



**Universidade Federal de São Carlos**

**Campus Sorocaba**

**Ciência da Computação**

**Bancos de Dados**

Profa. Dra. Sahudy Montenegro González

**Empresa de Celular – Grupo 15**

**Integrantes:**

André Alves Baggio Esteves: 823928

Gabriel Henrique Urbano: 824031

Pedro Henrique Pereira Machado: 828632

Fase Final, 16 de Fevereiro de 2025

## Índice

<b>1. Descrição do Problema.....</b>	<b>3</b>
1.1 Consultas.....	4
<b>2. Projeto Conceitual.....</b>	<b>5</b>
<b>3. Modelo/Diagrama Entidade-Relacionamento.....</b>	<b>8</b>
<b>4. Projeto Lógico.....</b>	<b>9</b>
4.1 Detalhes da Conversão para o Modelo Relacional.....	10
<b>5. Projeto Físico do Banco de Dados.....</b>	<b>13</b>
5.1 Criação do Banco de Dados.....	13
5.2 Adoção de Políticas de Restrições de Integridade.....	13
5.3 Alimentação do Banco de Dados.....	19
<b>6. Especificação de Consultas em Álgebra Relacional e na SQL.....</b>	<b>19</b>
<b>7. Triggers e Views.....</b>	<b>23</b>
<b>8. Considerações Finais.....</b>	<b>23</b>
<b>9. Referências.....</b>	<b>24</b>

## **1. Descrição do Problema**

A empresa Orion, especializada na fabricação e comercialização de telefones celulares, possui diversas áreas, como produção, vendas, estoque, e suporte. A fábrica necessita de um software para auxiliar o gerenciamento do dia a dia da empresa.

A companhia precisa de um cadastro para cada cliente, identificando ele pelo CPF, e possuir nome, email, telefones e endereço. Ao realizar um pedido, é gerado um número único identificador da venda, além de registrar o valor da venda feita, a descrição, o cliente, qual vendedor fez a venda, a data da compra, os produtos vendidos e quantos foram. Quanto aos funcionários, são atribuídos seu CPF, que é usado para identificá-los, nome, salário, telefones, data de nascimento e idade, calculado a partir da data de nascimento, todo funcionário deve ser maior de 18 anos. No entanto, ele deve estar cadastrado em algum departamento, seja o estoque, vendas, produção ou suporte. É preciso registrar os turnos específicos(matutino, diurno e vespertino) de cada funcionário do estoque; aos que dão suporte técnico, seus cursos de formação; aos vendedores, suas metas de venda; e aos do setor de produção, suas horas semanais trabalhadas.

Além disso, a empresa é dividida em diferentes departamentos, cada um com um nome, e número de funcionários, e deve possuir pelo menos um funcionário ligado à ele. O departamento de Estoque deve registrar a capacidade máxima do estoque e seu estado (adequado, vazio ou cheio). Com a função de controle de fabricação existe a entidade Peças, onde estarão listadas todas as peças disponíveis (ramificado em tipo de peça e sua quantidade) para construção e/ou reposição de modelos produzidos pela empresa. Os celulares fabricados possuem um número de série, nome do modelo, especificações técnicas (tamanho da memória, duração da bateria, tipo do processador e da câmera), preço de venda e também a quantidade de exemplares disponíveis em estoque.

O setor especializado em assistência técnica dos produtos é composto por funcionários

especializados na área. Já o suporte funciona da seguinte maneira: um cliente pode fazer uma solicitação, será atribuído um código identificador à ela, uma descrição do problema, a data da solicitação, e seu um status (aberto, em andamento, concluída ou cancelada), então um funcionário da área irá resolver o problema do cliente.

## 1.1 Consultas

O sistema projetado deverá ser capaz de realizar as seguintes tarefas e consultas, a partir dos modelos apresentados.

**1 - Quantas vendas foram realizadas por mês e por modelo?**

Responsável: André

**2 - Quais os funcionários que pertencem ao setor de vendas e nunca venderam um celular do modelo Jaspion 3?** Responsável: André

**3 - Qual a média salarial dos funcionários do setor de produção entre 22 e 40 anos?** Responsável: Pedro

**4 - Quais clientes fizeram solicitações de suporte de um certo modelo em um determinado mês?** Responsável: Pedro

**5 - Quais dispositivos que foram vendidos nunca foram levados para suporte?** Responsável: Pedro

**6 - Quais e quantos modelos de celular possuem disponíveis no estoque?** Responsável: Gabriel

**7 - Quais as solicitações de suporte com status "em andamento" ou "aberto", incluindo o funcionário e cliente responsável?** Responsável: Gabriel

## 2. Projeto Conceitual

O projeto conceitual visa exibir toda a estrutura que será desenvolvida no banco de dados com todas as entidades e seus respectivos atributos presentes, como representado na tabela 7.

Tipo-Entidade	Atributo	Tipo	Restrição
Funcionário	idfuncionario	Identificador	Obrigatório
	cpf	Monovalorado	Obrigatório, Único
	nome	Monovalorado	Obrigatório
	telefone	Multivalorado	opcional
	data_nascimento	Monovalorado	Obrigatório, (dd/mm/yyyy) e $\geq 1960$ e $< 2006$
	salario	Monovalorado	Obrigatório, (em R\$/mês) e $\geq 1.518,00$
Suporte	tipo_formacao	Multivalorado	Obrigatório
Vendedor	meta_vendas	Monovalorado	Obrigatório
Produtor	horas_semanais	Monovalorado	Obrigatório

Estoquista	turno	Monovalorado	Obrigatório, (manhã, tarde ou noite)
Clientes	cpf	Identificador	Obrigatório
	nome	Monovalorado	Obrigatório
	email	Monovalorado	Obrigatório
	telefone	Monovalorado	Opcional
	endereço	Composto e Multivalorado	Obrigatório
Departamento	Nome	Identificador	Obrigatório
	num_funcionario	Calculado	Obrigatório, calculado a partir do relacionamento de Funcionario com Departamento
Estoque	capacidade_maxima	Monovalorado	Obrigatório, $\geq 0$
	status	Monovalorado	Obrigatório, (adequado, cheio ou vazio)
Celular	numero_serie	Identificador	Obrigatório
	tipo_modelo	Monovalorado	Obrigatório
Modelo	nome_modelo	Identificador	Obrigatório
	preco	Monovalorado	Obrigatório(em US\$) $>0$
	tamanho_memoria	Monovalorado	Obrigatório(em GB), entre 2 e 256 GB
	tipo_camera	Monovalorado	Obrigatório(em MP), entre 12 e 200 MP

	processador	Monovalorado	Obrigatório
	duracao_bateria	Monovalorado	Obrigatório, em horas e >0
	quantidade	Monovalorado	Obrigatório, Calculado a partir do número de celulares
Peça	tipo_peca	Identificador	Obrigatório,
	quantidade_disponivel	Monovalorado	Obrigatório, >=0
Solicitação	codigo_id	Identificador	Obrigatório
	descricao	Monovalorado	Obrigatório
	status	Monovalorado	Obrigatório(aberto, em andamento, concluído ou cancelado)
	data	Monovalorado	Obrigatório(dd/mm/yyyy)
Pedido	id_pedido	Identificador	Obrigatório
	descricao	Monovalorado	Obrigatório
	valor	Calculado	Obrigatório
	datahora	Monovalorado	Obrigatório

Tabela 7 - Projeto Conceitual

### 3. Modelo/Diagrama Entidade-Relacionamento

O MER surge como uma maneira de representar o mesmo banco do Projeto Conceitual, entretanto com todas as relações e dependências. Produzido a partir do projeto conceitual do apresentado na seção anterior.

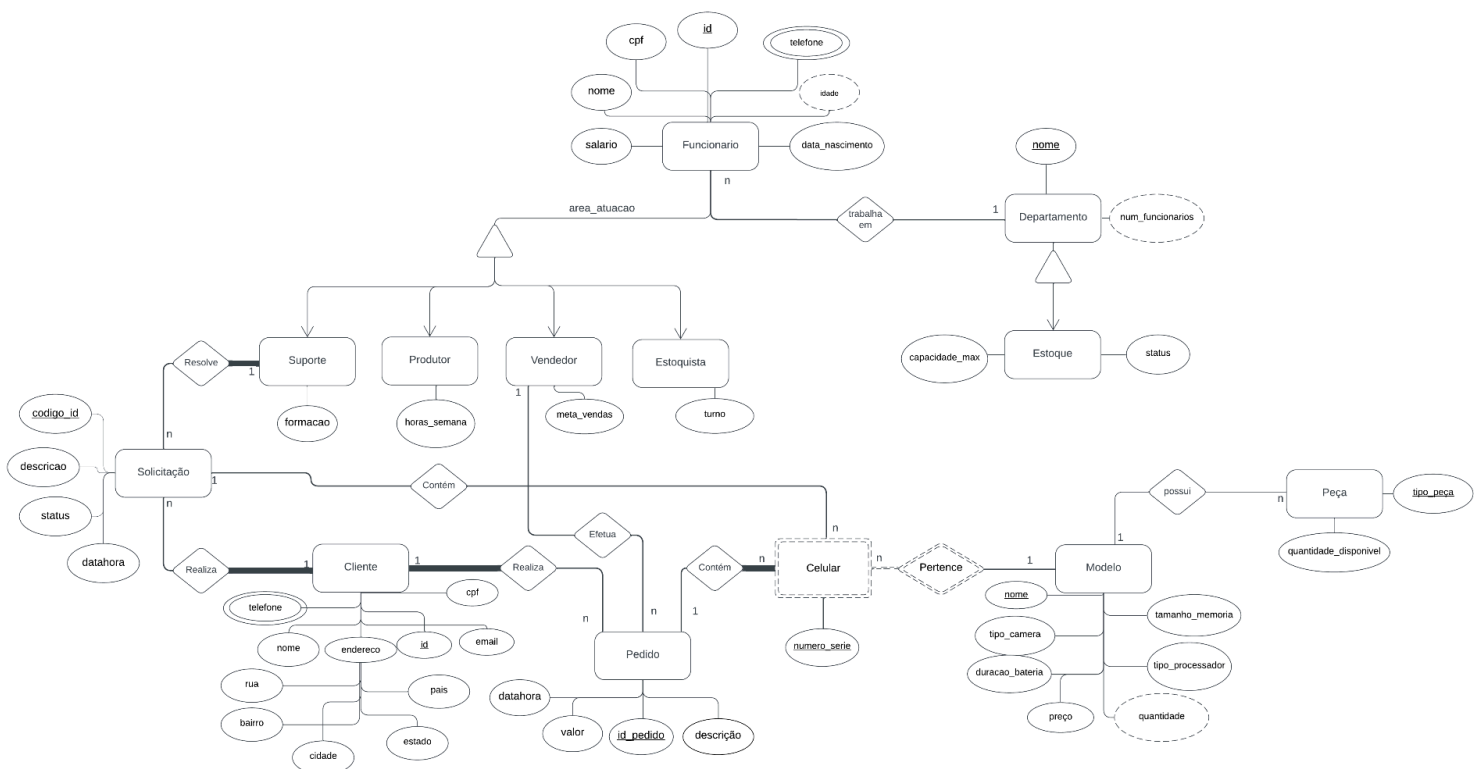


Figura 1 - Modelo/Diagrama Entidade-Relacionamento

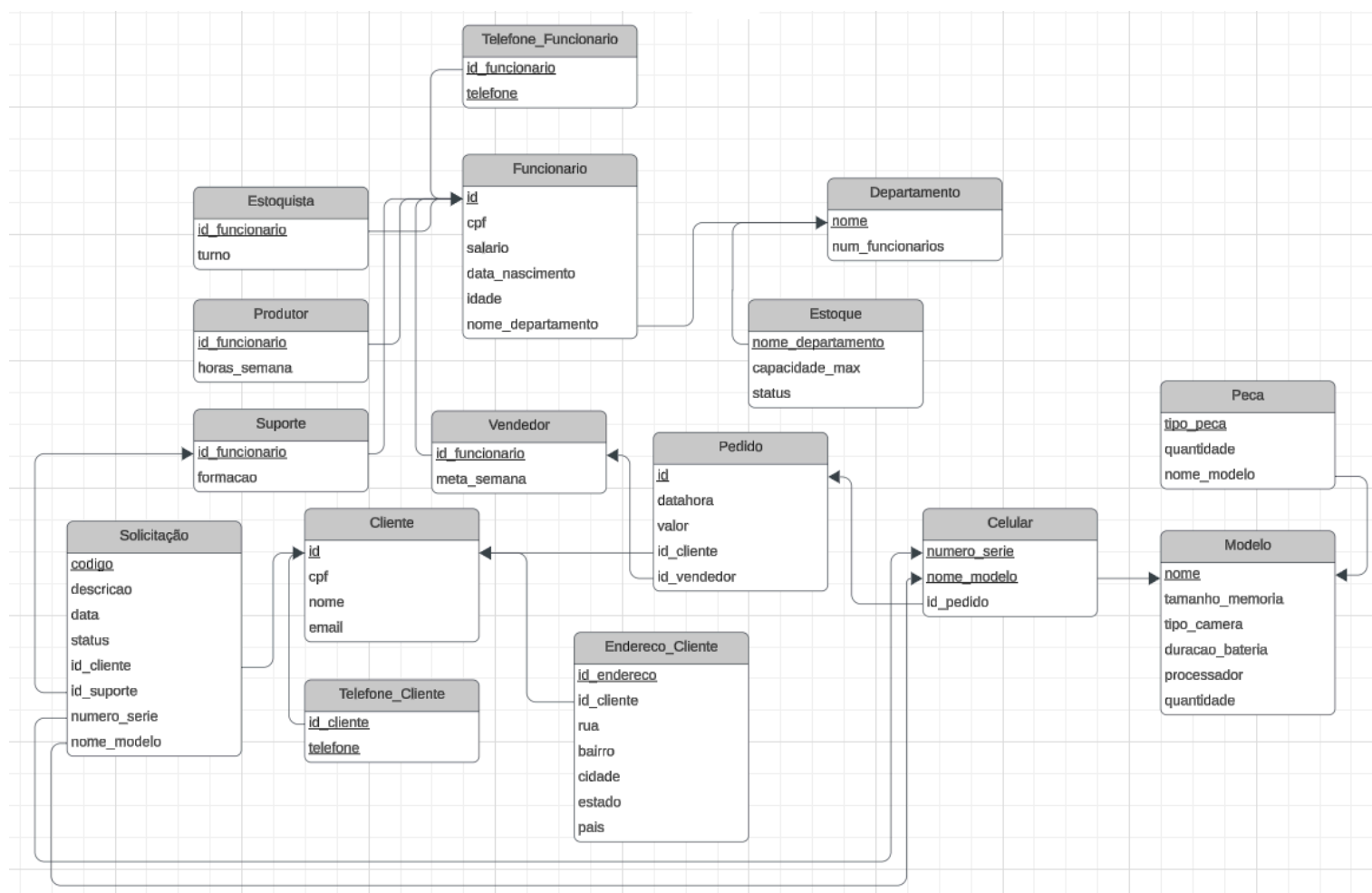


Também é possível visualizá-lo através do link:

[https://lucid.app/lucidchart/b47a4019-0481-40d5-9313-7d33fb7551e1/edit?invitationId=inv\\_ee0cbbc9-c738-48a8-927a-81e3a31a2584](https://lucid.app/lucidchart/b47a4019-0481-40d5-9313-7d33fb7551e1/edit?invitationId=inv_ee0cbbc9-c738-48a8-927a-81e3a31a2584)

#### 4. Projeto Lógico

Nesta seção iremos realizar a conversão do Modelo Entidade Relacionamento para o Modelo Relacional, com auxílio do Projeto Conceitual e do Modelo Entidade Relacionamento, que foi corrigido e colocado na notação vista em sala de aula. Como ferramenta para produzir os diagramas foi usado o software LucidChart: <https://www.lucidchart.com>



*Figura 2 - Modelo desenvolvido baseado em notações vistas em sala de aula*

Também disponível pelo link:

[https://lucid.app/lucidchart/c9dde8b5-6f29-433b-88f8-ecc82f8cc3fe/edit?viewport\\_lo c=5942%2C520%2C2555%2C1063%2C0\\_0&invitationId=inv\\_1b0e61f2-66fe-47d0-b183-29c2a55d8286](https://lucid.app/lucidchart/c9dde8b5-6f29-433b-88f8-ecc82f8cc3fe/edit?viewport_lo c=5942%2C520%2C2555%2C1063%2C0_0&invitationId=inv_1b0e61f2-66fe-47d0-b183-29c2a55d8286)

#### 4.1 Detalhes da Conversão para o Modelo Relacional

##### **Funcionários:**

Foi utilizada esta forma de conversão das especializações, pois assim possuímos menos informações redundantes quanto o possível no Banco de Dados, e também eliminamos atributos opcionais desnecessários. Ao criar tabelas separadas para as especializações, cada uma terá somente seu atributo específico e a referência para funcionário, logo, se for preciso de informações adicionais sobre o funcionário basta utilizar a chave estrangeira `cpf_funcionario`. Contudo, esta alternativa torna necessário o uso de mais junções para recuperar dados, o que pode impactar o desempenho.

##### **Solicitação**

Como cada solicitação será feita por um cliente, e será resolvida por um funcionário. Foram colocadas duas chaves referenciais na tabela Solicitação, `cpf_cliente` que faz referência para Cliente, e `cpf_suporte` que faz referência a `cpf_funcionario` em Suporte. Esta é alternativa é possível pois a cardinalidade de Solicitação é (N:1) Tanto com Suporte quanto para Cliente.

##### **Celular**

Como especificado no MER, Celular é uma entidade fraca de modelo. Deste modo, a chave primária desta entidade deve ser, tanto o nome do modelo que ele

pertence, que é uma chave estrangeira para Modelo, quanto o número de série do celular, atributo da própria entidade Fraca.

### **Peça**

Cada peça irá possuir uma chave estrangeira, especificando a qual modelo de celular que ela está associada, já que a cardinalidade de Peça para Modelo é de (N:1) e também a quantidade disponível no estoque da empresa e sua chave primária, tipo\_peca.

### **Atributos Multivalorados**

Para estar de acordo com a Forma Normal 1, e evitar a utilização de atributos opcionais desnecessários. Foi criada uma tabela separada, que possui o telefone, e o cpf da pessoa a qual possui aquele telefone. Logo os dois atributos se tornam a chave primária desta relação. Deste modo é possível associar mais de um telefone, tanto para cliente, quanto para funcionário. Apesar de ser necessário mais junções para resgatar todos os telefones da pessoa, é a melhor alternativa ao invés de criar campos opcionais.

### **Pedido e Celular**

Como cada pedido pode possuir mais de um celular, a cardinalidade deste relacionamento é de (N:N), logo foi criada uma tabela separada para realizar esta conversão de maneira mais adequada. Esta tabela irá possuir a id do pedido, chave estrangeira que faz referência a id em Pedido, e nome\_modelo e numero\_serie chaves estrangeiras que fazem referência a nome\_modelo e numero\_serie em Celular. Logo, esses três atributos irão compor a super chave desta relação.

### **Departamento**

Semelhantemente a como foi convertido a tabela Funcionário, o mesmo foi feito para Departamento, pois desta maneira não é necessário utilizar atributos opcionais, apesar de ser necessário a criação de mais uma tabela, a tabela de cada especialização possui somente seus dados. No entanto, a principal desvantagem deste método é o custo de junção para recuperar dados das especializações.

Relação	Dependências Funcionais
Funcionario	cpf -> salario, data_nascimento, idade, nome_departamento
Departamento	nome -> num_funcionarios
Estoque	nome-> capacidade_maxima, status
Vendas	nome-> faturamento_dia, faturamento_semana, faturamento_mes, faturamento_ano
Modelo	nome-> preco, quantidade, tipo_camera, duracao_bateria, processador, tamanho_memoria
Peca	tipo_peca-> quantidade, nome_modelo
Pedido	id_pedido-> valor, data, hora, cpf_cliente, cpf_vendedor
Cliente	cpf-> nome, endereco, email
Solicitacao	codigo -> data, status, descricao, cpf_cliente, cpf_suporte

Tabela 8 - Dependências Funcionais

### **Forma Normal**

Levando em consideração que todos os atributos devem estar de acordo com:

A partir da conversão do Modelo Entidade Relacionamento para o Modelo Relacional, ambos apresentados anteriormente, é possível concluir que o modelo já está na terceira forma normal, pois, todos os atributos são valores atômicos (como os atributos multivalorados já foram tratados), não há dependência parcial entre atributos (em todas as relações a única dependência funcional é a dos atributos com a chave primária) e, por fim, não há dependência transitiva entre os atributos não primos. Logo, é possível concluir que Modelo Relacional apresentado está na 3ª Forma Normal.

## **5. Projeto Físico do Banco de Dados**

### **5.1 Criação do Banco de Dados**

Para criar o Banco de Dados, utilizamos o nome `empresa_celular` e criamos as tabelas especificadas no modelo relacional. O script para a criação do banco de dados e das tabelas está no arquivo *esquema.sql*, junto das funções trigger. Já o script para a inserção de dados no banco está disponível no arquivo *insercoes.sql*.

### **5.2 Adoção de Políticas de Restrições de Integridade**

As restrições de identidade adotadas foram definidas com base nas diretrizes estabelecidas no Projeto Conceitual (Tabela 7), além de que foram cuidadosamente descritas para garantir a integridade e a consistência dos dados, alinhando-se às políticas estabelecidas.

#### **Departamento:**

**num\_funcionario:** O valor padrão é 0, pois quando um departamento acaba

de ser criado ele não possui funcionários ainda, assim foi usado a cláusula DEFAULT 0. Mas foi utilizado uma trigger para calcular esse valor a cada inserção/remoção de um funcionário.

**Estoque:**

**capacidade\_max:** Valor deve ser diferente de Nulo e maior ou igual a 0, logo foram usadas as cláusulas NOT NULL e CHECK (capacidade\_max >= 0).

**status:** Valor deve ser diferente de Nulo e deve ser “Manhã”, “Tarde” ou “Noite”, portanto usamos as cláusulas: NOT NULL e CHECK (STATUS IN('ADEQUADO', 'VAZIO', 'CHEIO')), para garantir que o valor esteja dentro do domínio especificado.

**Cliente:**

**cpf e email:** Apesar de não ser chave primária, deve ser único para cada funcionário, logo usamos a cláusula UNIQUE.

**nome:** Valor obrigatório, logo deve ser diferente de Nulo. Então usamos a cláusula NOT NULL para que o atributo assuma ao menos alguma String.

**Endereco\_Cliente:**

**rua, bairro, cidade, estado e pais:** Todos os valores são obrigatórios, então usamos a cláusula NOT NULL para cada um desses atributos.

**id\_cliente:** Chave estrangeira que referencia a chave primária idcliente na relação Cliente. Desse modo, utilizamos a cláusula CASCADE, para as operações de DELETE e UPDATE, pois, em caso de remoção de um cliente, a chave estrangeira id\_cliente fará referência a um cliente que não existe. Ou se a chave primária de Cliente for atualizada, endereco\_cliente fará referência a uma chave estrangeira desatualizada. Portanto, para garantir a integridade referencial, usamos esta cláusula.

**Telefone\_cliente:**

**numero:** Valor obrigatório do tipo string. Portanto o atributo deve assumir algum valor, assim usamos a cláusula NOT NULL.

**id\_cliente:** Semelhantemente a relação endereco\_cliente, também usamos a cláusula CASCADE para UPDATE e DELETE, para que a chave estrangeira

id\_cliente acompanhe as mudanças na relação Cliente.

#### **Funcionario:**

**salario:** Valor numerico e obrigatório, e como seu valor mínimo deve ser 1518,00, usamos a cláusula NOT NULL e CHECK, para garantir a que o valor esteja dentro do intervalo mínimo.

**cpf:** Apesar de não ser chave primária, o cpf de cada funcionário deve ser único e não nulo. Portanto utilizamos as cláusulas NOT NULL e UNIQUE.

**nome:** Atributo String de valor obrigatório, logo usamos a cláusula NOT NULL.

**data nascimento:** Valor do tipo Date obrigatório, e o menor valor possível do ano de nascimento de cada funcionário é 1960, e o valor máximo deve ser 2006. Portanto usamos a cláusula CHECK, para conferir se a data está no intervalo especificado.

**nome\_departamento:** Chave estrangeira que faz referencia a nome na relação departamento. Por isso usamos a cláusula CASCADE, para DELETE, logo se o departamento for excluído o funcionário também será, e para UPDATE, para que a chave estrangeira acompanhe qualquer alteração no nome na relação Departamento.

**area\_atuacao:** Valor obrigatório que deve ser: 'Vendedor', 'Suporte', 'Produtor' e 'Estoquista'. Logo usamos as cláusulas NOT NULL e CHECK IN, para garantir que o atributo possua um valor dentro do domínio especificado.

#### **Vendedor:**

**meta\_semana:** Valor numérico e obrigatório, logo usamos a cláusula NOT NULL apenas.

**idfuncionario:** Chave estrangeira que faz referência a idfuncionario na relação Funcionário. Logo, usamos a cláusula CASCADE para operações de DELETE e UPDATE. para que a chave estrangeira faça referência a um funcionário válido e correto.

#### **Suporte:**

**formacao:** Valor String e obrigatório, logo usamos a cláusula NOT NULL

apenas.

**idfuncionario:** Chave estrangeira que faz referência a idfuncionario na relação Funcionário. Logo, usamos a cláusula CASCADE para operações de DELETE e UPDATE. para que a chave estrangeira faça referência a um funcionário válido e correto.

#### **Produtor:**

**horas\_semana:** Valor int e obrigatório, logo usamos a cláusula NOT NULL apenas.

**idfuncionario:** Chave estrangeira que faz referência a idfuncionario na relação Funcionário. Logo, usamos a cláusula CASCADE para operações de DELETE e UPDATE. para que a chave estrangeira faça referência a um funcionário válido e correto.

#### **Estoquista:**

**Turno:** Atributo que assume o valor de uma String que é obrigatório, deve assumir: 'Manhã', 'Tarde'ou 'Noite', logo usamos a cláusula NOT NULL apenas e CHECK para verificar se o valor está dentro do domínio especificado.

**idfuncionario:** Chave estrangeira que faz referência a idfuncionario na relação Funcionário. Logo, usamos a cláusula CASCADE para operações de DELETE e UPDATE. para que a chave estrangeira faça referência a um funcionário válido e correto.

#### **Telefone\_funcionario:**

**numero:** Valor obrigatório do tipo string. Portanto o atributo deve assumir algum valor, assim usamos a cláusula NOT NULL.

**id\_funcionario:** Semelhantemente a relação telefone\_cliente, também usamos a cláusula CASCADE para UPDATE e DELETE, para que a chave estrangeira id\_cliente acompanhe as mudanças na relação Funcionario.

#### **Modelo:**

**tamanho\_memoria:** Atributo do tipo Int e obrigatório, que deve estar dentro do intervalo  $2 \leq \text{tamanho\_memoria} \leq 256$ . Logo, usamos a cláusula NOT NULL e



CHECK, para garantir que esteja dentro do domínio especificado.

**tipo\_camera:** Atributo do tipo Int, é obrigatório, que deve estar dentro do intervalo  $12 \leq \text{tipo\_camera} \leq 200$ . Logo, usamos a cláusula NOT NULL e CHECK, para garantir que esteja dentro do domínio especificado.

**duracao\_bateria:** Atributo do tipo Int e obrigatório, que deve ser maior que 0. Logo, usamos a cláusula NOT NULL e CHECK, para garantir que esteja dentro do domínio especificado.

### **Pedido**

**valor:** Valor numérico e obrigatório, deve assumir um valor que seja maior ou igual que 999,99. Logo usamos a cláusula NOT NULL e CHECK, para verificar se o valor está no domínio correto.

**descricao:** Atributo do tipo string obrigatório, logo usamos a cláusula NOT NULL.

**datahora:** atributo do tipo timestamp obrigatório, logo usamos a cláusula NOT NULL.

**id\_cliente:** Chave estrangeira que faz referência a relação Cliente, caso o cliente seja deletado do banco essa chave estrangeira terá valor NULL, e se o cliente for alterado, a chave estrangeira deve acompanhar essa mudança. Logo usamos a cláusula ON DELETE SET NULL, e CASCADE para operação de UPDATE.

**id\_funcionario:** Chave estrangeira que faz referência a relação Funcionário, semelhante ao caso de id\_cliente, caso o funcionário seja deletado do banco essa chave estrangeira terá valor NULL, e se o funcionário for alterado, a chave estrangeira deve acompanhar essa mudança. Logo usamos a cláusula ON DELETE SET NULL, e CASCADE para operação de UPDATE.

### **Celular:**

**nome\_modelo:** Chave estrangeira que faz referência a relação Modelo. Foram utilizadas as cláusulas CASCADE em UPDATE e DELETE. Pois caso a chave primária nome da relação Modelo seja modificada, então as tuplas em Celular que fazem referência à essa chave devem acompanhar essa alteração. Caso contrário, a chave estrangeira faria uma referência a uma tupla espúria, violando a

restrição de integridade referencial. Já o uso do CASCADE em DELETE se dá por conta de, caso um modelo seja excluído do banco de dados, as tuplas em telefone que fazem referência a esse modelo estariam violando a restrição referencial, logo seria necessário excluir essas tuplas na relação Celular.

**id\_pedido:** Chave estrangeira que faz referência a relação Pedido. Assim como em nome\_modelo, usamos as cláusulas ON DELETE SET NULL, caso o pedido seja excluído, essa FK assumirá valor nulo e não faça referência a uma tupla que não existe. E também CASCADE para operação UPDATE, para atualizar a FK caso a chave primária da tupla em Pedido seja modificada.

#### **Peça:**

**nome:** Atributo do tipo String obrigatório, foi usada a cláusula NOT NULL.

**quantidade:** Atributo Int obrigatório, que deve assumir um valor maior ou igual a 0. Logo foram utilizadas as cláusulas NOT NULL e CHECK.

**nome\_modelo:** Chave estrangeira que faz referência a relação Modelo. Foram usadas as cláusulas NOT NULL e CASCADE para a operação DELETE. Para deletar a tupla em peça caso o seu respectivo modelo seja excluído.

#### **Solicitação:**

**descricao:** Atributo do tipo String obrigatório, foi usada a cláusula NOT NULL.

**datahora:** Atributo do tipo Timestamp obrigatório, foi usada a cláusula NOT NULL.

**status:** Atributo do tipo String obrigatório, que deve assumir o valor: 'Aberto', 'Em andamento', 'Concluído' ou 'Cancelado'.

**id\_cliente:** Chave estrangeira da relação Cliente. Foram usadas as cláusulas CASCADE para as operações de DELETE E UPDATE. Para manter atualizada a FK em caso de alterações na tupla em Cliente, e para deletar a solicitação caso o cliente seja deletado, pois toda solicitação deve obrigatoriamente estar associada a um cliente.

**id\_funcionario:** Chave estrangeira da relação Funcionário. Foram usadas as cláusulas CASCADE para as operações de DELETE E UPDATE. Para manter atualizada a FK em caso de alterações na tupla em Funcionário, e para deletar a

solicitação caso o funcionário responsável seja deletado do Banco, pois toda solicitação deve estar associada a um funcionário

**numero\_serie e nome\_modelo:** Fazem referência a relação Celular. E como toda solicitação deve estar associada a um dispositivo, foram usadas a cláusula CASCADE, para DELETE e UPDATE.

### 5.3 Alimentação do Banco de Dados

Respeitando todas as restrições impostas anteriormente no Projeto Conceitual, as inserções de alimentação do banco de dados estão no arquivo insercoes.sql.

## 6. Especificação de Consultas em Álgebra Relacional e na SQL

Nesta seção, serão descritas as consultas de seleção especificadas na Seção 3, seguindo a Álgebra Relacional (AR) e a SQL. Todas foram ajustadas para atender aos seguintes requisitos:

- Todas as consultas devem envolver duas ou mais tabelas.
- Pelo menos duas consultas devem utilizar funções de agregação (COUNT, SUM, MIN, MAX, AVG) com agrupamento (GROUP BY).
- Consultas que envolvem ordenamento não precisam representar essa ordenação na Álgebra Relacional.
- Para cálculos envolvendo datas, como a idade de um estudante, utilizamos a projeção generalizada: (today – data\_nascimento).

### Consulta 1 – Quantas vendas foram realizadas por mês e por modelo?

Essa consulta retorna a quantidade total de vendas realizadas, agrupadas por mês e modelo do celular. Responsável: André

**Álgebra Relacional (AR):**

$\gamma_{mes, modelo, COUNT(*) \rightarrow total\_vendas}(venda \bowtie celular)$

**SQL:**

```

SELECT
    EXTRACT(MONTH FROM v.data_venda) AS mes,
    c.modelo,
    COUNT(*) AS total_vendas
FROM venda v
JOIN celular c ON v.id_celular = c.id_celular
GROUP BY mes, c.modelo;

```

**Consulta 2 - Quais os funcionários que pertencem ao setor de vendas e nunca venderam um celular do modelo Jaspion 3?**

Essa consulta identifica os funcionários do setor de vendas que nunca realizaram uma venda do modelo Jaspion 3. Responsável: André

**Álgebra Relacional (AR):**

$$\sigma_{area\_atuacao = 'VENDEDOR'}(funcionario) \bowtie (\sigma_{modelo = 'Jaspion 3'}(venda \bowtie celular))$$

**SQL:**

```

SELECT f.nome
FROM funcionario f
LEFT JOIN venda v ON f.idfuncionario = v.idfuncionario
LEFT JOIN celular c ON v.id_celular = c.id_celular AND
c.modelo = 'Jaspion 3'
WHERE f.area_atuacao = 'VENDEDOR' AND c.id_celular IS
NULL;

```

**Consulta 3 - Qual a média salarial dos funcionários do setor de produção entre 22 e 40 anos?**

Calcular a média salarial dos funcionários do setor de produção que possuem entre 22 e 40 anos. Responsável: Pedro

**Álgebra Relacional (AR):**

$$temp1 \leftarrow \Pi_{salario}(\sigma_{area\_atuacao = 'PRODUTOR' \wedge 22 \leq IDADE \leq 40}(FUNCIONARIO))$$

$$\gamma_{media}(salario)(temp1)$$

**SQL:**

```
SELECT AVG(SALARIO) AS MEDIA_SALARIAL FROM FUNCIONARIO
WHERE AREA_ATUACAO = 'PRODUTOR' AND IDADE BETWEEN 22 AND
40;
```

**Consulta 4 - Quais clientes fizeram solicitações de suporte de um modelo em um determinado mês?**

Identificar os clientes que já realizaram solicitações de suporte de um modelo em um determinado mês. Responsável: Pedro

**Álgebra Relacional (AR)**

$$clientesolicitacao \leftarrow CLIENTE \bowtie SOLICITACAO$$

$$clientemodelo \leftarrow \sigma_{nome\_modelo = \langle modelo \rangle}(clientesolicitacao)$$

$$clientemodelomes \leftarrow \sigma_{MONTH(datahora) = \langle mes \rangle}(clientemodelo)$$

$$\pi(id, nome, cpf, email)(clientemodelomes)$$
**SQL**

```
SELECT DISTINCT c.IDCLIENTE, c.NOME, c.CPF, c.EMAIL
FROM CLIENTE c
JOIN SOLICITACAO s on c.IDCLIENTE = s.IDSOLICITACAO
WHERE s.NOME_MODELO = <nome_modelo> AND
      EXTRACT(MONTH FROM s.DATAHORA) = <mes>
```

**Consulta 5 - Quais dispositivos que foram vendidos nunca foram levados para suporte?**

Listar os dispositivos que foram vendidos e nunca foram levados para suporte. Responsável: Pedro

**Álgebra Relacional (AR)**

$$celularesvendidos \leftarrow \sigma_{idpedido \neq NULL}(CELULAR)$$

$$celularsuporte \leftarrow CELULAR \bowtie SUPORTE$$

$$celularnuncasuporte \leftarrow celularesvendidos - celularsuporte$$

$$\pi(num\_serie, nome\_modelo)(celularnuncasuporte)$$
**SQL**

```
SELECT c.NUMERO_SERIE, c.NOME_MODELO
FROM CELULAR c
LEFT JOIN SOLICITACAO s ON c.NUMERO_SERIE = s.NUMERO_SERIE
WHERE c.IDPEDIDO IS NOT NULL AND s.NUMERO_SERIE IS NULL;
```

**Consulta 6 - Quais e quantos modelos de celular possuem disponíveis no estoque?**

Verificar quais modelos de celular estão disponíveis no estoque e quantas unidades há de cada um. Responsável: Gabriel

**Álgebra Relacional (AR):**

$$\Pi_{NOME\_MODELO, COUNT(NUMERO\_SERIE)}(\sigma_{IDPEDIDO IS NULL}(CELULAR)) \text{ GROUP BY } NOME\_MODELO$$
**SQL:**

```
SELECT C.NOME_MODELO, COUNT(*) AS QUANTIDADE_DISPONIVEL
FROM CELULAR C
LEFT JOIN PEDIDO P ON C.IDPEDIDO = P.IDPEDIDO
WHERE C.IDPEDIDO IS NULL
GROUP BY C.NOME_MODELO;
```

**Consulta 7 - Quais as solicitações de suporte com status "em andamento" ou "aberto", incluindo o funcionário e cliente responsável?**

Listar as solicitações de suporte com status "em andamento" ou "aberto", incluindo o funcionário responsável e o cliente solicitante. Responsável: Gabriel

**Álgebra Relacional (AR):**

$$\Pi_{S.CODIGO, S.DESCRICAO, S.DATAHORA, S.STATUS, F.NOME, C.NOME}(\sigma_{S.STATUS \in \{"EM ANDAMENTO", "ABERTO"\}}(S \bowtie S.ID\_SUPORTE = F.IDFUNCIONARIO \bowtie S.ID\_CLIENTE = C.IDCLIENTE))$$
**SQL:**

```
SELECT
    S.CODIGO,
    S.DESCRICAO,
    S.DATAHORA,
    S.STATUS,
    F.NOME AS FUNCIONARIO_RESPONSAVEL,
    C.NOME AS CLIENTE_SOLICITANTE
FROM SOLICITACAO S
JOIN FUNCIONARIO F ON S.ID_SUPORTE = F.IDFUNCIONARIO
JOIN CLIENTE C ON S.ID_CLIENTE = C.IDCLIENTE
WHERE S.STATUS IN ('EM ANDAMENTO', 'ABERTO');
```

**7. Triggers e Views**

Para a entrega final e auxiliar na manutenção do Banco de Dados, o grupo criou Triggers e Views, a fim de completar o projeto. Foram criadas Duas trigger para a função atualizar\_num\_funcionarios, que faz uma nova contagem no número de funcionários de cada departamento, sempre que um novo funcionário é adicionado (INSERT) ou deletado (DELETE), e outras duas trigger para a função atualiza\_idade, que faz o cálculo de todo funcionário assim que ele sofre uma atualização (UPDATE) ou é inserido no banco (INSERT). Por fim, foram criadas três views, uma para que retorna todos os funcionários que são Vendedores, para verificar a ID\_FUNCIONARIO em Pedido, uma view que retorna todos que são Suporte, para verificar a ID\_FUNCIONARIO em Solicitação, e por fim, uma view que retorna quantos celulares cada modelo possui. O script com o código está no arquivo trigger\_views.sql.

## 8. Considerações Finais

O desenvolvimento deste trabalho permitiu uma compreensão mais aprofundada sobre os bancos de dados, desde os conceitos teóricos até a aplicação prática por meio da modelagem e da construção de consultas. Durante o processo, foram utilizadas ferramentas fundamentais, como os Modelos e Diagramas Entidade-Relacionamento (MER e DER), que possibilitaram a organização e representação estruturada dos dados de forma clara e eficiente.

A criação do MER permitiu uma visualização conceitual das entidades, atributos e seus relacionamentos, enquanto a conversão para DER trouxe um nível mais detalhado, incorporando a definição de chaves primárias e estrangeiras, assegurando a consistência e integridade do modelo.

Além disso, a formulação de consultas por meio da Álgebra Relacional (AR) e da SQL demonstrou a importância das operações de seleção, projeção e junção para a extração e manipulação eficiente dos dados. Essa abordagem destacou a relevância das boas práticas na modelagem e no desenvolvimento de consultas otimizadas, essenciais para o desempenho dos sistemas.

Por fim, a construção de um banco de dados fictício proporcionou uma experiência prática e completa, evidenciando a necessidade de um planejamento estruturado e consultas eficientes. O conhecimento adquirido ao longo do trabalho será de grande utilidade para aplicações futuras, tanto no âmbito acadêmico quanto profissional, contribuindo para o desenvolvimento de sistemas de gerenciamento de dados mais eficazes e confiáveis.

## 9. Referências

UDEMY. Bancos de Dados Relacionais: do Básico ao Avançado. Udeemy, [2024]. Disponível em: <http://www.udemy.com/course/bancos-de-dados-relacionais-basico-avancado/>. Acesso em: 13/01/2025.