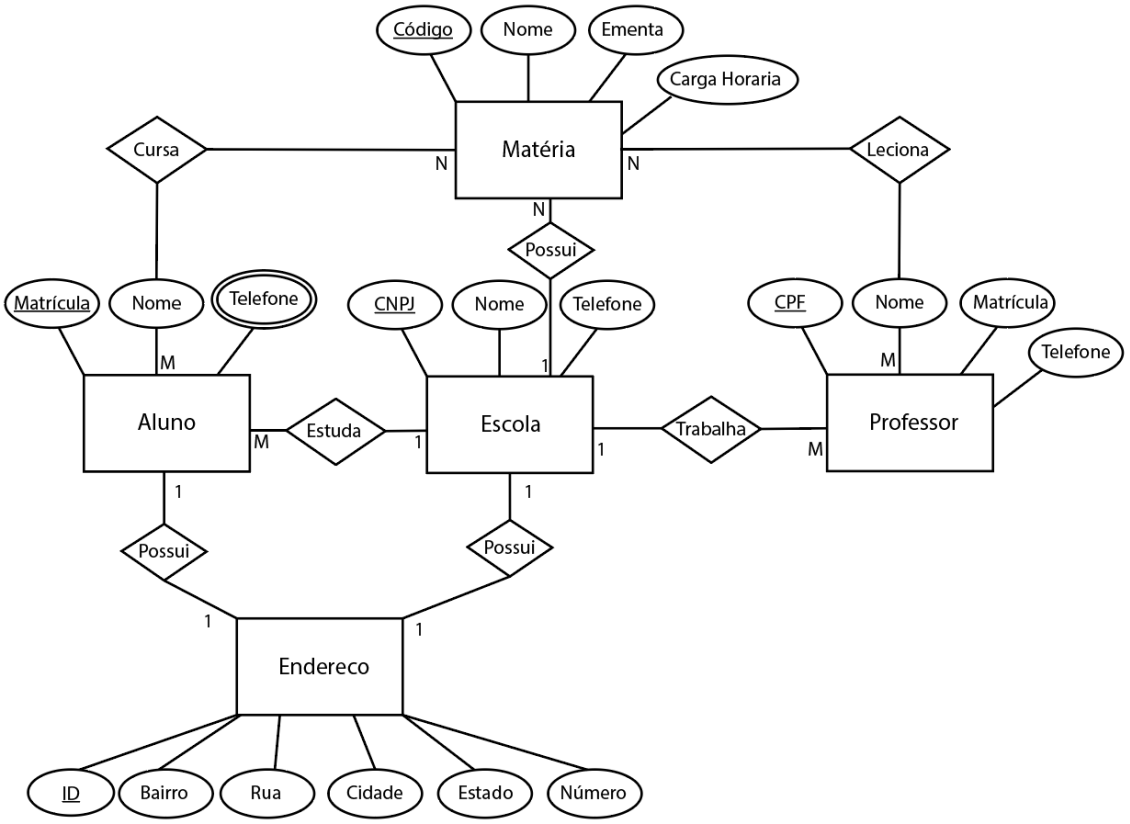
**UNIVERSIDADE FEDERAL DE MINAS GERAIS**
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA: **INTRODUÇÃO A BANCO DE DADOS** - DCC011
2º SEMESTRE DE 2022

# TRABALHO PRÁTICO 2

Carlos Ferreira dos Santos Junior
Mateus Henrique coelho de Paulo
Vinicius Gabriel de Carvalho

## 1. Introdução

O tema escolhido para o nosso trabalho é Escolas, pois as regras referentes a esse modelo de banco de dados já são estabelecidas e bem conhecidas por todos, o que nos possibilita maior entendimento sobre as consultas e seus relacionamentos. Seguindo as orientações passadas em sala, o banco de dados foi criado de acordo com o Esquema Relacional entregue na primeira fase do trabalho. De acordo com o auxílio dos monitores o ER final (com as correções necessárias) se apresenta na figura abaixo:



A modelagem de dados seguiu os padrões propostos no enunciado do trabalho, ou seja, criamos 5 entidades, com 3 formas distintas de relacionamento e pelo menos um atributo multivalorado.

Abaixo está descrito alguns relacionamentos de restrição de integridade e o esquema lógico de nosso banco:

**Atributos:**

Aluno(Matricula, Telefone, nome )
Escola(Cnpj, Nome, Telefone)
Professor(Cpf, Nome, Matrícula, Telefone)
Endereço(Id, Bairro, Rua, Cidade, Estado, Número)
Matéria(Código, Nome, Ementa, Carga Horária)

**Chaves estrangeiras:**

Cursa[Matrícula] - Matéria[Código]
Leciona[Cpf] - Matéria[Código]
Trabalha[Cpf] - Escola[Cnpj]
Possui[Cnpj] - Endereço[Id]
Possui[Matrícula] - Endereço[Id]
Estuda[Matrícula] - Escola[Cnpj]
Possui[Cnpj] - Matéria[Código]

**Restrições de remoção:**

Telefone[Aluno] ->**p** Aluno[Matrícula]
Telefone[Professor] **->p** Professor[Cpf]
Matéria[Código] ->**b** Professor[Cpf]
(Para aluno, resolvemos não remover caso a matéria seja removida, devido ao histórico escolar)
Professor[Cpf] ->**p** Matéria[Código]

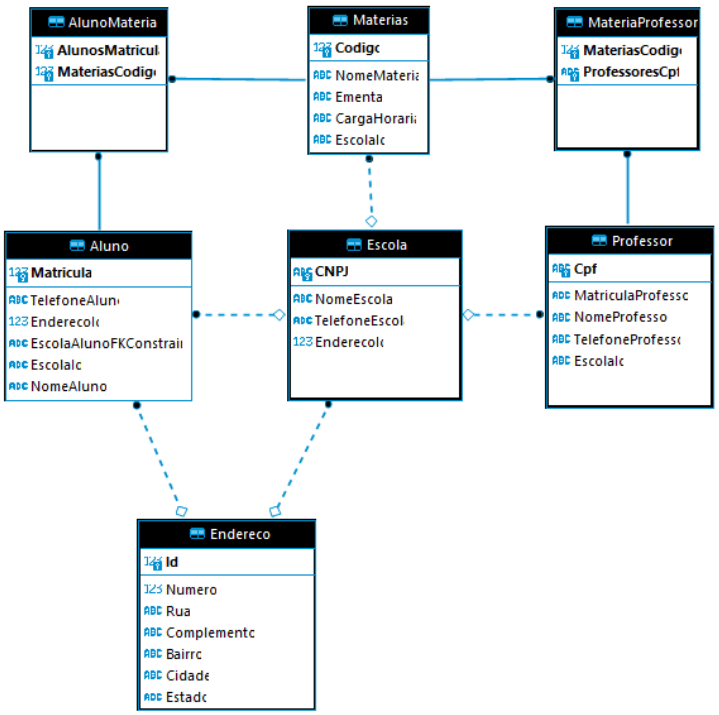
## 2. Características do Banco de Dados

Após a definição das entidades, o banco foi criado seguindo o modelo Entidade Relacionamento previamente proposto e o SGBD utilizado foi o PostgreSQL. Além disso, o banco de dados está hospedado em nuvem utilizando o servidor da AWS e por fim, foram criados alguns scripts para popular o banco com dados. No final o banco ficou com um pouco mais de 22 mil linhas de dados distribuídos da seguinte forma:

- **Escola:** 1 linha
- **Matérias**: 20 linhas
- **Professor**: 255 linhas
- **Endereço**: 1001 linhas
- **Aluno**: 2000 linhas

O restante das linhas foram ocupadas pelas tabelas de relacionamento.

Os scripts utilizaram um gerador de números aleatórios para selecionar as matérias relacionadas a cada aluno e a cada professor, o mesmo vale para a tabela de endereço. A imagem a seguir é o esquema relacional resultante:




## 3. Características do Banco de Dados

**Consultas envolvendo apenas seleção e projeção:**

- Retornar professores com telefone terminado em “7”

```
SELECT "NomeProfessor", "Cpf"
FROM "Professor"
WHERE "TelefoneProfessor" LIKE '%7'
```

**alternativa**: Diferença entre usar ‘*’ e definir colunas
SELECT *
FROM “Professor”
WHERE "TelefoneProfessor" LIKE ‘%7’

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.065 ms | 0.087 ms |
| Alternativa | 0.088 ms | 0.097 ms |

- Retornar matérias com carga horária > 00:45

```
SELECT "NomeMateria"
FROM "Materias"
WHERE "CargaHoraria" > '00:45'
```

**alternativa:** Diferença entre usar ‘*’ e definir colunas

```
SELECT *
FROM "Materias"
WHERE "CargaHoraria" > '00:45'
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.0564 ms | 0.0496 ms |
| Alternativa | 0.0668 ms | 0.0544 ms |

**Consultas que envolvem a junção de duas relações:**

- Retornar Alunos e seus endereços

```
SELECT "NomeAluno" , "Rua", "Numero" , "Bairro" , "Cidade" ,
"Estado" , "Complemento"
FROM  "Aluno" a   JOIN "Endereco" ed
on a."EnderecoId"  = ed."Id"
```

**alternativa**: Sem o uso de JOIN

```
SELECT "NomeAluno" , "Rua", "Numero" , "Bairro" , "Cidade" ,
"Estado" , "Complemento"
FROM  "Aluno" a, "Endereco" ed
WHERE  a."EnderecoId"  = ed."Id"
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.3098 ms | 2.7378 ms |
| Alternativa | 0.3136 ms | 2.8588 ms |

- Alunos que moram em um determinado bairro

```
SELECT a."NomeAluno" , e."Bairro"
FROM "Aluno" a JOIN "Endereco" e
on a."EnderecoId" = e."Id"
WHERE e."Bairro" = 'Bairro 170'
```

**alternativa**: Alterar a ordem das relações

```
SELECT a."NomeAluno" , e."Bairro"
FROM "Endereco" e  JOIN "Aluno" a
on a."EnderecoId" = e."Id"
WHERE e."Bairro" Like 'Bairro 170'
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.3248 ms | 0.3382 ms |
| Alternativa | 0.326 ms | 0.3796 ms |

- Retornar endereços com 2 ou mais alunos

```
SELECT *
FROM (SELECT  e."Rua" , e."Numero" , e."Bairro" , e."Cidade" ,
        count(a."Matricula") as qtalunos
        FROM "Aluno" a JOIN "Endereco" e on a."EnderecoId" = e."Id"
        group by e."Id"
        ) qta
WHERE qtalunos > '1'
```

**alternativa**: Inverter FROM e WHERE ao invés de JOIN

```
SELECT *
FROM (SELECT  e."Rua" , e."Numero" , e."Bairro" ,
         e."Cidade" , count(a."Matricula") as qtalunos
         FROM  "Endereco" e, "Aluno" a
         WHERE  a."EnderecoId" = e."Id"
        group by e."Id"
         ) qta
```

```
WHERE qtalunos > '1'
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.3522 ms | 3.9324 ms |
| Alternativa | 0.3618 ms | 3.824 ms |

**Consultas que envolvem a junção de três ou mais relações:**

- Quais matérias o professor ‘Professor 150’ leciona

```
SELECT  p."NomeProfessor"  , m."NomeMateria"
FROM "Materias" m
JOIN "MateriaProfessor" mp on m."Codigo" = mp."MateriasCodigo"
JOIN "Professor" p on mp."ProfessoresCpf" = p."Cpf"
WHERE p."NomeProfessor" = 'Professor150'
```

**alternativa**:Inverter ordem de FROM

```
SELECT  p."NomeProfessor"  , m."NomeMateria"
FROM "Professor" p
JOIN "MateriaProfessor" mp on  mp."ProfessoresCpf" = p."Cpf"
JOIN "Materias" m on mp."MateriasCodigo" = m."Codigo"
WHERE p."NomeProfessor" = 'Professor150'
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.7154 ms | 0.1434 ms |
| Alternativa | 1.0172 ms | 0.1398 ms |

- Retornar lista de alunos inscritos na matéria ‘Xadrez’

```
SELECT a."NomeAluno" , m."NomeMateria"
FROM "AlunoMateria" am  JOIN "Materias" m
ON am."MateriasCodigo" = m."Codigo"
JOIN "Aluno" a
on a."Matricula" = am."AlunosMatricula"
WHERE m."NomeMateria" = 'Xadrez'
```

**alternativa**: Não usar JOIN

```
SELECT a."NomeAluno", m."NomeMateria"
FROM "Aluno" a , "AlunoMateria" am , "Materias" m
WHERE a."Matricula"  = am."AlunosMatricula"
```

```
AND am."MateriasCodigo" = m."Codigo"
AND m."NomeMateria" = 'Xadrez'
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.5164 ms | 2.6566 ms |
| Alternativa | 0.602 ms | 2.553 ms |

- Retornar matéria com menos professores

```
SELECT   m."NomeMateria" , count(p."Cpf")
FROM "Materias" m
JOIN "MateriaProfessor" mp on m."Codigo" = mp."MateriasCodigo"
JOIN "Professor" p on mp."ProfessoresCpf" = p."Cpf"
group by m."Codigo"  order by "count" limit 1
```

**alternativa**: Uso de WHERE ao invés de JOIN

```
SELECT m."NomeMateria" , count(p."Cpf")
FROM "Materias" m, "MateriaProfessor" mp, "Professor" p
WHERE m."Codigo" = mp."MateriasCodigo" and mp."ProfessoresCpf" = p."Cpf"
group by m."Codigo"  order by "count" limit 1
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.7452 ms | 1.7668 ms |
| Alternativa | 0.7384 ms | 1.753 ms |

**Consultas que envolvem funções de agregação sobre o resultado da junção de pelo menos duas relações:**

- Retornar quantidade de minutos do professor com maior carga horária

```
SELECT sum(chminutos::integer) as QtMinutos
FROM (SELECt substring(m."CargaHoraria", 4) as chminutos,
        p."Cpf" , p."NomeProfessor"
        FROM "Materias" m
        JOIN "MateriaProfessor" mp on m."Codigo" = mp."MateriasCodigo"
        JOIN "Professor" p on mp."ProfessoresCpf" = p."Cpf"
      ) ch
group by ch."Cpf" order by QtMinutos desc limit 1
```

**alternativa**: Usar max() ao invés de ORDER e LIMIT

```
SELECT max(qtminutos) as qtMinutosMax
FROM (SELECT sum(chminutos::integer) as QtMinutos
        FROM (SELECt substring(m."CargaHoraria", 4) as chminutos,
                p."Cpf" , p."NomeProfessor"
                FROM "Materias" m
                JOIN "MateriaProfessor" mp on m."Codigo" = mp."MateriasCodigo"
                JOIN "Professor" p on mp."ProfessoresCpf" = p."Cpf"
                 ) ch
        group by ch."Cpf"
        ) as qt
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.7974 ms | 2.4684 ms |
| Alternativa | 0.7494 ms | 2.3388 ms |

- Retornar quantidade de Alunos de cada Endereço

```
SELECT  e."Rua" , e."Numero" , e."Bairro" , e."Cidade" , count(a."Matricula")
FROM "Aluno" a JOIN "Endereco" e on a."EnderecoId" = e."Id"
group by e."Id"
```

**alternativa**: Sem JOIN

```
SELECT  e."Rua" , e."Numero" , e."Bairro" , e."Cidade" , count(a."Matricula")
FROM "Aluno" a, "Endereco" e
WHERE  a."EnderecoId" = e."Id"
group by e."Id"
```

| | Planning Time (Média) | Execution Time (Média) |
|---|---|---|
| Base | 0.3024 ms | 4.047 ms |
| Alternativa | 0.3044 ms | 3.7618 ms |

**4. Conclusão**

Nosso grupo optou por modelar um banco com base em uma escola. Por ser algo próximo de uma realidade que já vivemos, pensamos que seria uma abordagem interessante, já que, poderíamos imputar nossas vivências e brincar com alguns cenários interessantes. A parte de desenhar o ER e modelar as entidades, decidindo o que teríamos em nosso banco ficou ao meu cargo, Mateus Henrique Coelho. Foi muito interessante decidir quais cenários abordar e como cada um iria impactar nas regras não só do nosso banco, mas também da nossa escola. Hoje em dia escutamos muito sobre

dados sensíveis e esse foi um dos pontos que tomamos cuidado. A modelagem de dados seguiu os padrões propostos no enunciado do trabalho, ou seja, criamos 5 entidades, com 3 formas distintas de relacionamento e pelo menos um atributo multivalorado.

Foi muito interessante tentar aplicar os conceitos aprendidos em sala nesse banco "real", pois pude perceber os desafios que a modelagem de um banco tem e o nível de importância de uma boa modelagem. Na segunda parte, feita pelo Vinicius, onde começamos a inserir os dados e fazer do banco algo real, acabamos nos deparando com um problema na modelagem em alguns atributos, como no de materiais e no de endereços, corrigir isso nos mostrou o quanto essa parte do processo é a mais importante. Acredito que nosso grupo teve um desempenho excelente, pois, aprendemos com nossos erros e conseguimos trazer a realidade uma ideia e modelar a mesma da maneira que aprendemos na matéria.

A princípio, optamos por usar o Spawner para gerar os dados artificiais no nosso banco, porém, por questões de familiaridade de ferramentas e organização decidimos utilizar scripts próprios para gerar os dados artificiais necessários. Como utilizamos um banco em nuvem possuíamos um desafio de armazenamento e esse foi um dos motivos pelos quais nós optamos por ter um banco de dados grande mas apenas o suficiente para podermos ter resultados concisos no momento das buscas.

Com relação às consultas, feita pelo Carlos, foi necessário compreender profundamente todo o conjunto de relações entre cada entidade e pensar nos diferentes cenários que essas relações poderiam gerar. É muito intrigante poder criar consultas com base nesses cenários e poder testá-las na prática durante a elaboração de cada uma, pois esse processo de criar e imediatamente ver o resultado dentro do banco de dados gera um entendimento muito maior da teoria. Para cada consulta, procurei entender todos os agentes do esquema e me colocar no papel de cada um, seja como de diretoria da escola que precisa saber dos endereços dos alunos e atributos socioeconômicos como quantidade de moradores de cada endereço, professores da escola que precisam saber da carga horária das matérias e quais matérias eles lecionam, ou de um aluno que precisa saber em quais matérias está matriculado e quem são seus professores. Também foi curioso poder elaborar essas consultas de maneiras alternativas e observar o impacto dessas mudanças, detalhados através da análise de Planning e Execution time medidos em várias execuções do SQL, algo que pode acabar sendo desprezado mas é bem importante na prática.