

Guilherme Augusto de Macedo, Matheus Liberato Domingues da Silva, Victor  
Hugo Carlquist da Silva

# **Modelo de Banco de Dados para Gerenciamento de Pizzaria: Modelagem e Implementação**

**Campos do Jordão**

**2013**

Guilherme Augusto de Macedo, Matheus Liberato Domingues da Silva, Victor  
Hugo Carlquist da Silva

## **Modelo de Banco de Dados para Gerenciamento de Pizzaria: Modelagem e Implementação**

Trabalho final apresentado na disciplina de  
Banco de Dados II no quarto módulo do  
curso de Tecnologia em Análise e Desenvolvi-  
mento de Sistemas do IFSP-CJO.

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - *campus* Campos do  
Jordão

Orientador: Paulo Giovani de Faria Zeferino

Campos do Jordão

2013

---

Guilherme Augusto de Macedo, Matheus Liberato Domingues da Silva, Victor Hugo Carlquist da Silva

MODELO DE BANCO DE DADOS PARA GERENCIAMENTO DE PIZZARIA: MODELAGEM E IMPLEMENTAÇÃO/ Guilherme Augusto de Macedo, Matheus Liberato Domingues da Silva, Victor Hugo Carlquist da Silva. – Campos do Jordão, 2013-44 p. : il. (algumas color.) ; 30 cm.

Orientador: Paulo Giovani de Faria Zeferino

Trabalho Final – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - *campus* Campos do Jordão, 2013.

1. Complexidade de Algoritmo. 2. Processamento de Imagens. I. Autor. II. Título III. Orientador. IV. Faculdade. V. Título

CDU 02:141:005.7

---

Guilherme Augusto de Macedo, Matheus Liberato Domingues da Silva, Victor Hugo Carlquist da Silva

## **Modelo de Banco de Dados para Gerenciamento de Pizzaria: Modelagem e Implementação**

Trabalho final apresentado na disciplina de Banco de Dados II no quarto módulo do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFSP-CJO.

### **Banca Examinadora**

03 de dezembro de 2013

---

**Prof. Paulo Giovani de Faria Zeferino**  
Orientador

---

**Prof. Me. Alvaro Costa Neto**  
Convidado 1

---

**Prof. Esp. Alisson Ribeiro**  
Convidado 2

Campos do Jordão  
2013

# RESUMO

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

**Palavras-chaves:** Complexidade de Algoritmos. Processamento de Imagens. Computação Heterogênea.

# ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

**Key-words:** Algorithm Complexity. Image Procesing. Heterogeneous Computing.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Epatas da metodologia . . . . .	10
Figura 2 – . . . . .	11
Figura 3 – . . . . .	11
Figura 4 – . . . . .	12
Figura 5 – . . . . .	13
Figura 6 – . . . . .	14
Figura 7 – . . . . .	14
Figura 8 – . . . . .	15
Figura 9 – . . . . .	16
Figura 10 – Resultado do select . . . . .	24
Figura 11 – Resultado do select . . . . .	25
Figura 12 – Resultado do select . . . . .	26
Figura 13 – Resultado do select . . . . .	27
Figura 14 – Resultado do select . . . . .	28
Figura 15 – Resultado do select . . . . .	29
Figura 16 – Resultado do select . . . . .	30
Figura 17 – Resultado do select . . . . .	31
Figura 18 – Resultado do select . . . . .	32
Figura 19 – Resultado do select . . . . .	33
Figura 20 – Resultado do select . . . . .	33

# LISTA DE TABELAS



# SUMÁRIO

	<b>Introdução</b>	<b>9</b>
<b>1</b>	<b>Metodologia Proposta</b>	<b>10</b>
<b>2</b>	<b>Modelo Conceitual</b>	<b>11</b>
<b>3</b>	<b>Modelo Lógico</b>	<b>14</b>
<b>4</b>	<b>Implementação</b>	<b>17</b>
<b>5</b>	<b>Execução e Testes</b>	<b>24</b>
5.1	Consultas	24
5.2	Procedimentos armazenados	29
	<b>Considerações Finais</b>	<b>34</b>
	<b>Referências</b>	<b>35</b>
	<b>Anexos</b>	<b>36</b>
	<b>ANEXO A – Dados inseridos para teste</b>	<b>37</b>

# INTRODUÇÃO

O projeto proposto tem por objetivo a modelagem conceitual, lógica e física de um projeto de Banco de Dados para gerenciamento/automatização de uma pizzaria. A modelagem foi realizada tomando por base os seguintes requisitos:

1. Opção de realização de pedidos online;
2. Pizzaria delivery;
3. Após cadastro, opção do cliente cadastrar dependentes;
4. Registro de admissão e demissão de funcionários;
5. Log automático das atividades dos funcionários;
6. Controle de estoque com base nos fornecedores e nos ingredientes das pizzas;
7. Esquema de backup automático da base de dados.

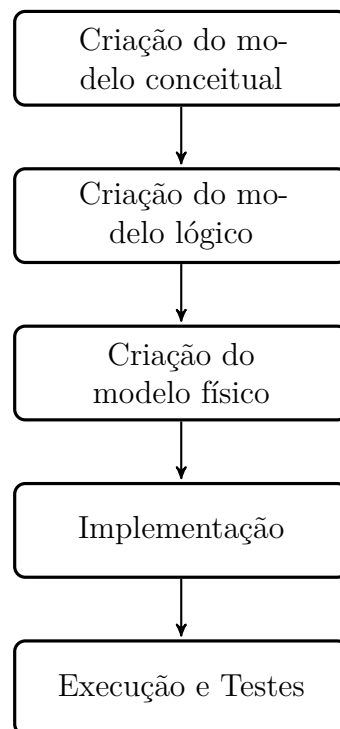
Depois de gerado o modelo físico, implementou-se a solução utilizando o *SQL Server Management Studio*. Com base nessa implementação, consultas, *views*, *triggers*, entre outras rotinas, foram criadas para fins de execução e testes.

Os capítulos seguintes estão divididos em Metodologia Proposta, onde é detalhada a metodologia utilizada para a execução o projeto, seguidos de explicações a respeito do modelo conceitual, lógico e físico. Posteriormente, as consultas realizadas são explicadas, assim como o restante das rotinas elaboradas.

# 1 METODOLOGIA PROPOSTA

Para a execução dessa trabalho a metodologia foi dividida em três etapas: *Criação do modelo conceitual*, *Criação do modelo lógico*, *Criação do modelo físico*, *Implementação* e *Execução e Testes*. A figura 1 ilustra a sequência de execução destas etapas.

Figura 1 – Epatas da metodologia



Fonte: Autor

## 2 MODELO CONCEITUAL

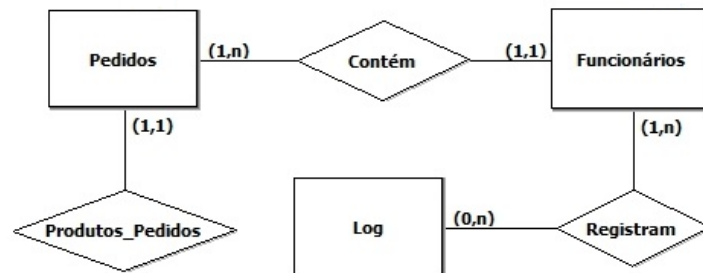


Figura 2 –

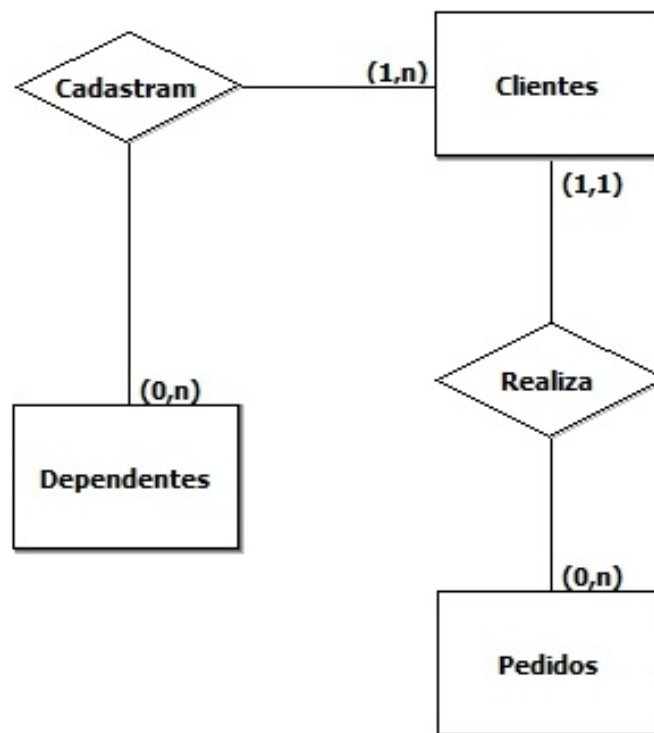


Figura 3 –

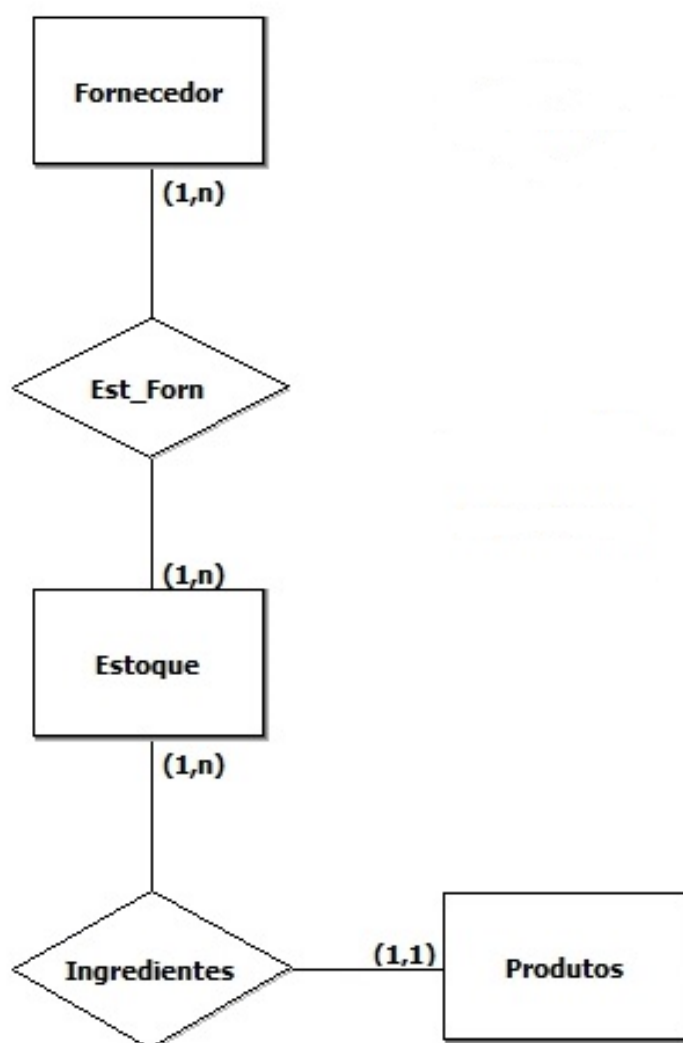


Figura 4 –



### 3 MODELO LÓGICO

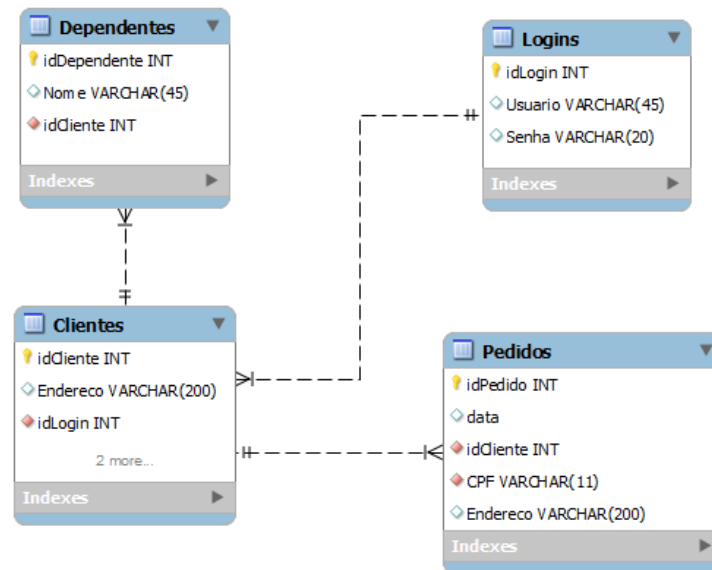


Figura 6 –

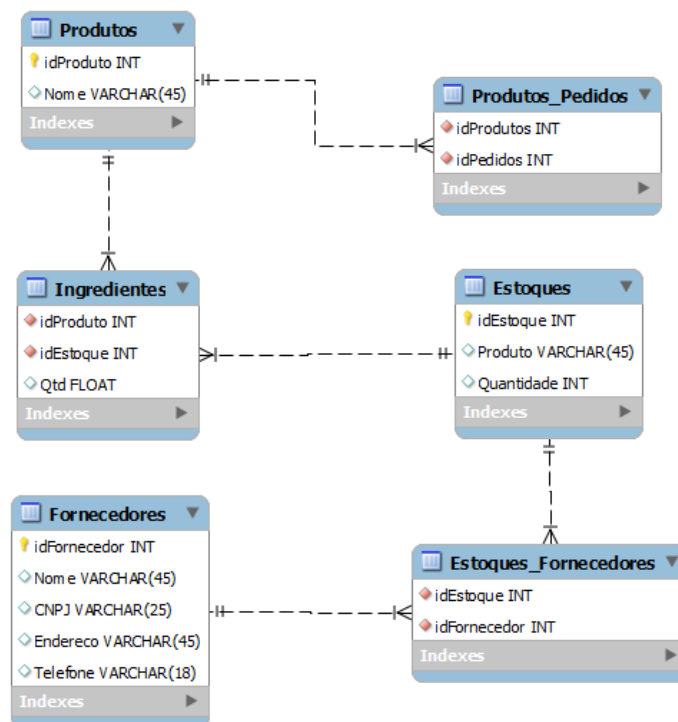


Figura 7 –

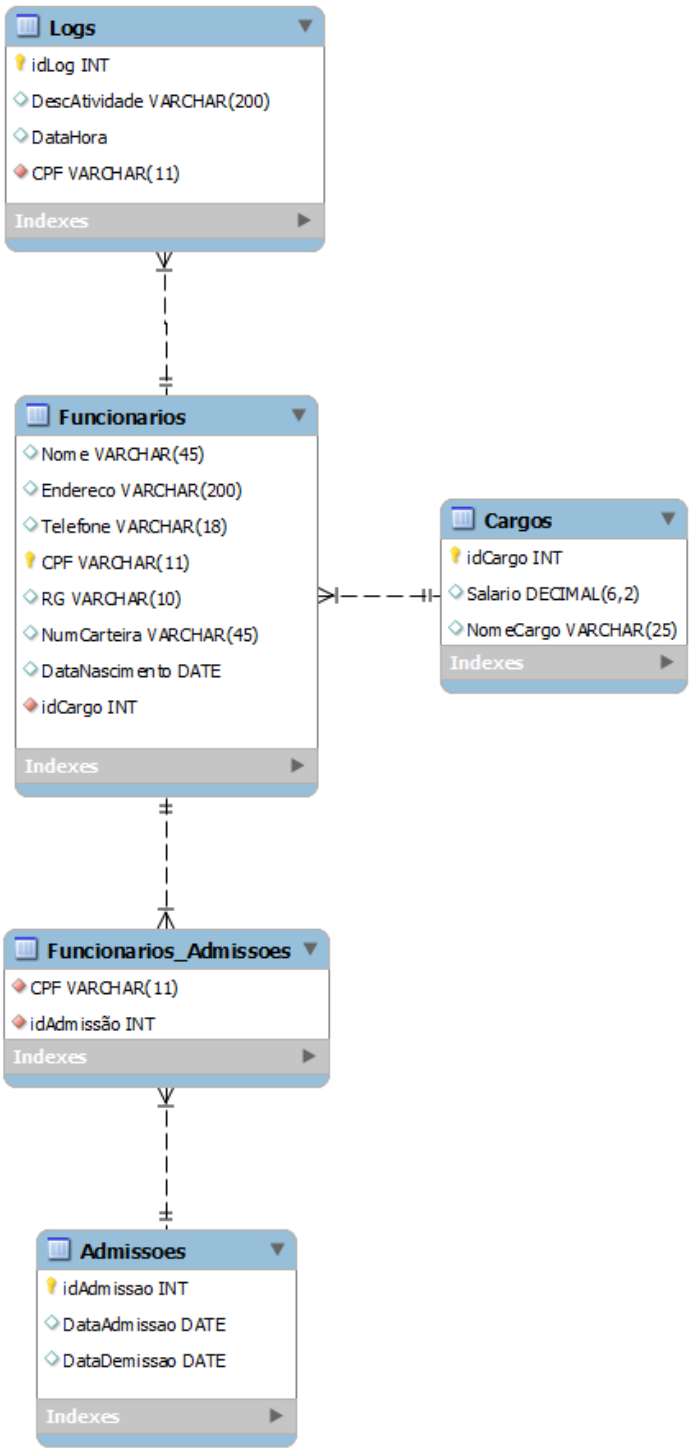


Figura 8 –





Figura 9 –

## 4 IMPLEMENTAÇÃO

O banco de dados foi implementado utilizando o *software SQL Server 2010*.

```
USE master
```

```
GO
```

```
IF EXISTS (select name from sys.databases where name = 'Pizzaria')
    DROP DATABASE Pizzaria
```

```
go
```

```
CREATE DATABASE Pizzaria
```

```
go
```

```
USE Pizzaria
```

```
go
```

```
SET DATEFORMAT dmy
```

```
go
```

---

```
— Table Pizzaria.Logins
```

---

```
CREATE TABLE Logins (
    idLogin INT NOT NULL,
    Usuario VARCHAR(45) NULL,
    Senha VARCHAR(20) NULL,
    PRIMARY KEY (idLogin)
)
GO
```

---

```
— Table Pizzaria.Clientes
```

---

```
CREATE TABLE Clientes (
    idCliente INT NOT NULL PRIMARY KEY,
    Nome VARCHAR (200) NOT NULL,
    Endereco VARCHAR(200) NULL,
```

```
idLogin INT DEFAULT NULL,  
Telefone VARCHAR(18) NULL,  
CONSTRAINT fk_Clientes_Logins  
    FOREIGN KEY (idLogin)  
    REFERENCES Logins (idLogin)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
)  
GO
```

---

```
— Table Pizzaria.Cargos
```

---

```
CREATE TABLE Cargos (  
    idCargo INT NOT NULL,  
    Salario DECIMAL(6,2) NULL,  
    NomeCargo VARCHAR(25) NULL,  
    PRIMARY KEY (idCargo)  
)  
GO
```

---

```
— Table Pizzaria.Funcionarios
```

---

```
CREATE TABLE Funcionarios (  
    Nome VARCHAR(45) NULL,  
    Endereco VARCHAR(200) NULL,  
    Telefone VARCHAR(18) NULL,  
    CPF VARCHAR(11) NOT NULL,  
    RG VARCHAR(10) NULL,  
    NumCarteira VARCHAR(45) NULL,  
    DataNascimento DATE NULL,  
    idCargo INT NOT NULL,  
    PRIMARY KEY (CPF),  
    CONSTRAINT fk_Funcionarios_Cargos  
        FOREIGN KEY (idCargo)  
        REFERENCES Cargos (idCargo)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION
```

```
)  
GO
```

---

```
— Table Pizzaria.Pedidos
```

---

```
CREATE TABLE Pedidos (  
    idPedido INT NOT NULL,  
    data DATETIME NULL,  
    idCliente INT NOT NULL,  
    CPF VARCHAR(11) NOT NULL,  
    Endereco VARCHAR(200) NULL,  
    PRIMARY KEY (idPedido),  
    CONSTRAINT fk_Pedidos_Clientes  
        FOREIGN KEY (idCliente)  
        REFERENCES Clientes (idCliente)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT fk_Pedidos_Funcionarios  
        FOREIGN KEY (CPF)  
        REFERENCES Funcionarios (CPF)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
)  
GO
```

---

```
— Table Pizzaria.Dependentes
```

---

```
CREATE TABLE Dependentes (  
    idDependentes INT NOT NULL,  
    Nome VARCHAR(45) NULL,  
    idCliente INT NOT NULL,  
    PRIMARY KEY (idDependentes),  
    CONSTRAINT fk_Dependentes_Clientes  
        FOREIGN KEY (idCliente)  
        REFERENCES Clientes (idCliente)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION
```

```
)  
GO
```

---

```
-- Table Pizzaria.Produtos
```

---

```
CREATE TABLE Produtos (  
    idProduto INT NOT NULL,  
    Nome VARCHAR(45) NULL,  
    PRIMARY KEY (idProduto)  
)  
GO
```

---

```
-- Table Pizzaria.Estoques
```

---

```
CREATE TABLE Estoques (  
    idEstoque INT NOT NULL,  
    Produto VARCHAR(45) NULL,  
    Quantidade INT NULL,  
    PRIMARY KEY (idEstoque)  
)  
GO
```

---

```
-- Table Pizzaria.Ingredientes
```

---

```
CREATE TABLE Ingredientes (  
    idProduto INT NOT NULL,  
    idEstoque INT NOT NULL,  
    Qtd FLOAT NOT NULL,  
    FOREIGN KEY (idProduto)  
        REFERENCES Produtos (idProduto)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    FOREIGN KEY (idEstoque)  
        REFERENCES Estoques (idEstoque)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION
```

```
)  
GO
```

---

```
— Table Pizzaria.Fornecedores
```

---

```
CREATE TABLE Fornecedores (  
    idFornecedor INT NOT NULL,  
    Nome VARCHAR(45) NULL,  
    CNPJ VARCHAR(25) NULL,  
    Endereco VARCHAR(95) NULL,  
    Telefone VARCHAR(18) NULL,  
    PRIMARY KEY (idFornecedor)  
)  
GO
```

---

```
— Table Pizzaria.Estoques_Fornecedores
```

---

```
CREATE TABLE Estoques_Fornecedores (  
    idEstoque INT NOT NULL,  
    idFornecedor INT NOT NULL,  
    CONSTRAINT fk_Estoque_has_Fornecedor_Estoque  
        FOREIGN KEY (idEstoque)  
        REFERENCES Estoques (idEstoque)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT fk_Estoque_has_Fornecedor_Fornecedor  
        FOREIGN KEY (idFornecedor)  
        REFERENCES Fornecedores (idFornecedor)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
)  
GO
```

---

```
— Table Pizzaria.Produtos_Pedidos
```

---

```
CREATE TABLE Produtos_Pedidos (  
    idProduto INT NOT NULL,  
    idPedido INT NOT NULL,  
    CONSTRAINT fk_Produtos_Pedidos_Produto  
        FOREIGN KEY (idProduto)  
        REFERENCES Produtos (idProduto)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT fk_Produtos_Pedidos_Pedido  
        FOREIGN KEY (idPedido)  
        REFERENCES Pedidos (idPedido)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
)  
GO
```

```
idProduto INT NOT NULL,  
idPedido INT NOT NULL,  
CONSTRAINT fk_Produtos_has_Pedidos_Produtos  
    FOREIGN KEY (idProduto)  
    REFERENCES Produtos (idProduto)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
CONSTRAINT fk_Produtos_has_Pedidos_Pedidos  
    FOREIGN KEY (idPedido)  
    REFERENCES Pedidos (idPedido)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
)  
GO
```

---

```
-- Table Pizzaria.Admissoes
```

---

```
CREATE TABLE Admissoes (  
    idAdmissao INT NOT NULL,  
    DataAdmissao DATE NULL,  
    DataDemissao DATE NULL,  
    PRIMARY KEY (idAdmissao)  
)  
GO
```

---

```
-- Table Pizzaria.Funcionarios_Admissoes
```

---

```
CREATE TABLE Funcionarios_Admissoes (  
    CPF VARCHAR(11) NOT NULL,  
    idAdmissão INT NOT NULL,  
    CONSTRAINT fk_Funcionarios_has_Admissão_Funcionarios  
        FOREIGN KEY (CPF)  
        REFERENCES Funcionarios (CPF)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT fk_Funcionarios_has_Admissão_Admissão  
        FOREIGN KEY (idAdmissão)
```

```
REFERENCES Admissoes (idAdmissao)
ON DELETE NO ACTION
ON UPDATE NO ACTION
)
GO
```

---

```
-- Table Pizzaria.Logs
```

---

```
CREATE TABLE Logs (
    idLog INT NOT NULL,
    DescAtividade VARCHAR(200) NULL,
    DataHora DATETIME NULL,
    CPF VARCHAR(11) NOT NULL,
    PRIMARY KEY (idLog),
    CONSTRAINT fk_Log_Funcionarios
        FOREIGN KEY (CPF)
        REFERENCES Funcionarios (CPF)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
GO
```



## 5 EXECUÇÃO E TESTES

As execuções e os testes foram feitos utilizando o *software SQL Server Management Studio 2010*.

### 5.1 CONSULTAS

```
USE Pizzaria
```

```
go
```

---

```
-- Lista alimentos e seus fornecedores
```

---

```
SELECT Estoques.Produto as [Alimento], Fornecedores.Nome as [Fornecedor]
      INNER JOIN Estoques ON Estoques.idEstoque = Estoques_Fornecedores
      INNER JOIN Fornecedores ON Fornecedores.idFornecedor = Estoques_Fornecedores
      ORDER BY Fornecedores.Nome, Estoques.Produto
```

```
GO
```

	Alimento	Fornecedor
1	Abobrinha	Alimentos Já
2	Bacon	Alimentos Já
3	Beringela	Alimentos Já
4	Calabresa	Alimentos Já
5	Came Seca	Alimentos Já
6	Champignon	Alimentos Já
7	Farinha de Trigo	Alimentos Já
8	Lombo	Alimentos Já
9	Ovo	Alimentos Já
10	Requeijão Cre...	Alimentos Já
11	Bróculis	Boa Massa
12	Cebola	Boa Massa
13	Extrato de To...	Boa Massa
14	Frango desfiado	Boa Massa
15	Manjeriço	Boa Massa
16	Oregano	Boa Massa
17	Palmito	Boa Massa
18	Queijo Mussar...	Boa Massa

Figura 10 – Resultado do select

```
USE Pizzaria
```

```
go
```

```
--
-- Lista os nomes dos produtos, seus ingredientes e a
-- quantidade em estoque
--
```

```
SELECT Produtos.Nome, Estoques.Produto, Estoques.Quantidade FROM Ingredientes
INNER JOIN Produtos ON Produtos.idProduto = Ingredientes.idProduto
INNER JOIN Estoques ON Estoques.idEstoque = Ingredientes.idEstoque
ORDER BY Produtos.Nome, Estoques.Produto
```

```
GO
```

	Nome	Produto	Quantidade
1	Calabresa	Calabresa	7
2	Calabresa	Cebola	13
3	Calabresa	Extrato de Tomate	12
4	Calabresa	Queijo Mussarela	10
5	Frango C/ Catupiry	Extrato de Tomate	12
6	Frango C/ Catupiry	Frango desfiado	14
7	Frango C/ Catupiry	Requeijão Cremoso	10
8	Frango Especial	Bacon	18
9	Frango Especial	Extrato de Tomate	12
10	Frango Especial	Frango desfiado	14
11	Frango Especial	Oregano	4
12	Frango Especial	Requeijão Cremoso	10
13	Lombo	Extrato de Tomate	12
14	Lombo	Extrato de Tomate	12
15	Lombo	Queijo Mussarela	10
16	Margarita	Extrato de Tomate	12
17	Margarita	Manjerição	7
18	Margarita	Queijo Mussarela	10
19	Margarita	Queijo parmesão	13

Figura 11 – Resultado do select

```
USE Pizzaria
```

```
go
```

```
--
-- Lista os clientes e os logins de quem o tiver.
--
```

```
CREATE VIEW ClientesComLogin
```

```
AS
```

```

SELECT Logins.Usuario, Clientes.idCliente FROM Logins
      RIGHT JOIN Clientes ON Logins.idLogin = Clientes.idLogin
GO

SELECT * FROM ClientesComLogin
GO

```

	idCliente	Usuario
1	1	Guilherme
2	2	Matheus
3	3	Victor
4	4	Marcelo
5	5	Pedro
6	6	Joao
7	7	NULL
8	8	NULL
9	9	NULL
10	10	NULL

Figura 12 – Resultado do select

```

USE Pizzaria
go

```

---

```

-- Lista produtos pedidos

```

---

```

CREATE VIEW PedidosRealizados
AS

```

```

      SELECT Produtos.Nome AS [Produto], Pedidos.idCliente FROM Produtos
      INNER JOIN Pedidos ON Pedidos.idProduto = Produtos.idProduto
      INNER JOIN Pedidos ON Pedidos.idPedido = Pedidos.idPedido
GO

```

```

SELECT * FROM PedidosRealizados

```

GO

	Produto	idCliente
1	Calabresa	1
2	Frango C/ Catupiry	1
3	Lombo	1
4	Margarita	2
5	Portuguesa	2
6	Napolitana	4
7	Frango Especial	4
8	Toscana	3
9	Nordestina	2
10	Vegetariana	3

Figura 13 – Resultado do select

```
USE Pizzaria
```

```
go
```

```
-- Lista dos clientes que fizeram pedidos.
```

```
CREATE VIEW ClientesQueFizeramPedidos
```

```
AS
```

```
SELECT ClientesComLogin.Usuario ,
```

```
        PedidosRealizados.Produto
```

```
        FROM PedidosRealizados
```

```
        INNER JOIN ClientesComLogin ON ClientesComLogin.idCliente = PedidosRealizados.idCliente
```

```
GO
```

```
SELECT ClientesQueFizeramPedidos.Usuario , COUNT(*) AS [Quantidade de Pedidos]
```

```
        GROUP BY ClientesQueFizeramPedidos.Usuario
```

```
USE Pizzaria
```

```
go
```

	Usuario	Quantidade de Pedidos
1	Guilherme	3
2	Marcelo	2
3	Matheus	3
4	Victor	2

Figura 14 – Resultado do select

---

```
-- Clientes e seus dependentes
```

---

```
SELECT ClientesComLogin.Usuario, Dependentes.Nome [Nome do dependente] FROM
    INNER JOIN ClientesComLogin ON ClientesComLogin.idCliente = Depen
GO
```

```
USE Pizzaria
go
```

---

```
-- Funcionrios e Cargos
```

---

```
SELECT Funcionarios.Nome, Funcionarios.CPF, Cargos.NomeCargo, Cargos.Sala
    INNER JOIN Cargos ON Cargos.idCargo = Funcionarios.idCargo
    ORDER BY Cargos.NomeCargo, Funcionarios.Nome
GO
```

```
USE Pizzaria
go
```

---

```
-- Funcionrios, cargos e suas admissões
```

	Usuario	Nome do dependente
1	Guilherme	José da Silva
2	Matheus	Bertoldo Moraes
3	Victor	Geovane Cardoso

Figura 15 – Resultado do select

---

```

SELECT Funcionarios.Nome, Admissoes.DataAdmissao, Cargos.NomeCargo, Cargos.idCargo
    INNER JOIN Funcionarios ON Funcionarios.CPF = Funcionarios_Admissao.CPF
    INNER JOIN Admissoes ON Admissoes.idAdmissao = Funcionarios_Admissao.idAdmissao
    INNER JOIN Cargos ON Cargos.idCargo = Funcionarios.idCargo
GO

```

## 5.2 PROCEDIMENTOS ARMAZENADOS

```

USE Pizzaria
GO

```

---

```

-- Stored Procedures – Calcula a idade dos funcionarios

```

---

```

IF EXISTS (select name from sys.procedures where name = 'usp_idadeFuncionarios')
    DROP PROCEDURE usp_idadeFuncionarios
GO

```

```

CREATE PROCEDURE usp_idadeFuncionarios

```

	Nome	CPF	NomeCargo	Salario
1	Amanda Silveira	12332112365	Balconista	1000.00
2	Carlos Eduardo	12332112366	Balconista	1000.00
3	Catarina Santos	12332112361	Balconista	1000.00
4	Miguel de Souza	12332112362	Entregador	1500.00
5	Sérgio Malandro	12332112363	Entregador	1500.00
6	Carlos Belozo	12332112368	Garçon	1500.00
7	Sandra de Sá	12332112369	Garçon	1500.00
8	Roberto Jefferson	12332112364	Gerente	2500.00
9	José Benedito	12332112360	Pizzaiolo	2000.00
10	Miguel de Arrais	12332112367	Pizzaiolo	2000.00

Figura 16 – Resultado do select

AS

```
SELECT Nome, DATEDIFF(YEAR, DataNascimento, GETDATE()) - CASE
    WHEN GETDATE() < DATEADD(YEAR, DA
    THEN 1
    ELSE 0
END AS 'Idade', CONVERT(VARCHAR(
```

```
FROM Funcionarios
```

GO

```
EXEC usp_idadeFuncionarios
```

GO

```
USE Pizzaria
```

GO

---

```
— Stored Procedures – Retorna pedidos realizados
```

---

```
IF EXISTS (select name from sys.procedures where name = 'usp_pedidosRealiz
DROP PROCEDURE usp_pedidosRealizados
```

GO

	Nome	DataAdmissao	NomeCargo	Salario
1	José Benedito	2005-08-30	Pizzaiolo	2000.00
2	Catania Santos	2007-04-28	Balconista	1000.00
3	Miguel de Souza	2009-06-30	Entregador	1500.00
4	Sérgio Malandro	2009-10-14	Entregador	1500.00
5	Roberto Jefferson	2010-08-15	Gerente	2500.00
6	Amanda Silveira	2010-08-25	Balconista	1000.00
7	Carlos Eduardo	2011-09-30	Balconista	1000.00
8	Miguel de Arrais	2011-10-01	Pizzaiolo	2000.00
9	Carlos Belozo	2011-11-30	Garçon	1500.00
10	Sandra de Sá	2012-04-01	Garçon	1500.00

Figura 17 – Resultado do select

```

CREATE PROCEDURE usp_pedidosRealizados
    @nome VARCHAR(45)
AS
    SELECT F.Nome, C.NomeCargo as 'Cargo', Prod.Nome as 'Nome Produto'
FROM Funcionarios F
    INNER JOIN Cargos C ON C.idCargo = F.idCargo
    INNER JOIN Pedidos P ON P.CPF = F.CPF
    INNER JOIN Produtos_Pedidos PP ON PP.idPedido = P.idPedido
    INNER JOIN Produtos Prod ON Prod.idProduto = PP.idProduto
    WHERE F.Nome = @nome
GO

EXEC usp_pedidosRealizados "Guilherme"
GO

USE Pizzaria
GO

```

---

— Stored Procedures – Retorna os pedidos do cliente



	Nome	Idade	Data de Nascimento
1	José Benedito	31	14/09/1982
2	Catarina Santos	32	18/09/1981
3	Miguel de Souza	26	08/08/1987
4	Sérgio Malandro	28	18/04/1985
5	Roberto Jefferson	44	01/12/1969
6	Amanda Silveira	39	23/07/1974
7	Carlos Eduardo	40	02/03/1973
8	Miguel de Arrais	22	09/12/1990
9	Carlos Belozo	23	15/08/1990
10	Sandra de Sá	22	11/09/1991

Figura 18 – Resultado do select

```

IF EXISTS (select name from sys.procedures where name = 'usp_pedidosRealizados')
    DROP PROCEDURE usp_pedidosRealizadosCliente
GO

CREATE PROCEDURE usp_pedidosRealizadosCliente
    @nome VARCHAR(45)
AS
    SELECT Cli.Nome,      as 'Cargo',  Prod.Nome, CONVERT(VARCHAR(10),P.data) as 'Data'
    FROM
        Clientes Cli
        INNER JOIN Cargos C ON C.idCargo = F.idCargo
        INNER JOIN Pedidos P ON P.CPF = F.CPF
        INNER JOIN Produtos_Pedidos PP ON PP.idPedido = P.idPedido
        INNER JOIN Produtos Prod ON Prod.idProduto = PP.idProduto
    WHERE F.Nome = @nome
GO

EXEC usp_pedidosRealizados 'Miguel de Souza'
GO

```

	Nome	Cargo	Nome Produto	Data do Pedido
1	Sérgio Malandro	Entregador	Calabresa	01/12/2013
2	Sérgio Malandro	Entregador	Frango C./ Catupiry	01/12/2013
3	Sérgio Malandro	Entregador	Lombo	01/12/2013
4	Miguel de Souza	Entregador	Margarita	30/11/2013
5	Miguel de Souza	Entregador	Portuguesa	30/11/2013
6	Miguel de Souza	Entregador	Napolitana	30/11/2013
7	Miguel de Souza	Entregador	Frango Especial	30/11/2013
8	Sérgio Malandro	Entregador	Toscana	30/11/2013
9	Miguel de Souza	Entregador	Nordestina	30/11/2013
10	Sérgio Malandro	Entregador	Vegetariana	30/11/2013

Figura 19 – Resultado do select

	Nome	Cargo	Nome Produto	Data do Pedido
1	Sérgio Malandro	Entregador	Calabresa	01/12/2013
2	Sérgio Malandro	Entregador	Frango C./ Catupiry	01/12/2013
3	Sérgio Malandro	Entregador	Lombo	01/12/2013
4	Miguel de Souza	Entregador	Margarita	30/11/2013
5	Miguel de Souza	Entregador	Portuguesa	30/11/2013
6	Miguel de Souza	Entregador	Napolitana	30/11/2013
7	Miguel de Souza	Entregador	Frango Especial	30/11/2013
8	Sérgio Malandro	Entregador	Toscana	30/11/2013
9	Miguel de Souza	Entregador	Nordestina	30/11/2013
10	Sérgio Malandro	Entregador	Vegetariana	30/11/2013

Figura 20 – Resultado do select

## CONSIDERAÇÕES FINAIS

Apesar de parecer simples, criar um banco de dados para uma pizzeria mostrou-se uma tarefa cheia de detalhes a se pensar. Ao ser implementado, tornou-se funcional, sendo possível utilizá-lo em um ambiente real.

## REFERÊNCIAS

## Anexos

# ANEXO A – DADOS INSERIDOS PARA TESTE

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

USE Pizzaria  
GO

---

— Table Pizzaria.Logins

---

INSERT INTO Logins VALUES

```
(1, 'Guilherme', 'egmdc321'),
(2, 'Matheus', 'egmdc321'),
(3, 'Victor', 'egmdc321'),
(4, 'Marcelo', 'egmdc321'),
(5, 'Pedro', 'egmdc321'),
(6, 'Joao', 'egmdc321');
```

GO

---

— Table Pizzaria.Clientes

---

INSERT INTO Clientes VALUES

```
(1, 'Robervaldo', 'Av Ministro Nelson Hungria, 280, Centro, Santo
(2, 'Valdomiro', 'Av Coronel Sebastio Marcondes da Silva, 149, C
(3, 'Cleidiane', 'Rua Sao Joo, 455, Centro, So Jos de Campos-SP -
(4, 'Wanilda', 'Rua Quinze De Novembro, 394, Centro, Taubat-SP -
(5, 'Soleneusa', 'Rua Sao Sebastiao, 289, Centro, Trememb-SP - CE
(6, 'Godofredo', 'Rua Santos Dumont, 876, Centro, Ubatuba-SP - CE
```

```
(7, 'Jaime', 'Rua Belo Horizonte, 255, Centro, Londrina-PR - CEP
(8, 'Jean', 'Rua Jos Bonifcio, 580, Centro, Maring-PR - CEP 12440
(9, 'Claudisney', 'Rua Vinte e Trs, 290, Centro, Barbosa-SP - CEP
(10, 'Flvio', 'Rua Santa Rita, 276, Centro, Manaus-AM - CEP 12440
```

GO

---

```
— Table Pizzaria.Cargos
```

---

```
INSERT INTO Cargos VALUES
```

```
(1, 1500, 'Entregador'),
(2, 1000, 'Balconista'),
(3, 2500, 'Gerente'),
(4, 2000, 'Pizzaiolo'),
(5, 1500, 'Garon');
```

GO

---

```
— Table Pizzaria.Funcionarios
```

---

```
INSERT INTO Funcionarios VALUES
```

```
('Roberto Jefferson', 'Rua Conde de Bobadela, 225, Centro, Rio Br
('Amanda Silveira', 'Rua Senador Rocha Lagoa, 235, Centro, Cuiab
('Carlos Eduardo', 'Praa Reinaldo Alves de Brito, 325, Centro, C
('Miguel de Arrais', 'Rua Conde de Bobadela, 223, Centro, Joo Pes
('Carlos Belozo', 'Praa Silviano Brando, 245, Centro, Belm', '9-9
('Sandra de S', 'Rua Conde de Bobadela, 224, Centro, Teresina',
('Srgio Malandro', 'Rua Alvarenga, 425, Centro, Natal', '9-9909-
('Miguel de Souza', 'Rua Randolpho Bretas, 525, Centro, Porto Aleg
('Catarina Santos', 'Rua Antnio de Albuquerque, 255, Centro, Flor
('Jos Benedito', 'Praa Baro do Rio Branco, 909, Centro, Aracaj',
```

GO

---

```
— Table Pizzaria.Pedidos
```

---

```
INSERT INTO Pedidos VALUES
```

```
(1, '01-12-2013', 1, '12332112363', 'AV Andrmeda, 720, C
(2, '30-11-2013', 2, '12332112362', 'Av Anahanguera, 820,
```

```
(3, '30-11-2013', 3, '12332112363', 'Rua So Joo, 520, Cer  
(4, '30-11-2013', 4, '12332112362', 'Rua Nelson de Ftima,  
(5, '29-11-2013', 5, '12332112362', 'Rua Sebastio da Ros  
(6, '01-12-2013', 6, '12332112363', 'AV Andrmeda, 720, C  
(7, '30-11-2013', 7, '12332112362', 'Av Anahanguera, 820,  
(8, '30-11-2013', 8, '12332112363', 'Rua So Joo, 520, Cer  
(9, '30-11-2013', 9, '12332112362', 'Rua Nelson de Ftima,  
(10, '29-11-2013', 10, '12332112362', 'Rua Sebastio da F
```

GO

---

```
— Table Pizzaria.Dependentes
```

---

```
INSERT INTO Dependentes VALUES  
    (1, 'Jos da Silva ', 1),  
    (2, 'Bertoldo Moraes ', 2),  
    (3, 'Geovane Cardoso ', 3)
```

GO

---

```
— Table Pizzaria.Produtos
```

---

```
INSERT INTO Produtos VALUES  
    (1, 'Calabresa '),  
    (2, 'Frango C/ Catupiry '),  
    (3, 'Lombo '),  
    (4, 'Margarita '),  
    (5, 'Portuguesa '),  
    (6, 'Napolitana '),  
    (7, 'Frango Especial '),  
    (8, 'Toscana '),  
    (9, 'Nordestina '),  
    (10, 'Vegetariana ')
```

GO

---

```
— Table Pizzaria.Estoques
```

---

```
INSERT INTO Estoques VALUES  
    (1, 'Extrato de Tomate ', 12),
```



```
(2,'Requeijo Cremoso',10),
(3,'Farinha de Trigo',20),
(4,'Queijo Mussarela',10),
(5,'Frango desfiado',14),
(6,'Oregano',4),
(7,'Calabresa',7),
(8,'Bacon',18),
(9,'Ovo',29),
(10,'Cebola',13),
(11,'Queijo parmeso',13),
(12,'Manjerico',7),
(13,'Abobrinha',3),
(14,'Beringela',9),
(15,'Brucalis',10),
(16,'Palmito',21),
(17,'Champignon',12),
(18,'Lombo',11),
(19,'Tomate',2),
(20,'Carne Seca',2);
```

GO

---

— Table Pizzaria.Ingredientes

---

INSERT INTO Ingredientes VALUES

```
(1,1,1),
(1,4,1),
(1,7,1),
(1,10,1),
```

```
(2,1,1),
(2,5,1),
(2,2,1),
```

```
(3,1,1),
(3,4,1),
(3,1,18),
```

```
(4,1,1),
```

(4,4,1) ,  
(4,19,1) ,  
(4,11,1) ,  
(4,12,1) ,

(5,1,1) ,  
(5,9,1) ,  
(5,4,1) ,  
(5,10,1) ,

(6,1,1) ,  
(6,4,1) ,  
(6,11,1) ,  
(6,19,1) ,

(7,1,1) ,  
(7,5,1) ,  
(7,2,1) ,  
(7,8,1) ,  
(7,6,1) ,

(8,1,1) ,  
(8,4,1) ,  
(8,7,1) ,  
(8,6,1) ,

(9,1,1) ,  
(9,2,1) ,  
(9,20,1) ,  
(9,10,1) ,

(10,1,1) ,  
(10,13,1) ,  
(10,14,1) ,  
(10,15,1) ,  
(10,16,1) ,  
(10,17,1);

GO

---

— Table Pizzaria.Fornecedores

---

INSERT INTO Fornecedores VALUES

(1,'Alimentos J', '01010101-01010','Rua Carlos Bom Tempo, 2215, C  
(2,'Boa Massa', '02020202-02020','Rua Conde de Monte Cristo, 21,  
(3, 'Frutas ATC', '091942-00130', 'Av. Ministro Celso de Melo, 24  
(4, 'RMC Verduras', '192814-9049', 'Av do Povo, 545, Taubat', '54

GO

---

— Table Pizzaria.Estoques\_Fornecedores

---

INSERT INTO Estoques\_Fornecedores VALUES

(1,2),  
(2,1),  
(3,1),  
(4,2),  
(5,2),  
(6,2),  
(7,1),  
(8,1),  
(9,1),  
(10,2),  
(11,2),  
(12,2),  
(13,1),  
(14,1),  
(15,2),  
(16,2),  
(17,1),  
(18,1),  
(19,2),  
(20,1)

GO

---

— Table Pizzaria.Produtos\_Pedidos

---

INSERT INTO Produtos\_Pedidos VALUES

```
(1, 1),  
(2, 1),  
(3, 1),  
(4, 2),  
(5, 2),  
(6, 4),  
(7, 4),  
(8, 3),  
(9, 2),  
(10, 3)
```

GO

---

```
-- Table Pizzaria.Admissoes
```

---

```
INSERT INTO Admissoes VALUES
```

```
(1, '30-08-2005', ''),  
(2, '28-04-2007', '11-07-2007'),  
(3, '30-06-2009', ''),  
(4, '14-10-2009', ''),  
(5, '15-08-2010', ''),  
(6, '25-08-2010', ''),  
(7, '30-09-2011', ''),  
(8, '01-10-2011', ''),  
(9, '30-11-2011', ''),  
(10, '01-04-2012', '')
```

GO

---

```
-- Table Pizzaria.Funcionarios_Admissoes
```

---

```
INSERT INTO Funcionarios_Admissoes VALUES
```

```
('12332112360', 1),  
( '12332112361', 2),  
( '12332112362', 3),  
( '12332112363', 4),  
( '12332112364', 5),  
( '12332112365', 6),  
( '12332112366', 7),
```

```
( '12332112367' , 8) ,  
( '12332112368' , 9) ,  
( '12332112369' , 10)
```

```
GO
```

---

```
-- Table Pizzaria.Logs
```

---