

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, 2013-40 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 Desenvolvi-
mento 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 – Epistas da metodologia	10
Figura 2 – Utilização de triggers para alimentar a tabela <i>Log</i>	12
Figura 3 – Entidades: Dependentes, Clientes e Pedidos.	13
Figura 4 – Entidades: Fornecedor, Estoque e Produtos	14
Figura 5 – Modelo Conceitual Completo.	15
Figura 6 – Modelo Lógico: Dependentes, Clientes, Logins e Pedidos	16
Figura 7 – Modelo Lógico: Produtos, Ingredientes, Produtos_Pedidos, Estoques, Estoques_Fornecedores e Fornecedores.	17
Figura 8 – Modelo Lógico: Logs, Funcionários, Cargos, Funcionarios_Admissoes e Admissoes.	18
Figura 9 – Modelo Lógico completo.	19
Figura 10 –Resultado do select Lista alimentos e seus fornecedores	27
Figura 11 –Resultado do select	28
Figura 12 –Resultado do select lista os clientes e os logins de quem o tiver.	29
Figura 13 –Resultado do select lista produtos pedidos.	30
Figura 14 –Resultado do select lista dos clientes que fizeram pedidos.	31
Figura 15 –Resultado do select clientes e seus dependentes	31
Figura 16 –Resultado do select Funcionários e Cargos	32
Figura 17 –Resultado do select funcionários, cargos e suas admissões	33
Figura 18 –Procedimento Armazenado para calcular idade.	34
Figura 19 –Resultado do procedimento armazenado que retorna os pedidos reali- zados.	35
Figura 20 –Procedimento Armazenado que retorna os pedidos de um determinado cliente via parâmetro do nome.	36
Figura 21 –Procedimento Armazenado que retorna os pedidos de um determinado cliente via parâmetro do nome.	36

LISTA DE TABELAS

SUMÁRIO

Introdução	9
1 Metodologia Proposta	10
2 Regras de Negócio	11
3 Modelo Conceitual	12
4 Modelo Lógico	16
5 Implementação	20
6 Execução e Testes	26
6.1 Consultas	26
6.2 Procedimentos armazenados	33
Considerações Finais	37
Referências	38
Anexos	39
ANEXO A Dados inseridos para teste	40

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.

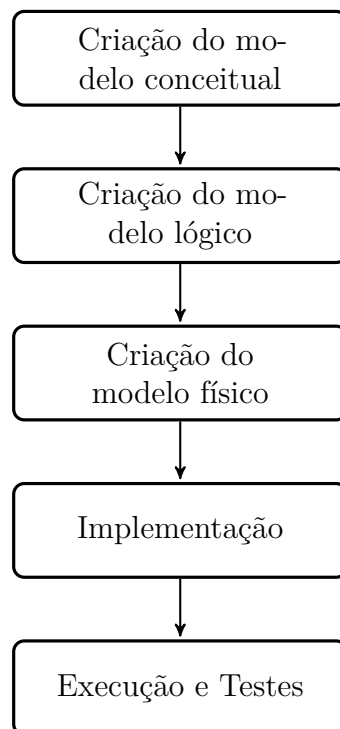
Depois de gerado o modelo físico, implementou-se a solução utilizando o *SQL Server Management Studio*. Com base nessa implementação, consultas, *views*, *triggers*, entre outras rotinas, foram criadas para fins de execução e testes.

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 – Etapas da metodologia



Fonte: Autor

2 REGRAS DE NEGÓCIO

A modelagem foi realizada tomando por base as seguintes regras de negócio 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.

3 MODELO CONCEITUAL

O modelo conceitual foi elaborado no programa *BrModelo*. A Figura 2 mostra como foi feita a modelagem para que os pedidos realizados pelos funcionários fossem armazenados na tabela *Log*. Isso é feito através de triggers.

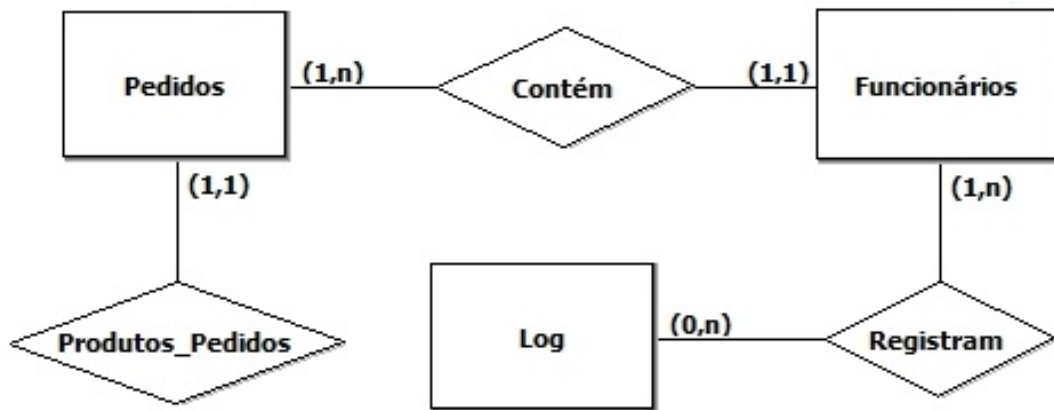


Figura 2 – Utilização de triggers para alimentar a tabela *Log*

Na Figura 3 é possível notar que cada funcionário pode ter nenhum ou vários dependentes. Também é possível observar que os clientes podem realizar nenhum ou vários pedidos, mas cada pedido pertence a um único cliente.

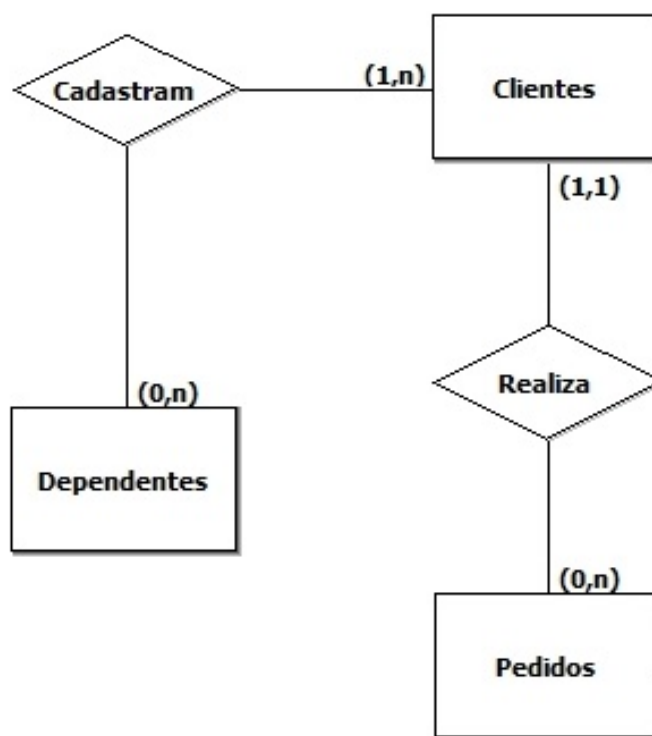


Figura 3 – Entidades: Dependentes, Clientes e Pedidos.

De acordo com a Figura 4, é possível observar que o Fornecedor alimenta o estoque e os produtos são feitos com ingredientes retirados do estoque.

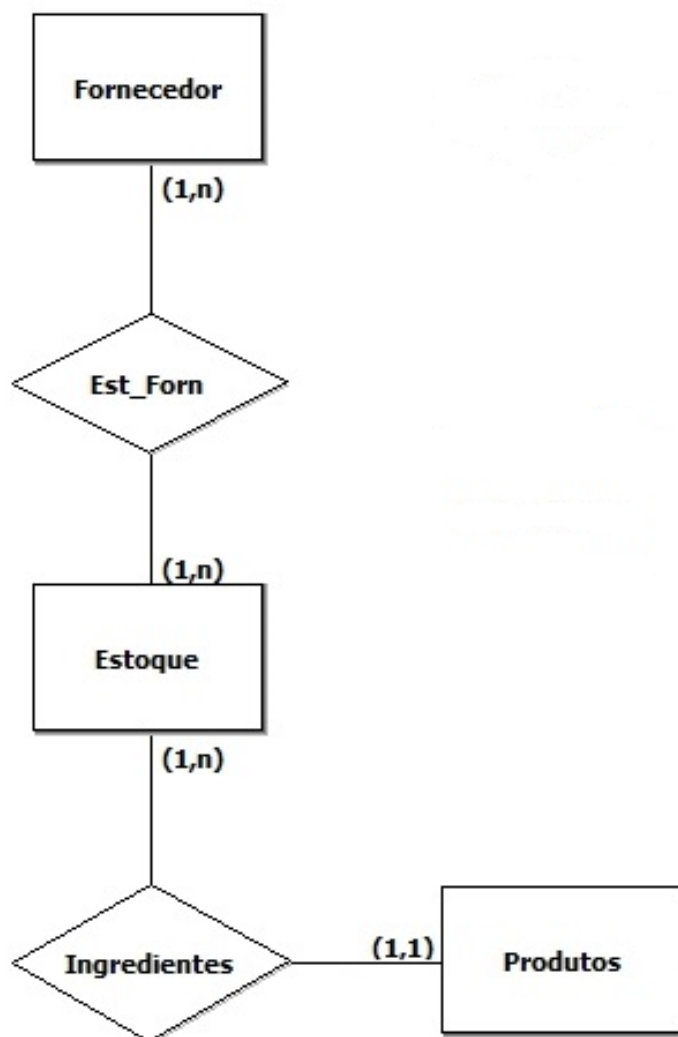


Figura 4 – Entidades: Fornecedor, Estoque e Produtos

Na Figura 5 é possível observar como ficou a modelagem completa do sistema.

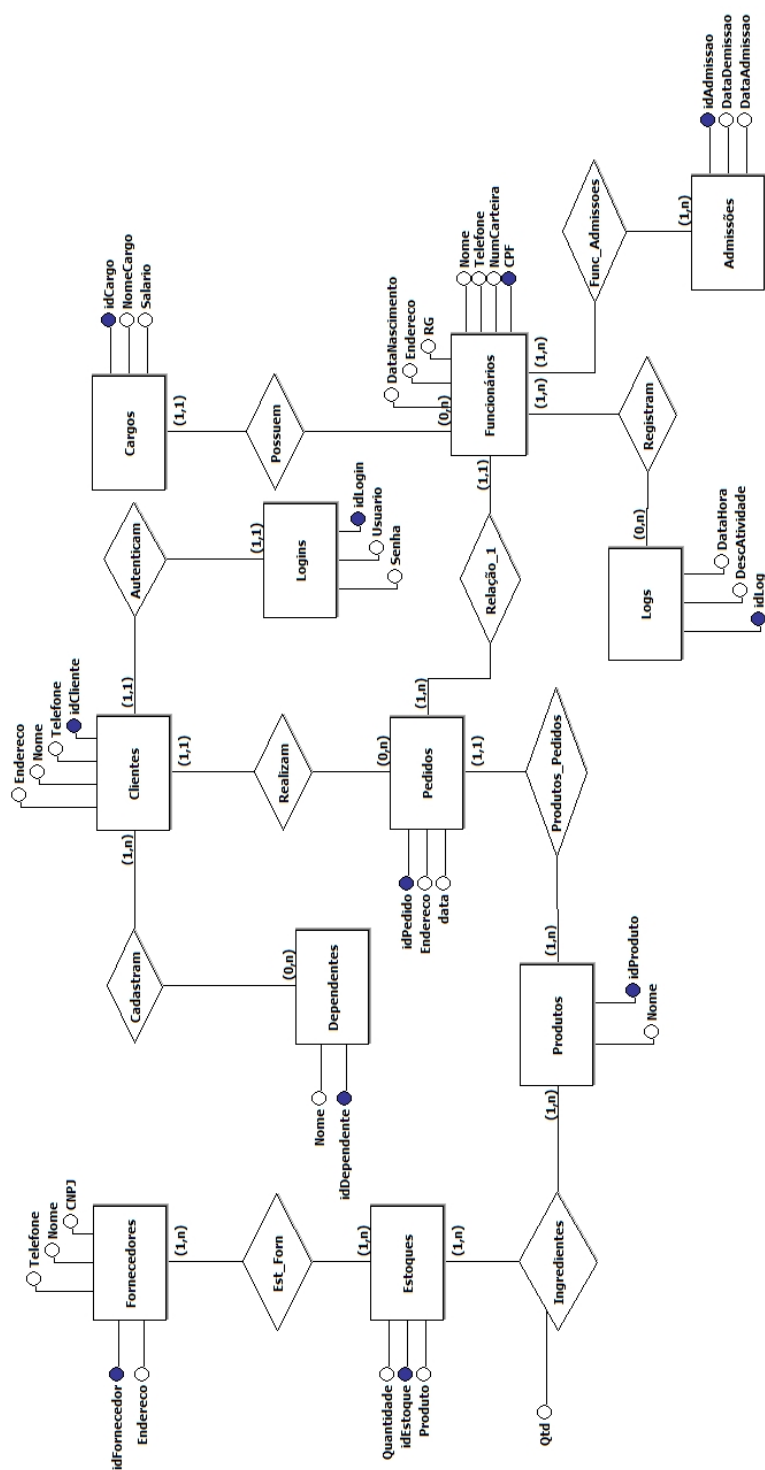


Figura 5 – Modelo Conceitual Completo.

4 MODELO LÓGICO

A Figura 6 representa, conforme o modelo conceitual, a possibilidade do cliente ter ou não login. Isso não impede que o mesmo efetue pedido. Isso aconteceria, por exemplo, no caso do cliente nunca ter feito pedido online.

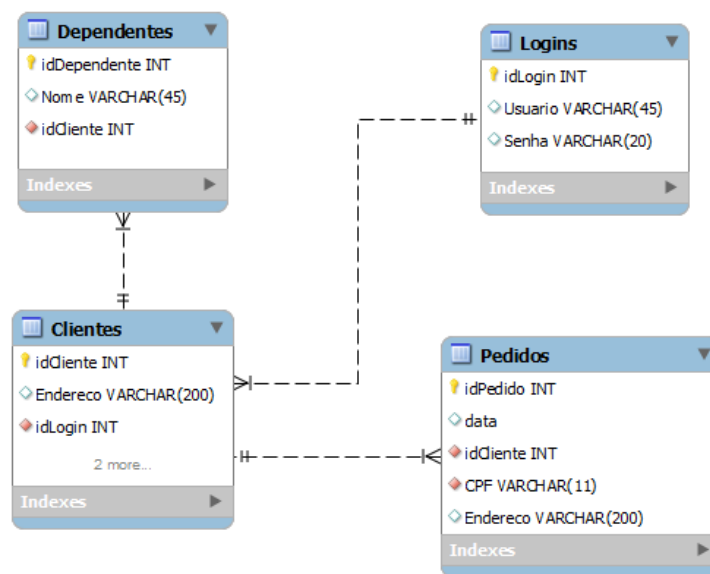


Figura 6 – Modelo Lógico: Dependentes, Clientes, Logins e Pedidos

Na Figura 7 é possível observar os produtos sendo compostos por um ou mais ingredientes; os ingredientes sendo compostos por um ou mais itens do estoque, mas cada item do estoque podendo ser utilizado apenas em uma lista de ingredientes. Também é possível observar a tabela Estoques_Fornecedores, podendo conter vários fornecedores vários itens para o estoque.

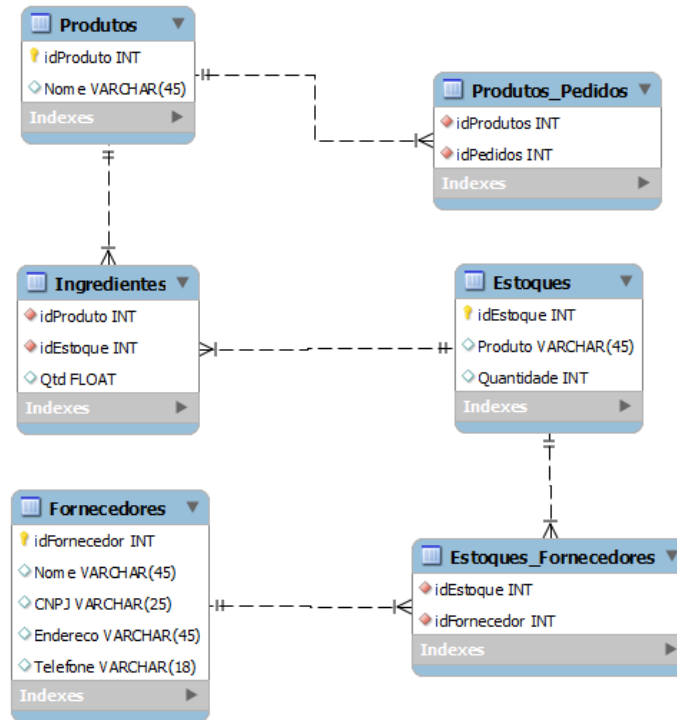


Figura 7 – Modelo Lógico: Produtos, Ingredientes, Produtos_Pedidos, Estoques, Estoques_Fornecedores e Fornecedores.

A Figura 8 mostra a tabela Logs dos funcionários. Essa tabela guarda todas as ações dos funcionários para possível auditorias. É possível observar também que os funcionários têm cargos e cada cargo pode ter muitos funcionários, mas cada funcionários pode ter apenas um cargo na empresa. Como um funcionário pode ser demitido e depois recontraído, existe uma tabela chamada *Funcionarios_Admissoes* onde são salvas as informações a respeito da contratação dos funcionários.

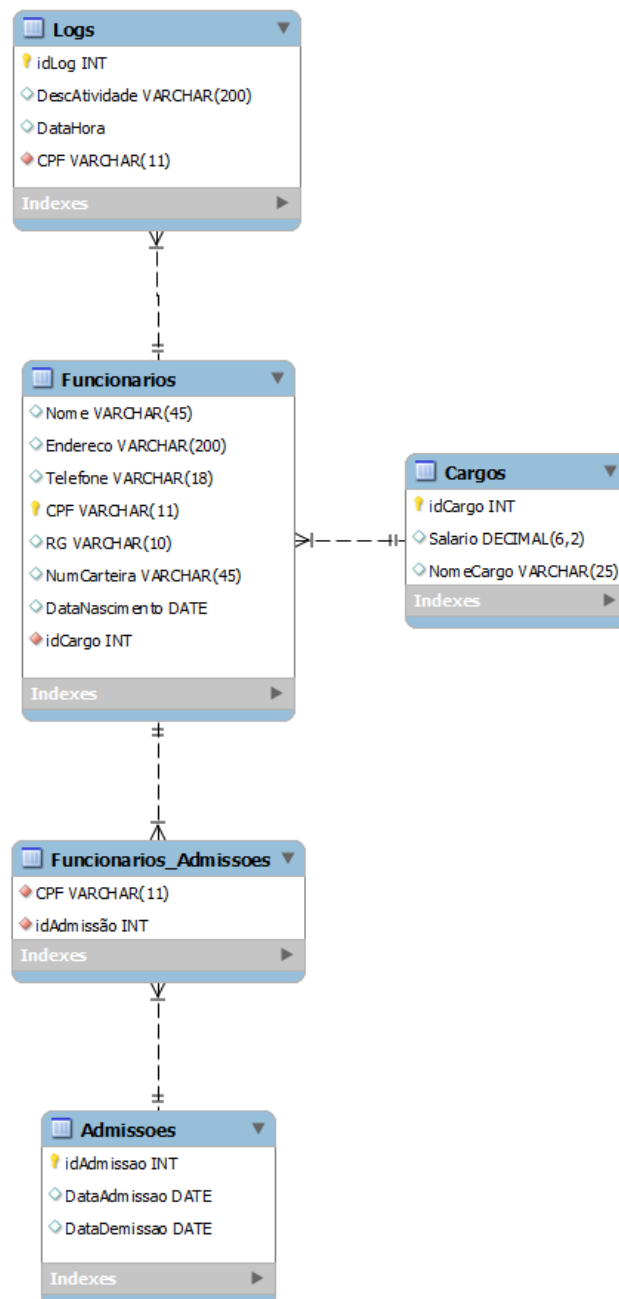


Figura 8 – Modelo Lógico: Logs, Funcionários, Cargos, Funcionarios_Admissoes e Admissoes.

A Figura 9 contém o modelo lógico completo.

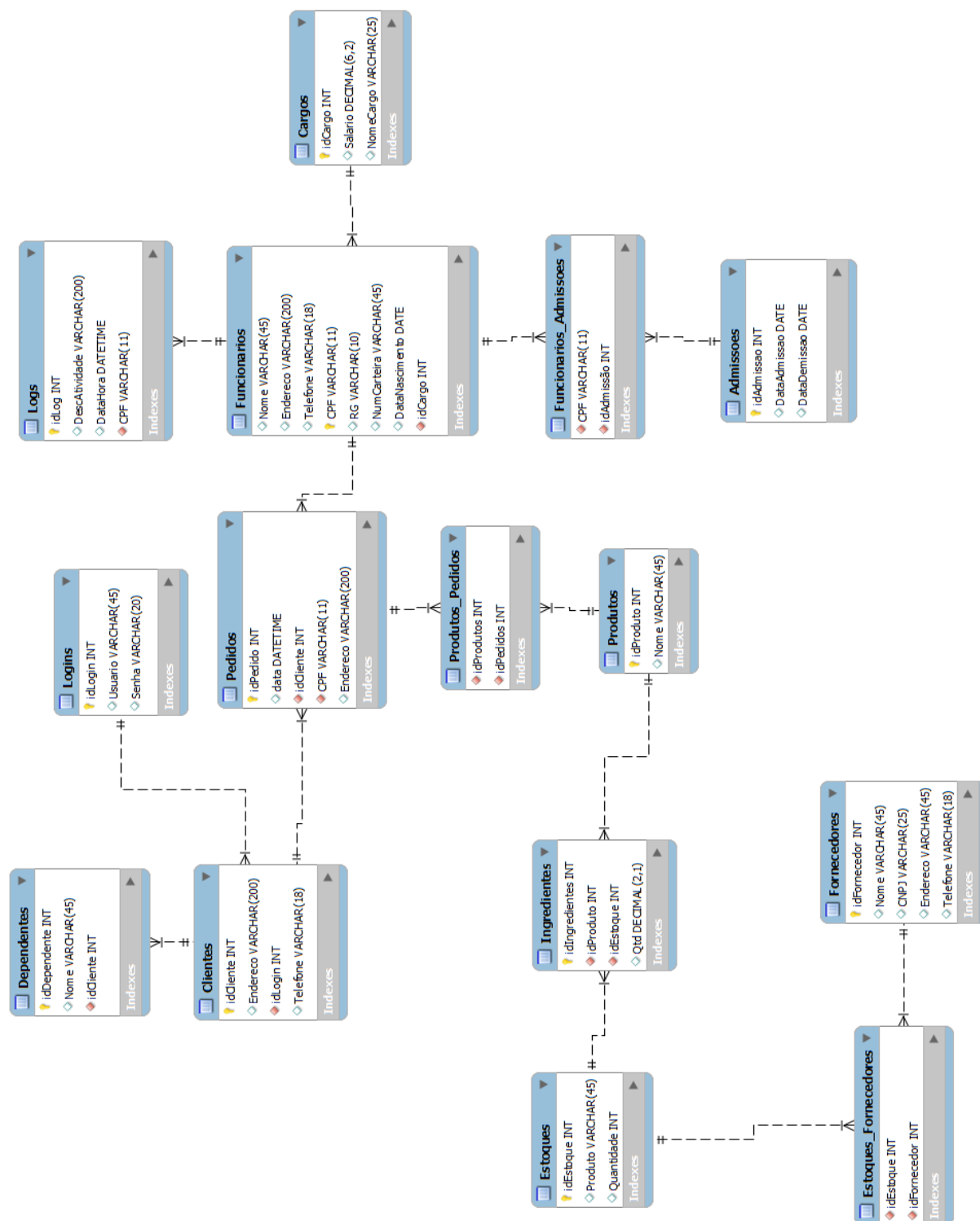


Figura 9 – Modelo Lógico completo.

5 IMPLEMENTAÇÃO

O banco de dados foi implementado utilizando o *software SQL Server 2010*. Segue o código de execução para a criação das tabelas:

```

1      USE master
2      GO
3
4      IF EXISTS (select name from sys.databases where name = 'Pizzaria
           ')
5          DROP DATABASE Pizzaria
6      go
7
8      CREATE DATABASE Pizzaria
9      go
10
11     USE Pizzaria
12     go
13
14     SET DATEFORMAT dmy
15     go
16
17     -----
18     -- Table Pizzaria.Logins
19     -----
20     CREATE TABLE Logins (
21         idLogin INT NOT NULL ,
22         Usuario VARCHAR(45) NULL ,
23         Senha VARCHAR(20) NULL ,
24         PRIMARY KEY (idLogin)
25     )
26     GO
27
28     -----
29     -- Table Pizzaria.Clientes
30     -----
31     CREATE TABLE Clientes (
32         idCliente INT NOT NULL PRIMARY KEY ,
33         Nome VARCHAR (200) NOT NULL ,
34         Endereco VARCHAR(200) NULL ,
35         idLogin INT DEFAULT NULL ,
36         Telefone VARCHAR(18) NULL ,
37         CONSTRAINT fk_Clientes_Logins
38             FOREIGN KEY (idLogin)
39             REFERENCES Logins (idLogin)

```

```
40         ON DELETE NO ACTION
41         ON UPDATE NO ACTION
42     )
43 GO
44
45 -----
46 -- Table Pizzaria.Cargos
47 -----
48 CREATE TABLE Cargos (
49     idCargo INT NOT NULL,
50     Salario DECIMAL(6,2) NULL,
51     NomeCargo VARCHAR(25) NULL,
52     PRIMARY KEY (idCargo)
53 )
54 GO
55
56 -----
57 -- Table Pizzaria.Funcionarios
58 -----
59 CREATE TABLE Funcionarios (
60     Nome VARCHAR(45) NULL,
61     Endereco VARCHAR(200) NULL,
62     Telefone VARCHAR(18) NULL,
63     CPF VARCHAR(11) NOT NULL,
64     RG VARCHAR(10) NULL,
65     NumCarteira VARCHAR(45) NULL,
66     DataNascimento DATE NULL,
67     idCargo INT NOT NULL,
68     PRIMARY KEY (CPF),
69     CONSTRAINT fk_Funcionarios_Cargos
70         FOREIGN KEY (idCargo)
71         REFERENCES Cargos (idCargo)
72         ON DELETE NO ACTION
73         ON UPDATE NO ACTION
74 )
75 GO
76
77 -----
78 -- Table Pizzaria.Pedidos
79 -----
80 CREATE TABLE Pedidos (
81     idPedido INT NOT NULL,
82     data DATETIME NULL,
83     idCliente INT NOT NULL,
84     CPF VARCHAR(11) NOT NULL,
85     Endereco VARCHAR(200) NULL,
86     PRIMARY KEY (idPedido),
```

```
87         CONSTRAINT fk_Pedidos_Clientes
88         FOREIGN KEY (idCliente)
89         REFERENCES Clientes (idCliente)
90         ON DELETE NO ACTION
91         ON UPDATE NO ACTION,
92     CONSTRAINT fk_Pedidos_Funcionarios
93     FOREIGN KEY (CPF)
94     REFERENCES Funcionarios (CPF)
95     ON DELETE NO ACTION
96     ON UPDATE NO ACTION
97 )
98 GO
99
100 -----
101 -- Table Pizzaria.Dependentes
102 -----
103 CREATE TABLE Dependentes (
104     idDependentes INT NOT NULL,
105     Nome VARCHAR(45) NULL,
106     idCliente INT NOT NULL,
107     PRIMARY KEY (idDependentes),
108     CONSTRAINT fk_Dependentes_Clientes
109     FOREIGN KEY (idCliente)
110     REFERENCES Clientes (idCliente)
111     ON DELETE NO ACTION
112     ON UPDATE NO ACTION
113 )
114 GO
115
116 -----
117 -- Table Pizzaria.Produtos
118 -----
119 CREATE TABLE Produtos (
120     idProduto INT NOT NULL,
121     Nome VARCHAR(45) NULL,
122     PRIMARY KEY (idProduto)
123 )
124 GO
125
126 -----
127 -- Table Pizzaria.Estoques
128 -----
129 CREATE TABLE Estoques (
130     idEstoque INT NOT NULL,
131     Produto VARCHAR(45) NULL,
132     Quantidade INT NULL,
133     PRIMARY KEY (idEstoque)
```

```
134 )
135 GO
136
137
138 -----
139 -- Table Pizzaria.Ingredientes
140 -----
141 CREATE TABLE Ingredientes (
142     idProduto INT NOT NULL,
143     idEstoque INT NOT NULL,
144     Qtd FLOAT NOT NULL,
145     FOREIGN KEY (idProduto)
146         REFERENCES Produtos (idProduto)
147         ON DELETE NO ACTION
148         ON UPDATE NO ACTION,
149     FOREIGN KEY (idEstoque)
150         REFERENCES Estoques (idEstoque)
151         ON DELETE NO ACTION
152         ON UPDATE NO ACTION
153 )
154 GO
155
156 -----
157 -- Table Pizzaria.Fornecedores
158 -----
159 CREATE TABLE Fornecedores (
160     idFornecedor INT NOT NULL,
161     Nome VARCHAR(45) NULL,
162     CNPJ VARCHAR(25) NULL,
163     Endereco VARCHAR(95) NULL,
164     Telefone VARCHAR(18) NULL,
165     PRIMARY KEY (idFornecedor)
166 )
167 GO
168
169 -----
170 -- Table Pizzaria.Estoques_Fornecedores
171 -----
172 CREATE TABLE Estoques_Fornecedores (
173     idEstoque INT NOT NULL,
174     idFornecedor INT NOT NULL,
175     CONSTRAINT fk_Estoque_has_Fornecedor_Estoque
176         FOREIGN KEY (idEstoque)
177         REFERENCES Estoques (idEstoque)
178         ON DELETE NO ACTION
179         ON UPDATE NO ACTION,
180     CONSTRAINT fk_Estoque_has_Fornecedor_Fornecedor
```



```
181         FOREIGN KEY (idFornecedor)
182         REFERENCES Fornecedores (idFornecedor)
183         ON DELETE NO ACTION
184         ON UPDATE NO ACTION
185     )
186 GO
187
188     -----
189     -- Table Pizzaria.Produtos_Pedidos
190     -----
191 CREATE TABLE Produtos_Pedidos (
192     idProduto INT NOT NULL,
193     idPedido INT NOT NULL,
194     CONSTRAINT fk_Produtos_has_Pedidos_Produtos
195         FOREIGN KEY (idProduto)
196         REFERENCES Produtos (idProduto)
197         ON DELETE NO ACTION
198         ON UPDATE NO ACTION,
199     CONSTRAINT fk_Produtos_has_Pedidos_Pedidos
200         FOREIGN KEY (idPedido)
201         REFERENCES Pedidos (idPedido)
202         ON DELETE NO ACTION
203         ON UPDATE NO ACTION
204 )
205 GO
206
207     -----
208     -- Table Pizzaria.Admissoes
209     -----
210 CREATE TABLE Admissoes (
211     idAdmissao INT NOT NULL,
212     DataAdmissao DATE NULL,
213     DataDemissao DATE NULL,
214     PRIMARY KEY (idAdmissao)
215 )
216 GO
217
218     -----
219     -- Table Pizzaria.Funcionarios_Admissoes
220     -----
221 CREATE TABLE Funcionarios_Admissoes (
222     CPF VARCHAR(11) NOT NULL,
223     idAdmissão INT NOT NULL,
224     CONSTRAINT fk_Funcionarios_has_Admissão_Funcionarios
225         FOREIGN KEY (CPF)
226         REFERENCES Funcionarios (CPF)
227         ON DELETE NO ACTION
```

```
228         ON UPDATE NO ACTION ,
229     CONSTRAINT fk_Funcionarios_has_Admissão_Admissão
230         FOREIGN KEY (idAdmissão)
231         REFERENCES Admissoes (idAdmissao)
232         ON DELETE NO ACTION
233         ON UPDATE NO ACTION
234     )
235 GO
236
237 -----
238 -- Table Pizzaria.Logs
239 -----
240 CREATE TABLE Logs (
241     idLog INT NOT NULL ,
242     DescAtividade VARCHAR(200) NULL ,
243     DataHora DATETIME NULL ,
244     CPF VARCHAR(11) NOT NULL ,
245     PRIMARY KEY (idLog),
246     CONSTRAINT fk_Log_Funcionarios
247         FOREIGN KEY (CPF)
248         REFERENCES Funcionarios (CPF)
249         ON DELETE NO ACTION
250         ON UPDATE NO ACTION
251 )
252 GO
```

6 EXECUÇÃO E TESTES

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

6.1 CONSULTAS

A consulta a seguir foi realizada utilizando as tabelas Estoques e Fornecedores e o resultado pode ser visualizado na figura 10

```
1  -- -----  
2  -- Lista alimentos e seus fornecedores  
3  -- -----  
4  SELECT  Estoques.Produto as [Alimento],  
5          Fornecedores.Nome as [Fornecedor]  
6  FROM    Estoques_Fornecedores  
7          INNER JOIN Estoques ON  
8              Estoques.idEstoque = Estoques_Fornecedores.idEstoque  
9          INNER JOIN Fornecedores ON  
10             Fornecedores.idFornecedor = Estoques_Fornecedores.idFornecedor  
11 ORDER BY Fornecedores.Nome, Estoques.Produto  
12 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	Manjericão	Boa Massa
16	Oregano	Boa Massa
17	Palmito	Boa Massa
18	Queijo Mussar...	Boa Massa
19	Queijo pamesão	Boa Massa
20	Tomate	Boa Massa

Figura 10 – Resultado do select Lista alimentos e seus fornecedores

A consulta a seguir foi realizada utilizando as tabelas Produtos e Estoques e o resultado pode ser visualizado na figura 11

```

1  -- -----
2  -- Lista os nomes dos produtos, seus ingredientes e a
3  -- quantidade em estoque
4  -- -----
5  SELECT  Produtos.Nome,
6          Estoques.Produto,
7          Estoques.Quantidade
8  FROM    Ingredientes
9          INNER JOIN Produtos ON
10         Produtos.idProduto = Ingredientes.idProduto
11         INNER JOIN Estoques ON
12         Estoques.idEstoque = Ingredientes.idEstoque
13  ORDER BY Produtos.Nome, Estoques.Produto
14  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 pamesão	13

Figura 11 – Resultado do select

A consulta a seguir foi realizada utilizando as tabelas Logins e Clientes e o resultado pode ser visualizado na figura 12.

```

1  -- -----
2  -- Lista os clientes e os logins de quem o tiver.
3  -- -----
4  CREATE VIEW ClientesComLogin
5  AS
6      SELECT  Logins.Usuario,
7              Clientes.idCliente FROM Logins
8              RIGHT JOIN Clientes ON
9              Logins.idLogin = Clientes.idLogin
10 GO

```

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

Figura 12 – Resultado do select lista os clientes e os logins de quem o tiver.

A consulta a seguir foi realizada utilizando as tabelas Produtos e Pedidos e o resultado pode ser visualizado na figura 13.

```
1  -----
2  -- Lista produtos pedidos
3  -----
4  CREATE VIEW PedidosRealizados
5  AS
6      SELECT  Produtos.Nome AS [Produto],
7              Pedidos.idCliente
8      FROM    Produtos_Pedidos
9      INNER JOIN Produtos ON
10             Produtos.idProduto = Produtos_Pedidos.idProduto
11      INNER JOIN Pedidos ON
12             Pedidos.idPedido = Produtos_Pedidos.idPedido
13  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 lista produtos pedidos.

A consulta a seguir foi realizada utilizando as views ClientesComLogin e Pedidos-Realizados e o resultado pode ser visualizado na figura 14.

```

1  -- -----
2  -- Lista dos clientes que fizeram pedidos.
3  -- -----
4  CREATE VIEW ClientesQueFizeramPedidos
5  AS
6  SELECT  ClientesComLogin.Usuario,
7          PedidosRealizados.Produto
8          FROM PedidosRealizados
9          INNER JOIN ClientesComLogin ON
10             ClientesComLogin.idCliente = PedidosRealizados.idCliente
11 GO
12
13
14 SELECT  ClientesQueFizeramPedidos.Usuario,
15          COUNT(*) AS [Quantidade de Pedidos]
16          FROM ClientesQueFizeramPedidos
17          GROUP BY ClientesQueFizeramPedidos.Usuario

```

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

Figura 14 – Resultado do select lista dos clientes que fizeram pedidos.

A consulta a seguir foi realizada utilizando a view ClientesComLogin e a tabela Dependentes e o resultado pode ser visualizado na figura 15.

```
1  -----
2  -- Clientes e seus dependentes
3  -----
4  SELECT  ClientesComLogin.Usuario ,
5          Dependentes.Nome [Nome do dependente]
6  FROM    Dependentes
7          INNER JOIN ClientesComLogin ON
8          ClientesComLogin.idCliente = Dependentes.idCliente
```

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

Figura 15 – Resultado do select clientes e seus dependentes

A consulta a seguir foi realizada utilizando as tabelas Funcionários e Cargos e o resultado pode ser visualizado na figura 16.

```
1  -----
2  -- Funcionários e Cargos
3  -----
4  SELECT  Funcionarios.Nome ,
5          Funcionarios.CPF ,
6          Cargos.NomeCargo ,
7          Cargos.Salario
8  FROM    Funcionarios
9          INNER JOIN Cargos ON
10         Cargos.idCargo = Funcionarios.idCargo
11 ORDER BY Cargos.NomeCargo , Funcionarios.Nome
12 GO
```

	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 Funcionários e Cargos

A consulta a seguir foi realizada utilizando as tabelas Funcionários e Cargos e o resultado pode ser visualizado na figura 17.

```

1
2  -----
3  -- Funcionários, cargos e suas admissões
4  -----
5  SELECT  Funcionarios.Nome ,
6          Admissoes.DataAdmissao ,
7          Cargos.NomeCargo ,
8          Cargos.Salario
9  FROM    Funcionarios_Admissoes
10         INNER JOIN Funcionarios ON
11             Funcionarios.CPF = Funcionarios_Admissoes.CPF
12         INNER JOIN Admissoes
13             ON Admissoes.idAdmissao = Funcionarios_Admissoes.idAdmissão
14         INNER JOIN Cargos
15             ON Cargos.idCargo = Funcionarios.idCargo
16 GO

```

	Nome	DataAdmissao	NomeCargo	Salario
1	José Benedito	2005-08-30	Pizzaiolo	2000.00
2	Catarina 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 funcionários, cargos e suas admissões

6.2 PROCEDIMENTOS ARMAZENADOS

Procedimento armazenado para cálculo do aniversários de cada funcionário na empresa.

```

1 CREATE PROCEDURE usp_idadeFuncionarios
2 AS
3     SELECT  Nome ,
4             DATEDIFF(YEAR, DataNascimento, GETDATE()) - CASE
5             WHEN GETDATE() < DATEADD(YEAR,
6             DATEDIFF(YEAR, DataNascimento,
7             GETDATE()),
8             DataNascimento)
9             THEN 1
10            ELSE 0
11            END AS 'Idade',
12            CONVERT(VARCHAR(10),
13            DataNascimento, 103) As 'Data de Nascimento'
14    FROM Funcionarios
15 GO
16
17 EXEC usp_idadeFuncionarios
18 GO

```

Depois de executado, o resultado obtido pode ser visualizado na Figura 18.

	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 – Procedimento Armazenado para calcular idade.

```

1 CREATE PROCEDURE usp_pedidosRealizados
2     @nome VARCHAR(45)
3 AS
4     SELECT F.Nome,
5            C.NomeCargo as 'Cargo',
6            Prod.Nome,
7            CONVERT(VARCHAR(10),P.data, 103) As 'Data do Pedido'
8     FROM Funcionarios F
9         INNER JOIN Cargos C ON
10            C.idCargo = F.idCargo
11         INNER JOIN Pedidos P ON
12            P.CPF = F.CPF
13         INNER JOIN Produtos_Pedidos PP ON
14            PP.idPedido = P.idPedido
15         INNER JOIN Produtos Prod ON
16            Prod.idProduto = PP.idProduto
17     WHERE F.Nome = @nome
18 GO

```

Depois de executado, o resultado obtido pode ser visualizado na Figura 19.

	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 procedimento armazenado que retorna os pedidos realizados.

```

1 CREATE PROCEDURE usp_pedidosRealizadosCliente
2     @nome VARCHAR(45)
3 AS
4     SELECT  Cli.Nome ,
5             Prod.Nome ,
6             CONVERT(VARCHAR(10) ,
7                 P.data, 103) As 'Data do Pedido'
8     FROM    Clientes Cli
9            INNER JOIN Pedidos P ON
10                P.idCliente = Cli.idCliente
11            INNER JOIN Produtos_Pedidos PP
12                ON PP.idPedido = P.idPedido
13            INNER JOIN Produtos Prod
14                ON Prod.idProduto = PP.idProduto
15     WHERE   Cli.Nome = @nome
16 GO

```

Depois de executado, o resultado obtido pode ser visualizado na Figura 20 e também na Figura ??.

	Nome	Nome	Data do Pedido
1	Robervaldo	Calabresa	01/12/2013
2	Robervaldo	Frango C/ Catupiry	01/12/2013
3	Robervaldo	Lombo	01/12/2013

Figura 20 – Procedimento Armazenado que retorna os pedidos de um determinado cliente via parâmetro do nome.

	Nome	Nome	Data do Pedido
1	Valdomiro	Margarita	30/11/2013
2	Valdomiro	Portuguesa	30/11/2013
3	Valdomiro	Nordestina	30/11/2013

Figura 21 – Procedimento Armazenado que retorna os pedidos de um determinado cliente via parâmetro do nome.

CONSIDERAÇÕES FINAIS

Apesar de parecer simples, criar um banco de dados para uma pizzaria 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.