

Assunto: Gerenciamento de uma Lanchonete

Grupo: Kevyn Swahnts dos Santos Ribeiro - 201465576A

Lucas Diniz da Costa - 201465524AC

1. INTRODUÇÃO

Quando pensamos em um sistema que tem o controle das vendas, pensamos em supermercados ou em lojas de departamento, este trabalho consiste em modelar e criar um sistema para o gerenciamento de vendas com foco em lanchonetes e permitir que consigam ter acesso aos dados de cada venda registrada para futuras análises e/ou consultas.

2. REQUISITOS DO PROJETO

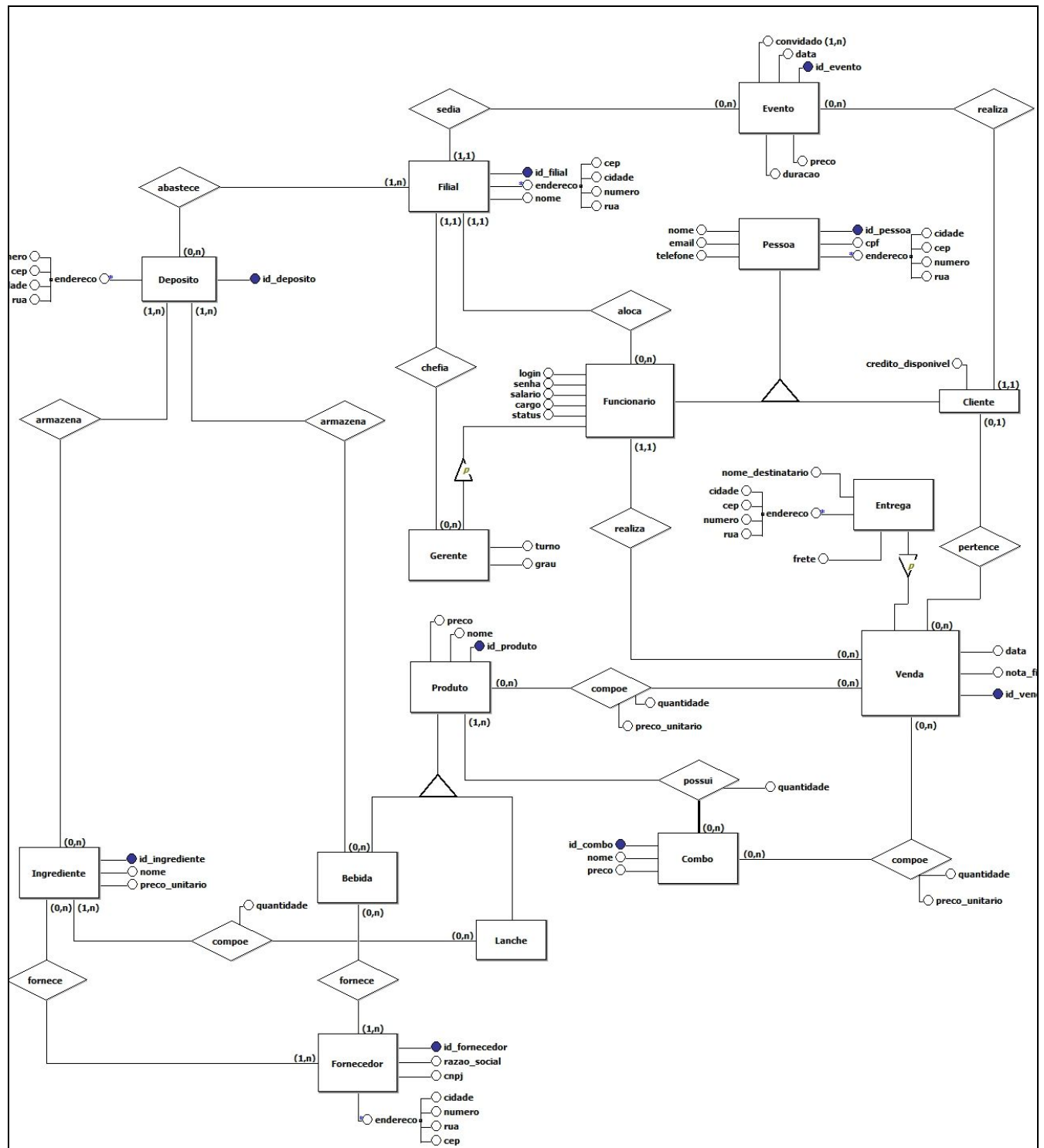
Requisitos do Sistema
<ol style="list-style-type: none">1. O sistema deve gerenciar o cadastro de filiais.2. O sistema deve gerenciar o cadastro de clientes.3. O sistema deve gerenciar o cadastro de funcionários.4. O sistema deve gerenciar o cadastro de depósitos.5. O sistema deve gerenciar o cadastro de fornecedores.6. O sistema deve gerenciar o cadastro de ingredientes.7. O sistema deve gerenciar o cadastro de eventos que serão realizados por clientes em uma filial.8. O sistema deve gerenciar o cadastro dos produtos a serem vendidos.9. O sistema deve gerenciar o cadastro de combos de produtos.10. O sistema deve gerenciar o cadastro de vendas voltadas para a entrega.11. O sistema deve permitir a criação de contas de acesso para os funcionários da empresa.12. O sistema deve permitir que os usuários do sistema realizem login.13. O sistema deve armazenar as vendas realizadas.

Regras de Negócio

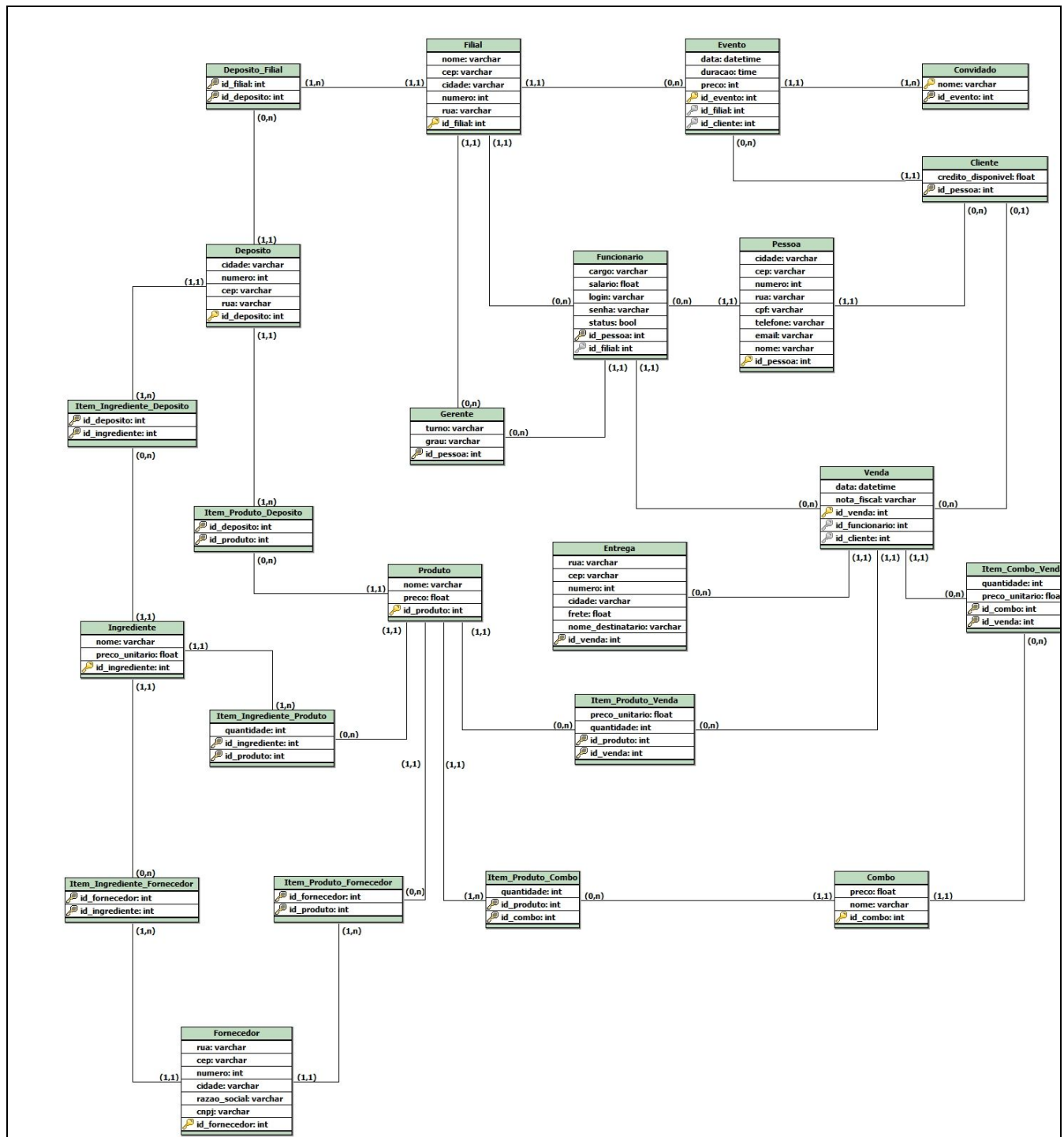
1. Uma filial aloca nenhum ou vários funcionários, e um funcionário é alocado em apenas uma filial.
2. Uma filial é chefiada por nenhum ou vários gerentes, e um gerente chefia apenas uma filial.
3. Uma filial sedia nenhum ou vários eventos, e um evento é necessariamente sediado por apenas uma filial.
4. Uma filial é abastecida por nenhum ou vários depósitos, e um depósito abastece uma ou mais filiais.
5. Um evento é realizado por apenas um cliente, e um cliente pode realizar nenhum ou mais eventos.
6. Uma venda pode possuir nenhum ou um cliente associado a ela, e um cliente pode possuir nenhuma ou várias vendas (compras).
7. Uma venda é composta por nenhum ou mais produtos, e um produto compõe nenhuma ou mais vendas.
8. Uma venda é composta por nenhum ou mais combos, e um combo compõe nenhuma ou mais vendas.
9. Um combo possui um ou mais produtos, e um produto pode pertencer a nenhum ou mais combos
10. Um funcionário realiza nenhuma ou várias vendas, e uma venda é realizada por apenas um funcionário.
11. Um fornecedor fornece nenhuma ou várias bebidas, e uma bebida é fornecida por um ou mais fornecedores.
12. Um fornecedor fornece nenhum ou vários ingredientes, e um ingrediente é fornecido por um ou mais fornecedores.
13. Um ingrediente compõe nenhum ou vários lanches, e um lanche é composto por um ou mais ingredientes.
14. Um depósito armazena nenhum ou vários ingredientes, e um ingrediente é armazenado em um ou mais depósitos.
15. Um depósito armazena nenhuma bebida ou mais, e uma bebida é armazenada em um ou mais depósitos.

3. MODELAGEM RELACIONAL

3.1 Diagrama Entidade Relacionamento



3.2 Modelagem Lógica



4 CONSULTAS EM ÁLGEBRA RELACIONAL

Consulta 1: Liste todos os dados das filiais cadastradas no sistema

$$\Pi_{\{id_filial, nome, cep, cidade, rua, numero\}} Filial$$

Consulta 2: Liste o nome e preço de todos ingredientes presentes no sistema

$$\Pi_{\{nome, preco_unitario\}} Ingrediente$$

Consulta 3: Liste o nome de todos os funcionários presentes no sistema

$$\Pi_{\{nome\}} (\sigma_{Funcionario.id_pessoa = Pessoa.id_pessoa} (Funcionario \otimes Pessoa))$$

Consulta 4: Liste o nome, cargo e salário dos funcionários ativo nas filiais de Juiz de Fora

$$\begin{aligned} getFunc &\leftarrow \sigma_{Funcionario.id_pessoa = Pessoa.id_pessoa} (Funcionario \otimes Pessoa) \\ getFuncFili &\leftarrow \sigma_{getFunc.id_filial = Filial.id_filial \text{ AND } Filial.cidade = "Juiz de Fora" \text{ AND } \\ &\quad getFunc.status = "Ativo"} (getFunc \otimes Filial) \\ Resultado &\leftarrow \Pi_{\{Pessoa.nome, Funcionario.cargo, Funcionario.salario\}} getFuncFili \end{aligned}$$

Consulta 5: Liste os produtos que a filial de nome “A” tem acesso

$Filial.id_filial = Deposito_Filial.id_filial \text{ AND } Filial.nome = "A"$

$DF \leftarrow (Filial \bowtie Deposito_Filial)$

$DFA \leftarrow \Pi_{\{id_deposito\}}(DF)$

$IPD \leftarrow Item_Produto_Deposito$

$DFA.id_deposito = Item_Produto_Deposito.id_deposito$

$P1 \leftarrow \Pi_{\{IPD.id_produto, DFA.id_deposito\}}(DFA \bowtie IPD)$

$P2 \leftarrow Produto$

$\Pi_{\{P2.nome, P2.preco, P1.id_deposito\}}(\sigma_{P1.id_produto = P2.id_produto} (P1 \otimes P2))$

Consulta 6: Liste todos os dados de Clientes cadastrados no sistema

$aux \leftarrow (\sigma_{Cliente.id_pessoa = Pessoa.id_pessoa}(Cliente \otimes Pessoa))$

$resultado \leftarrow \Pi_{\{nome, cpf, email, telefone, cep, cidade, rua, numero, credito_disponivel\}}aux$

Consulta 7: Liste o nome, cpf, salario de todos os funcionários que estão com status igual a 1

$funcPessoa \leftarrow (Funcionario \otimes Pessoa)$

$func \leftarrow (\sigma_{Funcionario.id_pessoa = Pessoa.id_pessoa \text{ AND } Funcionario.status=1}funcPessoa)$

$resultado \leftarrow \Pi_{\{nome, cpf, salario\}}func$

Consulta 8: Liste o nome, cpf, telefone dos gerentes das filiais de Juiz de Fora

$F \leftarrow Funcionario$

$G \leftarrow Gerente$

$func \leftarrow \Pi_{\{id_pessoa, id_filial\}}(\sigma_{G.id_pessoa = F.id_pessoa} (G \otimes F))$

$gerente \leftarrow (\sigma_{Pessoa.id_pessoa = func.id_pessoa} (Pessoa \otimes func))$

$fil \leftarrow (\sigma_{Filial.cidade = "Juiz de Fora"} Filial)$

$gerFil \leftarrow (\sigma_{gerente.id_filial = fil.id_filial} (gerente \otimes fil))$

$resultado \leftarrow \Pi_{\{Pessoa.nome, Pessoa.cpf, Pessoa.telefone\}} gerFil$

Consulta 9: Liste os eventos sediados na filial de nome "A" com nome e cpf do cliente responsável

$E \leftarrow Evento$

$FA \leftarrow \Pi_{\{id_filial, nome\}}(\sigma_{nome = "A"} Filial)$

$EFA \leftarrow (\sigma_{E.id_filial = FA.id_filial} (E \otimes FA))$

$C \leftarrow \Pi_{\{id_pessoa, nome, cpf\}}(\sigma_{Cliente.id_pessoa = Pessoa.id_pessoa} (Cliente \otimes Pessoa))$

$CEFA \leftarrow (\sigma_{EFA.id_cliente = C.id_pessoa} (EFA \otimes C))$

$resultado \leftarrow \Pi_{\{C.nome, C.cpf, E.preco, E.data, E.duracao\}} CEFA$

Consulta 10: Liste os eventos sediados nas filiais de Juiz de Fora com nome e cpf do cliente responsável

$$\begin{aligned}
 F &\leftarrow \Pi_{\{id_filial, nome, cidade\}}(\sigma_{Filial.cidade = "Juiz\ de\ Fora"}(Filial)) \\
 E &\leftarrow (\sigma_{Evento.id_filial = F.id_filial}(Evento \otimes F)) \\
 C &\leftarrow \Pi_{\{id_pessoa, nome, cpf\}}(\sigma_{Cliente.id_pessoa = Pessoa.id_pessoa}(Cliente \otimes Pessoa)) \\
 CE &\leftarrow (\sigma_{E.id_cliente = C.id_pessoa}(E \otimes C)) \\
 resultado &\leftarrow \Pi_{\{Filial.nome, C.nome, C.cpf, E.preco, E.data, E.duracao, Filial.cidade\}}CE
 \end{aligned}$$

Consulta 11: Liste o nome, cpf e nome da filial de todos os gerentes do sistema

$$\begin{aligned}
 F &\leftarrow Funcionario \\
 G &\leftarrow Gerente \\
 GF &\leftarrow (\sigma_{G.id_pessoa = F.id_pessoa}(G \otimes F)) \\
 dadosGer &\leftarrow (\sigma_{GF.id_pessoa = Pessoa.id_pessoa}(GF \otimes Pessoa)) \\
 projGer &\leftarrow \Pi_{\{id_filial, nome, cpf\}}dadosGer \\
 gerFilial &\leftarrow (\sigma_{projGer.id_filial = Filial.id_filial}(projGer \otimes Filial)) \\
 resultado &\leftarrow \Pi_{\{Pessoa.nome, Pessoa.cpf, Filial.nome\}}gerFilial
 \end{aligned}$$

Consulta 12: Liste os funcionários da filial “A” e da filial “B” [UNIÃO]

$$\begin{aligned} F &\leftarrow Funcionario \\ FA &\leftarrow \Pi_{\{id_filial, nome\}}(\sigma_{Filial.nome = "A"}(Filial)) \\ FB &\leftarrow \Pi_{\{id_filial, nome\}}(\sigma_{Filial.nome = "B"}(Filial)) \\ &F.id_filial = FA.id_filial \\ FFA &\leftarrow \Pi_{\{F.id_pessoa, FA.nome\}}(F \sqcap FA) \\ &F.id_filial = FB.id_filial \\ FFB &\leftarrow \Pi_{\{F.id_pessoa, FB.nome\}}(F \sqcap FB) \\ FFAB &\leftarrow FFA \cup FFB \\ f &\leftarrow \sigma_{Pessoa.id_pessoa = FFAB.id_pessoa}(Pessoa \otimes FFAB) \\ resultado &\leftarrow \Pi_{\{Pessoa.id_pessoa, Pessoa.nome, Pessoa.cpf, Filial.nome\}}f \end{aligned}$$

Consulta 13: Liste o nome e cpf de todos os funcionários que também são clientes da lanchonete [INTERSEÇÃO]

$$\begin{aligned} C &\leftarrow \Pi_{\{id_pessoa\}}Cliente \\ F &\leftarrow \Pi_{\{id_pessoa\}}Funcionario \\ FC &\leftarrow \Pi_{\{id_pessoa\}}(F \cap C) \\ FCP &\leftarrow \sigma_{Pessoa.id_pessoa = FC.id_pessoa}(Pessoa \otimes FC) \\ resultado &\leftarrow \Pi_{\{nome, cpf\}}FCP \end{aligned}$$

Consulta 14: Liste o nome e cpf de todos os funcionários que não são clientes da lanchonete [DIFERENÇA]

$$\begin{aligned}C &\leftarrow \Pi_{\{id_pessoa\}} Cliente \\F &\leftarrow \Pi_{\{id_pessoa\}} Funcionario \\f &\leftarrow \Pi_{\{id_pessoa\}} (F - C) \\dados &\leftarrow \sigma_{Pessoa.id_pessoa = f.id_pessoa} (Pessoa \otimes f) \\resultado &\leftarrow \Pi_{\{nome, cpf\}} dados\end{aligned}$$

Consulta 15: Liste todos os funcionários que não são gerentes [DIFERENÇA]

$$\begin{aligned}F &\leftarrow \Pi_{\{id_pessoa\}} Funcionario \\G &\leftarrow \Pi_{\{id_pessoa\}} Gerente \\funNaoGer &\leftarrow \Pi_{\{id_pessoa\}} (F - G) \\dados &\leftarrow \sigma_{Pessoa.id_pessoa = funNaoGer.id_pessoa} (Pessoa \otimes funNaoGer) \\resultado &\leftarrow \Pi_{\{id_pessoa, nome, cpf\}} dados\end{aligned}$$

**Consulta 16: Liste todos os gerentes que também são clientes da lanchonete
[DIVISÃO]**

$$\begin{aligned}
 G &\leftarrow Gerente \\
 P &\leftarrow Pessoa \\
 C &\leftarrow \Pi_{\{id_pessoa\}} Cliente \\
 dados &\leftarrow \Pi_{\{id_pessoa, nome, cpf\}} (\sigma_{P.id_pessoa = G.id_pessoa} (P \otimes G)) \\
 GC &\leftarrow (dados \div C) \\
 resultado &\leftarrow \Pi_{\{nome, cpf\}} GC
 \end{aligned}$$

Consulta 17: Listar nome e cpf dos funcionários que não possuem o cadastro de cliente e nome e cpf de clientes que não possuem cadastro de funcionário [UNIÃO EXCLUSIVA]

$$\begin{aligned}
 F &\leftarrow \Pi_{\{id_pessoa\}} Funcionario \\
 C &\leftarrow \Pi_{\{id_pessoa\}} Cliente \\
 FC &\leftarrow \Pi_{\{id_pessoa\}} (F \cup C) \\
 FCP &\leftarrow \sigma_{Pessoa.id_pessoa = FC.id_pessoa} (Pessoa \otimes FC) \\
 resultado &\leftarrow \Pi_{\{nome, cpf\}} FCP
 \end{aligned}$$

5 MODELO FÍSICO (Parte 1) E CARGA DE DADOS

5.1 Tabelas

O arquivo de criação das tabelas está no arquivo “SQL/MySQL/lanchonete.sql”

5.2 Verificação

O arquivo para fazer uma carga inicial de dados se encontra em “SQL/MySQL/carga_inicial.sql” e o arquivo para realizar a verificação é o “SQL/MySQL/valida_ddl.sql”

5.3 Carga de Dados

O arquivo para carga de dados é o “SQL/MySQL/carga_dados.sql”

6. CONSULTAS EM SQL

Consulta 1: Liste o identificador e nome de todas as filiais cadastradas no sistema [SELECT - FROM]

```
SELECT id_filial, nome FROM filial
```

Consulta 2: Liste o identificador e nome de todos os produtos cadastrados no sistema em ordem alfabética pelo nome [ORDER BY ASC]

```
SELECT id_produto, nome FROM produto ORDER BY nome ASC
```

Consulta 3: Liste todos os ingredientes do sistema em ordem decrescente do preço unitário [ORDER BY DESC]

```
SELECT * FROM ingrediente ORDER BY preco_unitario DESC
```

Consulta 4: Liste todas as filiais localizadas em Juiz de Fora [WHERE]

```
SELECT * FROM filial WHERE cidade='Juiz de Fora'
```

Consulta 5: Liste o nome dos convidados que comecem com a letra C do evento de identificador 1 [LIKE]

```
SELECT c.nome FROM convidado c WHERE nome LIKE 'C%' AND  
c.id_evento=1
```

Consulta 6: Liste o nome e identificador de todos os Clientes de Belo Horizonte [JOIN]

```
SELECT p.nome, p.id_pessoa FROM cliente c JOIN pessoa p ON  
c.id_pessoa=p.id_pessoa WHERE cidade='Belo Horizonte'
```

Consulta 7: Liste o nome de todos os funcionários da filial de identificador 12 ordenados pelo nome [INNER JOIN]

```
SELECT p.nome FROM pessoa p INNER JOIN funcionario fu ON  
p.id_pessoa=fu.id_pessoa INNER JOIN filial fi ON  
fu.id_filial=fi.id_filial WHERE fi.id_filial=12 ORDER BY  
p.nome
```

Consulta 8: Liste o nome do produto e a quantidade de ingredientes que ele possui, apenas dos produtos com menos de 5 ingredientes e indique aqueles que possuem que 5 ingredientes ou mais [LEFT JOIN]

```
SELECT p.nome, i.quantidade FROM produto p LEFT JOIN  
item_ingredientes_produto i ON p.id_produto=i.id_produto AND  
i.quantidade<5
```

Consulta 9: Liste o nome do combo e a quantidade de produtos que ele possui, apenas dos combos com mais de 5 produtos e indique aqueles que possuem 5 produtos ou menos [RIGHT JOIN]

```
SELECT c.nome, i.quantidade FROM item_produto_combo i RIGHT JOIN  
combo c ON i.id_combo=c.id_combo AND i.quantidade>5
```

Consulta 10: O número total de vendas cadastradas no sistema [COUNT]

```
SELECT COUNT(*) AS "Total Vendas" FROM venda
```

Consulta 11: Média salarial dos funcionários da filial de identificação 13 [AVG]

```
SELECT AVG(f.salario) AS "Média Salarial" FROM funcionario f WHERE  
f.id_filial=13
```

Consulta 12: O salário do gerente com o maior salário [MAX]

```
SELECT MAX(f.salario) AS "Maior Salário" FROM gerente g INNER JOIN  
funcionario f on g.id_pessoa=f.id_pessoa
```

Consulta 13: O menor crédito disponível dentre os cliente [MIN]

```
SELECT MIN(c.credito_disponivel) as "Menor Crédito Disponível" FROM  
cliente c
```

Consulta 14: O valor total recebido por todos os eventos [SUM]

```
SELECT SUM(e.preco) AS "Valor Total Eventos" FROM evento e
```

**Consulta 15: Liste o nome de todos os clientes com nomes diferentes
[SELECT DISTINCT]**

```
SELECT DISTINCT p.nome FROM cliente c INNER JOIN pessoa p ON  
c.id_pessoa=p.id_pessoa ORDER BY p.nome
```

Consulta 16: Número total de vendas realizadas por ano [GROUP BY]

```
SELECT COUNT(*) AS "Total Vendas", EXTRACT(YEAR FROM v.data) AS  
"ANO" FROM venda v GROUP BY EXTRACT(YEAR FROM v.data)
```

Consulta 17: Depósitos que possuem pelo menos 1 ingrediente [HAVING]

```
SELECT COUNT(*) AS "Número Ingredientes", d.id_deposito AS  
       "Depósito" FROM deposito d INNER JOIN  
       item_ingrediente_deposito i ON d.id_deposito=i.id_deposito  
       GROUP BY d.id_deposito HAVING COUNT(*) > 1
```

Consulta 18: Liste o nome dos produtos que não são vendidos em combo [IN]

```
SELECT p.nome, id_produto FROM produto p WHERE p.id_produto NOT IN  
       (SELECT i.id_produto FROM item_produto_combo i)
```

Consulta 19: Liste o identificador das vendas que foram realizadas apenas com combos [EXISTS]

```
SELECT v.id_venda FROM venda v WHERE EXISTS (SELECT * FROM  
       item_combo_venda ic WHERE ic.id_venda=v.id_venda) AND NOT  
       EXISTS (SELECT * FROM item_produto_venda ip WHERE  
       ip.id_venda=v.id_venda) ORDER BY v.id_venda
```

Consulta 20: Liste o nome de todos os funcionários e caso seja um gerente, apresente o seu grau de gerência [LEFT OUTER JOIN]

```
SELECT p.nome AS "Funcionário", g.grau AS "Grau Gerência" FROM  
       funcionario f LEFT OUTER JOIN gerente g ON  
       f.id_pessoa=g.id_pessoa INNER JOIN pessoa p ON  
       p.id_pessoa=f.id_pessoa ORDER BY p.nome
```

Consulta 21: Liste o identificador de todas as vendas e caso seja uma entrega, apresente o valor do frete [RIGHT OUTER JOIN]

```
SELECT v.id_venda AS "Venda", e.frete AS "Frete" FROM entrega e  
       RIGHT OUTER JOIN venda v ON e.id_venda=v.id_venda ORDER BY  
       v.id_venda
```

Consulta 22: Liste o nome dos clientes de Juiz de Fora com a inicial M [AND]

```
SELECT p.nome FROM pessoa p INNER JOIN cliente c ON
p.id_pessoa=c.id_pessoa WHERE p.cidade="Juiz de Fora" AND
p.nome LIKE "A%"
```

Consulta 23: Liste o nome das filiais de identificador 1 e 2 [OR]

```
SELECT f.id_filial, f.nome FROM filial f WHERE id_filial=1 OR
id_filial=2
```

Consulta 24: Liste a média salarial dos funcionários que não são gerentes com o nome da coluna de "MÉDIA SALARIAL" [AS]

```
SELECT AVG(f.salario) AS "MÉDIA SALARIAL" FROM funcionario f WHERE
f.id_pessoa NOT IN (SELECT g.id_pessoa FROM gerente g)
```

Consulta 25: Liste os eventos que ocorreram entre os anos de 1990 e 2000 [BETWEEN]

```
SELECT * FROM evento e WHERE EXTRACT(YEAR FROM e.data) BETWEEN 1990
AND 2000
```

Consulta 26: Liste todos os nomes dos funcionários do sistema, caso seja apenas um funcionário, imprima seu cargo e caso seja um gerente, imprima gerente [CASE]

```
SELECT p.nome, CASE WHEN f.id_pessoa NOT IN (SELECT g.id_pessoa
FROM gerente g) THEN f.cargo ELSE "Gerente" END AS "Cargo"
FROM pessoa p INNER JOIN funcionario f ON
p.id_pessoa=f.id_pessoa ORDER BY p.nome
```

Consulta 27: Liste os produtos que não foram vendidos ainda [IS NULL]

```
SELECT p.id_produto, p.nome FROM produto p WHERE (SELECT
i.id_produto FROM item_produto_venda i WHERE
i.id_produto=p.id_produto LIMIT 1) IS NULL
```


Consulta 28: Liste os 10 clientes com o crédito disponível [LIMIT]

```
SELECT c.id_pessoa, p.nome, c.credito_disponivel FROM cliente c
      INNER JOIN pessoa p ON c.id_pessoa=p.id_pessoa ORDER BY
      c.credito_disponivel DESC LIMIT 10
```

Consulta 29: Liste o total recebido de frete das entregas sem as casas decimais [ROUND()]

```
SELECT ROUND(SUM(e.frete), 0) FROM entrega e
```

Consulta 30: Liste todos os produtos e combos cadastrados [UNION]

```
SELECT p.id_produto AS "id", p.nome AS "nome" FROM produto p
      UNION
      SELECT c.id_combo, c.nome FROM combo c
      ORDER BY id
```

Consulta 31: Listar os nomes dos clientes que possuem nomes iguais ao de algum funcionário [INTERSECT]

```
SELECT * FROM (
SELECT p.nome as nome_cliente FROM cliente c INNER JOIN pessoa p ON
      c.id_pessoa=p.id_pessoa
      INTERSECT
      SELECT p.nome as nome_funcionario FROM funcionario f INNER JOIN
      pessoa p ON f.id_pessoa=p.id_pessoa
      ) tab ORDER BY tab.nome_cliente
```

Consulta 32: Liste o nome dos clientes que foram atendido pelos funcionários de identificação 67 e 68 [MINUS / EXCEPT] [DIVISÃO]

```
SELECT p.nome, p.id_pessoa FROM cliente c INNER JOIN pessoa p ON
      c.id_pessoa=p.id_pessoa WHERE NOT EXISTS(

      SELECT f.id_pessoa FROM funcionario f WHERE f.id_pessoa=67 OR
      f.id_pessoa=68

      EXCEPT

      SELECT v.id_funcionario FROM venda v WHERE
      c.id_pessoa=v.id_cliente)
```

Os seguintes operadores foram testados no **Postgres**, pois não possuem suporte no **MySQL**:

- *INTERSECT*
- *MINUS (EXCEPT)*

7 MODELO FÍSICO (Parte 2)

7.1 Gatilhos e Funções

O arquivo com os gatilhos e as funções se encontram em
“SQL/Postgres/04_gatilhos_e_funcoes.sql”

7.2 Verificação

O arquivo com as verificações dos gatilhos e das funções se encontra em
“SQL/Postgres/05_verificacoes.sql”

7.3 Visões

Visão 1: Destinado à gerência do sistema e ao setor financeiro

Objetivo: Facilitar a visualização de funcionários por filial e manter um controle dos que saíram e um controle melhor da folha de pagamento de salários

```
CREATE VIEW `funcionarioFiliais` AS

SELECT f.id_pessoa as idPessoa, p.nome as funcNome, f.status as
funcStatus, p.cpf, f.cargo, f.salario, p.telefone, p.email,
fi.id_filial as idFilial, fi.nome as filialNome, fi.cidade as
filialCidade

FROM Funcionario f

INNER JOIN Pessoa p ON (f.id_pessoa = p.id_pessoa)

INNER JOIN Filial fi ON (f.id_filial = fi.id_filial)

ORDER BY fi.id_filial ASC, f.id_pessoa ASC
```

Visão 2: Destinado à gerência do sistema e ao setor financeiro

Objetivo: Facilitar a contabilidade de vendas por filial de modo a se ter um maior controle quanto a lucratividade da empresa

```
CREATE VIEW `vendaFiliais` AS

SELECT fi.id_filial idFilial, fi.nome as nomeFilial, ptv.total as
precoVenda, id_venda, ptv.data as dataVenda

FROM Funcionario f

INNER JOIN Filial fi ON (f.id_filial = fi.id_filial)

INNER JOIN (

    SELECT total_produto + total_combo AS total , id_venda, t.data,
t.id_funcionario

    FROM (

        SELECT IFNULL(SUM(ip.preco_unitario * ip.quantidade), 0) as
total_produto,

            IFNULL(SUM(ic.preco_unitario * ic.quantidade), 0) as
total_combo, v.id_venda, v.data, v.id_funcionario

        FROM venda v

        LEFT JOIN item_produto_venda ip ON ip.id_venda=v.id_venda

        LEFT JOIN item_combo_venda ic ON ic.id_venda=v.id_venda

        GROUP By v.id_venda

    ) t

) as ptv ON (f.id_pessoa = ptv.id_funcionario)

ORDER BY fi.id_filial ASC, id_venda ASC
```

8 OTIMIZAÇÃO DO BANCO

8.1 Índices

Os índices criados estão no arquivo “SQL/Postgres/06_indices”

8.2 Verificação dos índices

O tempo de cada consulta é dado por uma média de 5 execuções consecutivas da mesma.

Consultas para Índice 1:

1. `SELECT * FROM item_combo_venda WHERE quantidade=8;`
2. `SELECT c.nome, i.quantidade FROM item_combo_venda i RIGHT JOIN combo c ON i.id_combo=c.id_combo AND i.quantidade>5;`

Consulta	Sem Índice	Com Índice	Desempenho
1	828ms	479ms	1.73x mais rápido
2	713ms	346ms	2.06x mais rápido

Consultas para Índice 2:

3. `SELECT p.nome FROM pessoa p INNER JOIN cliente c ON p.id_pessoa=c.id_pessoa WHERE p.cidade='Juiz de Fora' AND p.nome LIKE 'A%';`
4. `SELECT p1.nome FROM pessoa p1 WHERE p1.nome NOT IN (SELECT p2.nome FROM pessoa p2 WHERE p2.nome LIKE 'C%');`

Consulta	Sem Índice	Com Índice	Desempenho
3	650ms	461ms	1.41x mais rápido
4	684ms	469ms	1.46x mais rápido

9. TELAS DA APLICAÇÃO

O código foi criado na linguagem PHP utilizando um banco de dados em MySQL e se encontra na pasta “Códigos/”. Alguns exemplos das telas criadas podem ser vistas a seguir.

Menu

Cliente

Depósito

Evento

Filial



































Fornecedor



































Funcionário

Gerente

Filial

+ Adicionar Nova Filial

ID	Nome	Rua	Número	Cidade	CEP	Ações
1	G	Rua L	34	Juiz de Fora	40881-805	 
2	V	Rua K	195	Belo Horizonte	43402-882	 
3	H	Rua D	844	Sao Paulo	27298-750	 
4	T	Rua B	637	Porto Alegre	21897-753	 
5	A	Rua X	550	Rio de Janeiro	64172-248	 
6	I	Rua B	883	Campo Grande	26312-301	 
7	Q	Rua E	86	Juiz de Fora	04927-529	 
8	U	Rua P	967	Campo Grande	61489-480	 
9	L	Rua L	748	Belo Horizonte	36441-473	 
10	V	Rua V	895	Juiz de Fora	41799-070	 
11	J	Rua N	395	Juiz de Fora	29098-398	 
12	R	Rua C	664	Juiz de Fora	88842-891	 
13	S	Rua U	202	Belo Horizonte	17498-920	 
14	F	Rua V	52	Juiz de Fora	37672-002	 
15	T	Rua Y	976	Juiz de Fora	71945-764	 
16	A	Rua N	426	Juiz de Fora	68153-991	 
17	K	Rua T	385	Juiz de Fora	96165-130	 

Menu	Evento									
	<div> <div>+ Adicionar Novos Convidados</div> <div>+ Adicionar Novo Evento</div> </div>									
	Cliente									
	Depósito									
	Evento									
Filial	Fornecedor									
Funcionário	Gerente									
ID	Cliente	Data	Duração	Preço	Filial	Ações	Convidados			
1	Angelo Ribeiro	2006-08-26 17:30:00	04:30:00	5118.11	Y	 	Ana Carvalho Q. Y. ▾			
2	Lucas Ambrosio	2008-11-20 20:45:00	02:45:00	9523.52	T	 	Bruno Carvalho U. M. ▾			
3	Kevyn Silva	2008-03-23 19:45:00	05:00:00	4184.73	W	 	Douglas Santos D. T. ▾			
4	Mariana Santos	2008-03-20 19:00:00	02:45:00	3607.95	J	 	Ana Carvalho W. A. ▾			
5	Carlos Ribeiro	2016-09-11 20:30:00	04:45:00	6534.98	D	 	Carlos Costa F. O. ▾			
6	Pedro Silva	2008-11-24 12:00:00	03:30:00	5678.39	W	 	Ana Carvalho O. C. ▾			
7	Rodrigo Ribeiro	1981-05-12 13:45:00	04:00:00	1313.53	D	 	Andreza Diniz G. F. ▾			
8	Carlos Carvalho	1983-02-22 12:00:00	02:45:00	4277.55	T	 	Ana Costa B. X. ▾			
9	Cecilia Silva	1980-08-04 18:45:00	04:45:00	3565.43	U	 	Andreza Carvalho D. Q. ▾			
10	Franciane Silva	1994-01-03 17:00:00	05:30:00	5160.00	F	 	Angelo Santos F. C. ▾			
11	Mateus Diniz	1999-11-17 14:45:00	05:45:00	7025.02	J	 	Cecilia Carvalho U. I. ▾			
12	Veronica Ambrosio	2017-06-28 12:15:00	04:45:00	3270.87	B	 	Andreza Silva D. X. ▾			
13	Franciane Silva	1981-05-08 12:15:00	03:00:00	4207.96	G	 	Andreza Ribeiro P. C. ▾			
14	Rodrigo Carvalho	2002-10-02 14:15:00	05:30:00	6732.54	S	 	Bruno Santos K. R. ▾			
15	Isis Diniz	2017-09-21 14:30:00	02:45:00	2499.96	W	 	Ana Ambrosio X. K. ▾			
16	Douglas Santos	2017-03-08 16:30:00	04:30:00	414.48	Q	 	Andreza Diniz X. S. ▾			
17	Kevyn Silva	2018-06-14 17:45:00	04:45:00	505.89	L	 	Andreza Carvalho R. Y. ▾			