer;
```

```

    digito2 integer;
    tam integer;
    index integer;
    cpftemp varchar(11);

BEGIN
    cpfdado := new.CPF;

    IF char_length(cpfdado) != 11
    THEN
        RAISE NOTICE 'Formato inválido: %', cpfdado;
    END IF;

    temp1 := 0;
    soma := 0;
    digito1 := 0;
    digito2 := 0;
    index := 0;
    val_par_cpf := cpfdado;
    tam := char_length(cpftemp);
    temp1 := tam-1;

    -- Cálculo do dígito 1
    index :=1;
    WHILE index <= (tam -2)
    LOOP
        temp2 := CAST(substring(cpftemp from index for 1) AS NUMERIC);
        soma := soma + ( temp2 * temp1);
        temp1 := temp1 - 1;
        index := index + 1;
    END LOOP;

    digito1 := 11 - CAST((soma % 11) AS INTEGER);

    IF (digito1 = 10) THEN digito1 :=0 ; END IF;
    IF (digito1 = 11) THEN digito1 :=0 ; END IF;

    -- Cálculo do dígito 2
    temp1 := 11; soma :=0;
    index :=1;
    WHILE index <= (tam -1)
    LOOP
        soma := soma + CAST((substring(cpftemp FROM index FOR 1)) AS REAL) * temp1;
        temp1 := temp1 - 1;
        index := index + 1;
    END LOOP;

    digito2 := 11 - CAST ((soma % 11) AS INTEGER);

```

```
IF (digito2 = 10) THEN digito2 := 0; END IF;
IF (digito2 = 11) THEN digito2 := 0; END IF;

--Teste final
IF ( (digito1 || " " || digito2) = substring(cpftemp FROM tam-1 FOR 2) )
    THEN RETURN NEW;
ELSE
    RAISE EXCEPTION 'O CPF % é inválido',cpfdado;
END IF;

END $$ language plpgsql;
```