

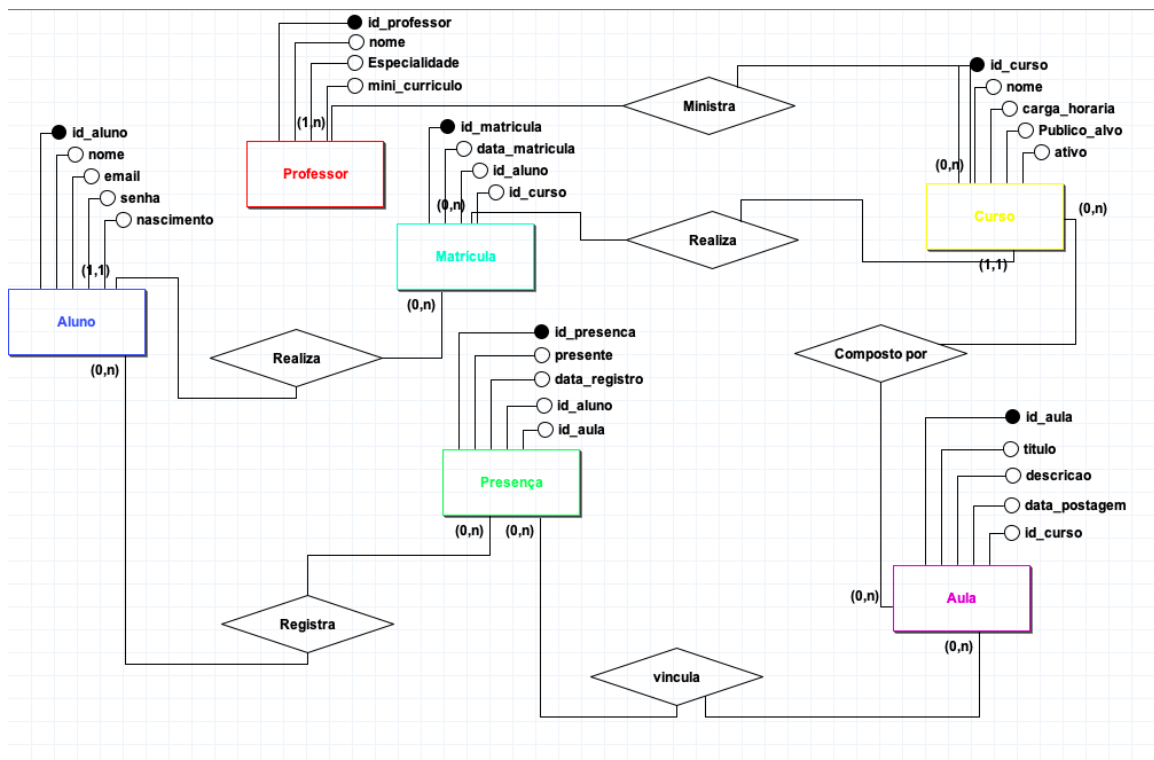
Projeto de Banco de Dados – Escola de Concursos

Parte I – Modelagem do Banco de Dados

1. Modelos Desenvolvidos

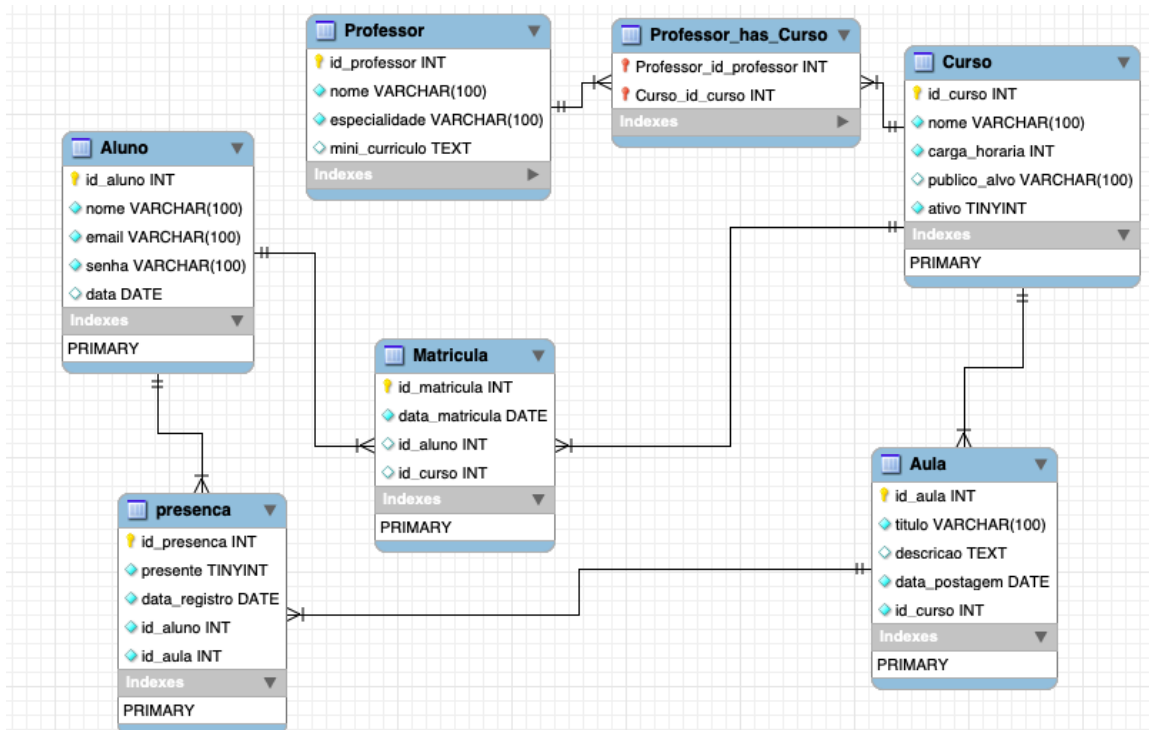
1.1 Modelo Conceitual (DER)

O modelo conceitual foi elaborado no BRModelo e contém as entidades: Aluno, Curso, Professor, Aula, Matrícula e Presença. Os relacionamentos foram definidos com cardinalidades apropriadas, resolvendo as ligações N:N com entidades associativas. O relacionamento Professor-Curso foi representado corretamente por uma tabela associativa.



1.2 Modelo Lógico (Diagrama Relacional)

O modelo lógico foi implementado visualmente no MySQL Workbench com base no modelo conceitual. Foram criadas 6 entidades principais mais uma tabela associativa. Todas as chaves primárias e estrangeiras foram definidas corretamente. Abaixo, o print do modelo:



1.3 Modelo Físico (Script SQL)

O modelo físico foi implementado diretamente em MySQL, incluindo a criação de todas as tabelas com chaves primárias e estrangeiras, e 10 registros inseridos em cada tabela, conforme exigido. O script final está disponível em anexo.

Parte II – Consultas e Operações em SQL

Este documento apresenta o conjunto de consultas SQL realizadas sobre o banco de dados "escola_concursos", com foco em operadores, agrupamentos, junções e subconsultas, conforme critérios de avaliação estabelecidos. Cada grupo apresenta a consulta, uma explicação resumida e a simulação de execução visual.

1. SELECT + WHERE

```
SELECT * FROM aluno WHERE nome LIKE 'M%';
```

Descrição: Seleciona todos os alunos cujo nome começa com 'M'.

```
SELECT nome, email FROM aluno WHERE YEAR(data_nascimento) < 1995;
```

Descrição: Seleciona nome e e-mail de alunos nascidos antes de 1995.

2. GROUP BY + ORDER BY + funções agregadas

```
SELECT id_curso, COUNT(*) AS total_alunos FROM matricula GROUP BY id_curso  
ORDER BY total_alunos DESC;
```

Descrição: Conta o número de matrículas por curso e ordena do maior para o menor.

```
SELECT presente, COUNT(*) FROM presenca GROUP BY presente;
```

Descrição: Conta quantos registros estão presentes ou ausentes.

3. Operadores Aritméticos

```
SELECT carga_horaria, carga_horaria + 20 AS ajustada FROM curso;
```

Descrição: Adiciona 20 horas à carga horária original.

```
SELECT id_aluno, YEAR(CURDATE()) - YEAR(data_nascimento) AS idade FROM aluno;
```

Descrição: Calcula a idade aproximada dos alunos.

4. Operadores de Comparação

```
SELECT * FROM curso WHERE carga_horaria > 200;
```

Descrição: Seleciona cursos com carga horária superior a 200 horas.

```
SELECT * FROM aula WHERE id_curso = 3;
```

Descrição: Seleciona aulas do curso com ID 3.

```
SELECT * FROM professor WHERE especialidade != 'Direito Penal';
```

Descrição: Seleciona professores que não são da área de Direito Penal.

5. Operadores Lógicos (AND, OR)

```
SELECT * FROM curso WHERE carga_horaria > 180 AND ativo = 1;
```

Descrição: Seleciona cursos ativos com mais de 180 horas.

```
SELECT * FROM aluno WHERE YEAR(data_nascimento) > 1990 OR nome LIKE 'J%';
```

Descrição: Seleciona alunos nascidos após 1990 ou cujo nome começa com 'J'.

6. Operadores com NOT

```
SELECT * FROM curso WHERE NOT ativo = 0;
```

Descrição: Seleciona cursos que estão ativos.

```
SELECT * FROM presenca WHERE NOT presente;
```

Descrição: Seleciona registros de ausência.

7. Operadores auxiliares (IS NULL, BETWEEN, LIKE, IN)

```
SELECT * FROM professor WHERE mini_curriculo IS NOT NULL;
```

Descrição: Seleciona professores que possuem mini currículo.

```
SELECT * FROM curso WHERE carga_horaria BETWEEN 150 AND 200;
```

Descrição: Seleciona cursos com carga horária entre 150 e 200.

```
SELECT * FROM aluno WHERE nome LIKE '%Silva%';
```

Descrição: Seleciona alunos com 'Silva' no nome.

8. Funções de agregação

```
SELECT AVG(carga_horaria) FROM curso;
```

Descrição: Retorna a média da carga horária dos cursos.

```
SELECT SUM(presente) FROM presenca;
```

Descrição: Retorna o total de presenças.

```
SELECT COUNT(*) FROM professor;
```

Descrição: Conta o número total de professores.

9. Funções de data

```
SELECT NOW();
```

Descrição: Retorna a data e hora atual do sistema.

```
SELECT nome, YEAR(data_nascimento) FROM aluno;
```

Descrição: Mostra o nome do aluno e o ano de nascimento.

10. Subconsultas

```
SELECT nome FROM curso WHERE id_curso IN (SELECT id_curso FROM matricula  
WHERE id_aluno = 1);
```

Descrição: Seleciona cursos nos quais o aluno com ID 1 está matriculado.

```
SELECT nome FROM aluno WHERE id_aluno IN (SELECT id_aluno FROM presenca  
WHERE presente = 0);
```

Descrição: Seleciona nomes de alunos que tiveram ausência registrada.

11. JOINS e visualização de tabelas

```
SELECT a.nome, c.nome FROM aluno a JOIN matricula m ON a.id_aluno = m.id_aluno  
JOIN curso c ON m.id_curso = c.id_curso;
```

Descrição: Mostra o nome dos alunos com os cursos em que estão matriculados.

12. Tipos de JOIN: INNER, LEFT, RIGHT

```
SELECT * FROM curso INNER JOIN professor_has_curso ON curso.id_curso =  
professor_has_curso.id_curso;
```

Descrição: INNER JOIN entre cursos e seus professores.

```
SELECT * FROM curso LEFT JOIN professor_has_curso ON curso.id_curso =  
professor_has_curso.id_curso;
```

Descrição: LEFT JOIN entre cursos e professores (mostra todos os cursos).

```
SELECT * FROM curso RIGHT JOIN professor_has_curso ON curso.id_curso =  
professor_has_curso.id_curso;
```

Descrição: RIGHT JOIN entre cursos e professores (mostra todos os vínculos com professores).