



## Lista de Exercícios 01 - LE1

Claudio da Aparecida Meireles Filho - 2321070

### Exercício 01:

A sequência de operações em Álgebra Relacional que produz como resultado uma relação com CPF e nome dos professores que não residem no mesmo município onde lecionam é:

```

$$\pi(\text{CPF}, \text{nome-prof}) (\sigma(\text{PROFESSOR.cod-municipio-residencia} \neq \text{ESCOLA.cod-municipio}) (\text{ESCOLA} \bowtie (\text{ESCOLA.cod-escola} = \text{PROFESSOR.cod-escola}) \text{PROFESSOR}))$$

```

Essa operação realiza uma junção natural entre as tabelas ESCOLA e PROFESSOR com base nos códigos das escolas e, em seguida, filtra os resultados para identificar os professores que não residem no município onde trabalham. Finalmente, a projeção seleciona apenas os atributos **CPF** e **nome-prof** para a relação resultante.

### Exercício 02:

A tradução da consulta SQL para Álgebra Relacional é:

```

$$\sigma(\text{Setor} = \text{'cadastro'}) (\text{Atendimento})$$

```

Essa expressão em Álgebra Relacional seleciona as tuplas da tabela **Atendimento** onde o atributo **Setor** é igual a 'cadastro', resultando na contagem dos registros que satisfazem essa condição.

### Exercício 03:

A expressão correta em Álgebra Relacional que permite obter o nome do diretor do filme com código igual a 3 é:

```

$$\pi(\text{dnome}) (\sigma(\text{fcodigo} = 3) (\text{FILMES} \bowtie (\text{FILMES.cod_diretor} = \text{DIRETORES.dcodigo}) \text{DIRETORES}))$$

```

Essa operação realiza uma junção entre as tabelas FILMES e DIRETORES com base nos códigos dos diretores e, em seguida, filtra para o filme com código igual a 3. Por fim, projeta apenas o nome do diretor (**dnome**).

#### Exercício 04:

A expressão da Álgebra Relacional que gera uma relação com duas colunas, sendo a primeira o nome do candidato e a segunda o número da sala onde este fará a prova, é:

```

$$\pi(\text{Nome}, \text{Numero}) ((\text{CANDIDATO} \bowtie \text{CANDIDATO.Inscricao} = \text{ALOCA.Inscricao} \text{ ALOCA}) \bowtie \text{ALOCA.Numero} = \text{SALA.Numero} \text{ SALA})$$

```

A operação realiza uma junção entre as tabelas CANDIDATO e ALOCA com base na inscrição, e em seguida, outra junção entre ALOCA e SALA com base no número da sala. Finalmente, projeta o nome do candidato e o número da sala.

#### Exercício 05:

A sequência de operações em Álgebra Relacional que resulta na relação exibida na Figura 2, contendo apenas a coluna C com os valores 'Fusca' e 'Opala', é:

```

$$\pi(C) (\sigma(A = G) (S \bowtie T))$$

```

Essa operação realiza uma junção entre as tabelas S e T com base na igualdade dos atributos A e G, filtra os registros onde essa condição é verdadeira e, finalmente, projeta apenas a coluna C.

#### Exercício 06:

A expressão da Álgebra Relacional que atribui a RES a sigla da UF dos clientes que realizaram compras com Valor\_Total superior a 5.000 é:

```

$$\pi(UF) ((\sigma(\text{Valor\_Total} > 5000) \text{ VENDAS}) \bowtie \text{VENDAS.IDCliente} = \text{CLIENTES.ID} \bowtie \text{CLIENTES.IDCidade} = \text{CIDADES.IDCid} \text{ CIDADES})$$

```

A operação começa selecionando as vendas onde o Valor\_Total é superior a 5.000, realiza uma junção entre as tabelas VENDAS e CLIENTES com base no ID do cliente e, por fim, junta a tabela resultante com CIDADES para obter a UF correspondente.

#### Exercício 07:

A expressão da Álgebra Relacional representando a tabela formada pelos códigos (COD\_FUNC) e nomes (NOME) dos funcionários que ganham salário (SAL) entre 1000 e 3000 reais e trabalham no departamento de código (COD\_DEP) 3 é:

```

$$\pi(\text{COD\_FUNC}, \text{NOME}) (\sigma(\text{SAL} \geq 1000 \wedge \text{SAL} \leq 3000 \wedge \text{COD\_DEP} = 3) (\text{TAB\_FUNC}))$$

```

Essa operação seleciona os registros onde o salário está entre 1000 e 3000 e o código do departamento é 3, projetando em seguida os códigos e nomes dos funcionários que satisfazem essas condições.

#### Exercício 08:

##### Definição das Chaves:

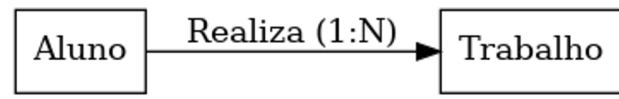
- **Tabela Cliente:**
  - **Chave Primária:** CodCliente
  - **Chaves Candidatas:** CodCliente
  - **Chaves Estrangeiras:** Nenhuma
- **Tabela Conta-Corrente:**
  - **Chave Primária:** NumConta
  - **Chaves Candidatas:** NumConta

- **Chaves Estrangeiras:** `CodCliente` , `CodAgencia`
  - **Tabela Agência:**
    - **Chave Primária:** `CodAgencia`
    - **Chaves Candidatas:** `CodAgencia`
    - **Chaves Estrangeiras:** Nenhuma
  - **Tabela Empréstimo:**
    - **Chave Primária:** `NumEmprest`
    - **Chaves Candidatas:** `NumEmprest`
    - **Chaves Estrangeiras:** `CodCliente` , `CodAgencia`
- a) O que acontece se deletarmos a linha da tabela Empréstimo?**
- Se deletarmos a linha na tabela Empréstimo, o registro específico desse empréstimo será removido. Isso não afetará diretamente outras tabelas, mas a informação sobre esse empréstimo será perdida.
- b) O que acontece se deletarmos a linha na tabela Agência?**
- Se deletarmos a linha na tabela Agência, todas as contas e empréstimos associados àquela agência serão afetados. Dependendo da regra de integridade referencial:
    - **Deleção em Cascata:** As contas e empréstimos relacionados serão excluídos.
    - **Chave Estrangeira Nula:** As referências à agência nas tabelas filhas serão definidas como nulas.
- c) Como fica a tabela ContaCorrente se deletarmos a linha seguinte da tabela Cliente, considerando a manutenção da integridade referencial?**
- Se deletarmos o cliente "Carlos Pereira" (`CodCliente` = 2), as contas associadas na tabela ContaCorrente serão afetadas de acordo com a regra de integridade referencial:
    - **Deleção em Cascata:** As linhas na tabela ContaCorrente com `CodCliente` = 2 serão excluídas.
    - **Chave Estrangeira Nula:** O campo `CodCliente` nas contas será definido como nulo.
- d) Diferença entre manter a integridade referencial por deleção em Cascata e por chaves estrangeiras nulas:**
- **Deleção em Cascata:** Registros relacionados na tabela filha são automaticamente excluídos quando o registro na tabela pai é deletado.
  - **Chaves Estrangeiras Nulas:** As chaves estrangeiras nos registros da tabela filha são definidas como nulas, permitindo que o registro na tabela pai seja deletado sem remover registros na tabela filha.
- e) Uma Tabela B pode ter duas Chaves Estrangeiras para uma Tabela A?**
- Sim, uma tabela B pode ter duas chaves estrangeiras que referenciam a mesma tabela A, representando diferentes relacionamentos entre as mesmas entidades.
- f) O que significa uma Chave Estrangeira com Valor Nulo?**
- Uma chave estrangeira com valor nulo indica que o registro na tabela filha não possui uma referência para um registro correspondente na tabela pai, o que pode ser permitido quando o relacionamento entre as tabelas é opcional.
- g) Uma Chave Alternativa pode ter valores repetidos na mesma tabela?**
- Não, uma chave alternativa, assim como a chave primária, deve conter valores únicos, garantindo que cada registro na tabela possa ser identificado de forma exclusiva por qualquer uma das chaves.
- h) Quais dos esquemas de BD apresentados estão errados?**
- **Esquema a e esquema d** estão incorretos:
    - **Esquema a:** tem um atributo duplicado (TELEFONE).
    - **Esquema d:** está incompleto, com uma vírgula adicional indicando falta de um campo.

## Exercício 09:

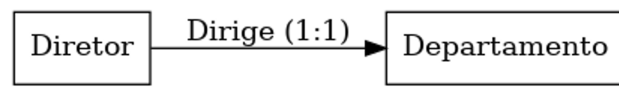
**Situação 1:** Aluno realiza vários trabalhos e um trabalho é realizado por um ou mais alunos.

- **Entidades:** Aluno, Trabalho
- **Relacionamento:** Realiza (Aluno 1Trabalho)
- 



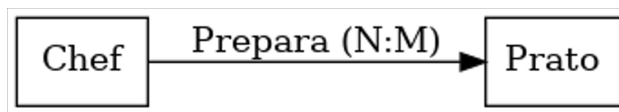
**Situação 2:** Diretor dirige um departamento e um departamento tem um diretor.

- **Entidades:** Diretor, Departamento
- **Relacionamento:** Dirige (Diretor 1:1 Departamento)



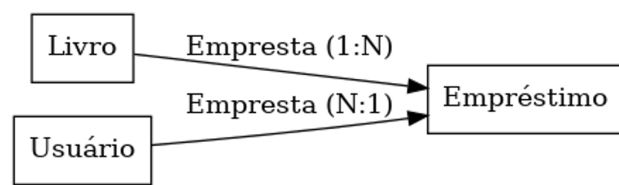
**Situação 3:** Restaurante oferece diversos pratos no menu, preparados por um ou mais chefs.

- **Entidades:** Prato, Chef
- **Relacionamento:** Prepara (Chef NPrato)



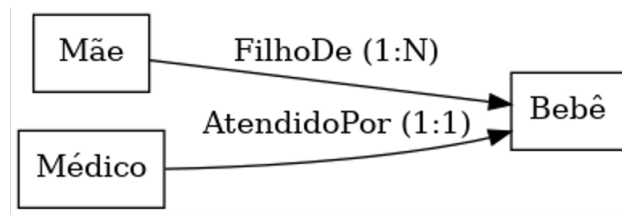
**Situação 4:** Biblioteca organiza livros e registra empréstimos feitos por usuários.

- **Entidades:** Livro, Usuário, Empréstimo
- **Relacionamento:** Empréstimo (Livro 1Empréstimo N:1 Usuário)



**Situação 5:** Berçário informatiza operações para registrar informações de bebês, mães e médicos.

- **Entidades:** Bebê, Mãe, Médico
- **Relacionamentos:** FilhoDe (Mãe 1Bebê), AtendidoPor (Bebê 1:1 Médico)



## Exercício 10:

### Modelo Conceitual:

- **Entidades:** Fornecedor, Cliente, Produto
- **Relacionamentos:** Fornecedor fornece produtos, Cliente compra produtos
- **Cardinalidade:** Fornecedor 1Produto, Cliente NProduto

No modelo conceitual, identificamos as principais entidades e seus relacionamentos. As cardinalidades foram definidas para refletir a realidade do negócio: um fornecedor pode fornecer vários produtos (1), e um cliente pode comprar vários produtos (N). Esse modelo abstrai os detalhes técnicos e se concentra nos elementos fundamentais do sistema, permitindo uma visão clara das interações entre as entidades.

### Modelo Lógico:

- **Tabelas:** Fornecedores, Clientes, Produtos, Fornece, Compra
- **Chaves Primárias:** CNPJ (Fornecedores), CPF (Clientes), Código (Produtos)
- **Chaves Estrangeiras:** CNPJ (Fornece), CPF (Compra)

O modelo lógico traduz as entidades e relacionamentos do modelo conceitual em tabelas e colunas de um banco de dados relacional. Aqui, definimos as chaves primárias e estrangeiras para garantir a integridade dos dados. Este modelo especifica a estrutura do banco de dados em termos que podem ser implementados em um Sistema de Gerenciamento de Banco de Dados (SGBD).

### Modelo Físico:

- **Tipos de Dados:** VARCHAR, INT, DECIMAL, DATE
- **Tabelas:** Definição detalhada das colunas, incluindo tipos de dados específicos.

No modelo físico, escolhemos tipos de dados específicos para as colunas, como VARCHAR para textos e DECIMAL para valores monetários. Essas escolhas são feitas para otimizar a performance do sistema e garantir a integridade e precisão dos dados. O modelo físico detalha a implementação final do banco de dados, incluindo índices e outras otimizações.

### Consultas:

1. Produtos fornecidos por um fornecedor específico:

```

π(Produtos.Código, Produtos.Descrição) (σ(Fornecedores.CNPJ = 'CNPJ_DESEJADO') (Fornece ⋈ Produtos))
  
```

2. Produtos comprados por um cliente específico:

```

π(Produtos.Código, Produtos.Descrição) (σ(Clientes.CPF = 'CPF_DESEJADO') (Compra ⋈ Produtos))
  
```

3. Clientes que compraram um produto específico:

```

π(Clientes.Nome) (σ(Produtos.Código = 'CODIGO_DESEJADO') (Compra ⋈ Clientes))
  
```

### Justificativa:

O modelo conceitual foi elaborado para capturar as principais entidades (Fornecedor, Cliente, Produto) que interagem em um sistema de gerenciamento de loja de roupas. Essas entidades são essenciais para rastrear as operações comerciais, como a compra de produtos por clientes e o fornecimento desses produtos por fornecedores. As cardinalidades refletem

a realidade operacional: um fornecedor pode fornecer vários produtos (1), e um cliente pode comprar diversos produtos enquanto cada produto pode ser adquirido por muitos clientes (N).

No modelo lógico, essas entidades foram mapeadas para tabelas, com chaves primárias e estrangeiras cuidadosamente designadas para manter a integridade dos dados. As chaves estrangeiras permitem a associação entre tabelas (e.g., Fornece e Compra), assegurando que as operações no sistema reflitam os relacionamentos do mundo real.

O modelo físico especifica tipos de dados e outras considerações técnicas necessárias para a implementação no banco de dados. Escolhas como VARCHAR para textos e DECIMAL para valores monetários foram feitas para otimizar a performance e garantir a precisão dos dados.

As consultas em Álgebra Relacional foram elaboradas para cobrir cenários comuns que surgem em um sistema de gerenciamento de loja, como identificar quais produtos foram fornecidos por um determinado fornecedor ou quais clientes compraram um produto específico. Essas consultas refletem necessidades práticas de negócios e demonstram a aplicação do modelo relacional para responder a perguntas comerciais importantes.