



EDUCALIVROS

GRAND FINALE

EducaLivros, O Sistema Educacional Inovador



Triggers

TRIGGER 1

Atualizar a quantidade de livros disponíveis após uma compra.

TRIGGER 2

Atualizar a quantidade de livros que o aluno comprou após uma compra.

TRIGGER 3

Atualizar o saldo da cantina do aluno após uma compra.

TRIGGER 4

Diminuir a quantidade de produtos na cantina após a compra.



Triggers



```
CREATE OR REPLACE FUNCTION diminuir_qtd_produto_cantina()
RETURNS TRIGGER AS $$

BEGIN
    UPDATE produtos_cantina
    SET qnt_produtos_cantina = qnt_produtos_cantina - 1
    WHERE nome_produto = NEW.fk_Produtos_cantina_nome_produto;

    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER diminuir_qtd_produto_cantina
AFTER INSERT ON compra
FOR EACH ROW
EXECUTE FUNCTION diminuir_qtd_produto_cantina();

/*
SELECT qnt_produtos_cantina, nome_produto
FROM Produtos_cantina;
*/
```

```
CREATE OR REPLACE FUNCTION atualizar_qnt_exemplares()
RETURNS TRIGGER AS $$
DECLARE
    produto_nome varchar;
BEGIN
    SELECT nome_produto INTO produto_nome
    FROM produto
    WHERE id = NEW.fk_produto_id;

    UPDATE livro
    SET qtd_exemplares = qtd_exemplares - 1
    WHERE nome_livro = produto_nome;

    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER atualizar_qnt_exemplares
AFTER INSERT ON contem
FOR EACH ROW
EXECUTE FUNCTION atualizar_qnt_exemplares();

/*
SELECT qtd_exemplares, nome_livro
FROM livro;
*/
```

Triggers

```
CREATE OR REPLACE FUNCTION atualizar_saldo_cantina()
RETURNS TRIGGER AS $$

DECLARE
    preco_produto_cantina REAL;
BEGIN
    SELECT preco INTO preco_produto_cantina
    FROM produtos_cantina
    WHERE nome_produto = NEW.fk_Produtos_cantina_nome_produto;

    UPDATE aluno
    SET saldo_da_cantina = saldo_da_cantina - preco_produto_cantina
    WHERE id_da_matricula = NEW.fk_Aluno_id_da_matricula;

    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER atualizar_saldo_cantina
AFTER INSERT ON compra
FOR EACH ROW
EXECUTE FUNCTION atualizar_saldo_cantina();

/*
SELECT saldo_da_cantina, id_da_matricula, nome
FROM aluno;
*/
```

```
CREATE OR REPLACE FUNCTION atualizar_qnt_livros_comprados()
RETURNS TRIGGER AS $$

BEGIN
    IF (SELECT fk_categoria_produto_id FROM produto WHERE id = NEW.fk_produto_id) = 1 THEN
        UPDATE aluno
        SET quantidade_livros_comprados = quantidade_livros_comprados + 1
        WHERE id_da_matricula = NEW.id_aluno_compra;
    END IF;

    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER atualizar_qnt_livros_comprados
AFTER INSERT ON contem
FOR EACH ROW
EXECUTE FUNCTION atualizar_qnt_livros_comprados();

/*
SELECT quantidade_livros_comprados, id_da_matricula, nome
FROM aluno;
*/
```



Stored Procedures



OFERECER DESCONTO POR CATEGORIA

Adiciona um desconto em uma categoria de produtos



CALCULAR TOTAL VENDAS DO PRODUTO

Calcula o total de vendas do produto em um intervalo de tempo



CRIAR RELATÓRIO DE VENDAS DO VENDEDOR

Cria um relatório de vendas de um vendedor em um intervalo de tempo



CRIAR RELATÓRIO DE MELHORES VENDEDORES

Cria um relatório para exibir os melhores vendedores em um intervalo

Stored Procedures



Aplicação em código: Desconto por categoria

```
2 -- Executando a função para ver os dados da tabela cadastro
3 SELECT * FROM aluno;
4
5
6 CREATE OR REPLACE PROCEDURE dar_desconto_por_categoria(category INT, percentage REAL)
7 LANGUAGE plpgsql
8 AS $$*
9 BEGIN
10    UPDATE produto
11    SET preco = preco * (1 - percentage)
12    WHERE fk_categoria_produto_id = category;
13 END;
14 $$;
15
16 CALL dar_desconto_por_categoria(1, 0.1);
```

Stored Procedures



Aplicação em código: Total de vendas do produto

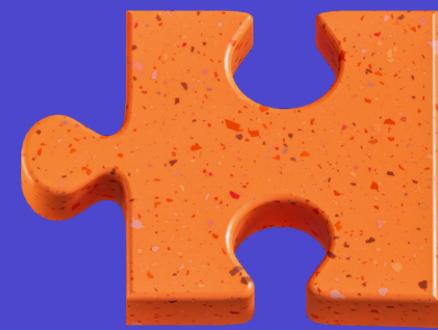
```
1 CREATE OR REPLACE PROCEDURE calcular_total_vendas_produto
2 (produto_id INT, data_inicio DATE, data_fim DATE)
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6
7     CREATE TABLE temp_resultado AS
8     SELECT
9         produto_id AS "Produto ID",
10        data_inicio AS "Data INício",
11        data_fim AS "Data Fim",
12        SUM(v.valor_total) AS "Total Vendas"
13    FROM
14        venda v
15    INNER JOIN
16        contem c ON v.id_venda = c.fk_venda_id_venda
17    WHERE
18        c.fk_produto_id = produto_id
19        AND v.data_venda BETWEEN data_inicio AND data_fim
20    GROUP BY
21        produto_id, data_inicio, data_fim;
22 END;
23 $$;
24
25 CALL calcular_total_vendas_produto(1, '2024-06-01', '2024-06-10');
```

Views



1. EaD x Presencial

Qual a quantidade de alunos cursando presencial e a quantidade de alunos cursando EAD?



2. Inadimplência

Quais responsáveis financeiros estão com pendências no status de pagamento de seus dependentes?



3. Bairros comuns

Quais os bairros mais comuns entre os alunos?



4. Mais vendidos

Quais são os produtos mais vendidos da livraria?

Views



```
create view servicos_maisRegistrados AS
    SELECT s.tipo_servico, COUNT(l.id_livro) AS total_por_tipo
    FROM livro l
    JOIN servico s ON l.fk_servico_id_servico = s.id_servico
    GROUP BY s.tipo_servico
    ORDER BY total_por_tipo DESC;
```

download, entrega ou retirada?

```
create view bairros_mais_comuns AS
    SELECT p.bairro, COUNT(*) AS total_alunos
    FROM aluno a
    JOIN pessoa p ON a.fk_pessoa_id = p.id
    GROUP BY p.bairro
    ORDER BY total_alunos DESC
```

Bairros mais comuns para fornecer transporte

Segurança

SEGURANÇA 1

Apenas coordenadores podem inserir ou deletar a tabela disciplina.

SEGURANÇA 2

Apenas os gerentes da escola podem inserir ou deletar a tabela departamento.

SEGURANÇA 3

Apenas os vendedores da livraria podem inserir ou deletar a tabela cadastro.

SEGURANÇA 4

Apenas organizadores de evento da livraria podem inserir ou deletar a tabela palestra_debate.

Segurança



```
CREATE ROLE coordenador;
CREATE ROLE gerente_escola;
CREATE ROLE vendedor_livraria;
CREATE ROLE organizador_eventos;

GRANT INSERT, DELETE ON disciplina TO coordenador;
GRANT INSERT, DELETE ON departamento TO gerente_escola;
GRANT INSERT, DELETE ON cadastro TO vendedor_livraria;
GRANT INSERT, DELETE ON palestra_debate TO organizador_eventos;
```

Normalização

tabela palestra_debate



id_evento	fk_livro_id_livro	taxa_ingeresso	sala	lotacao	horario	livro_tema	fk_livro_id_livro
321	0923	15	T-02	40	13:00	O Senhor dos Anéis	928
223	0954	25	A-04	80	18:30	Dom Casmurro	341

2FN:

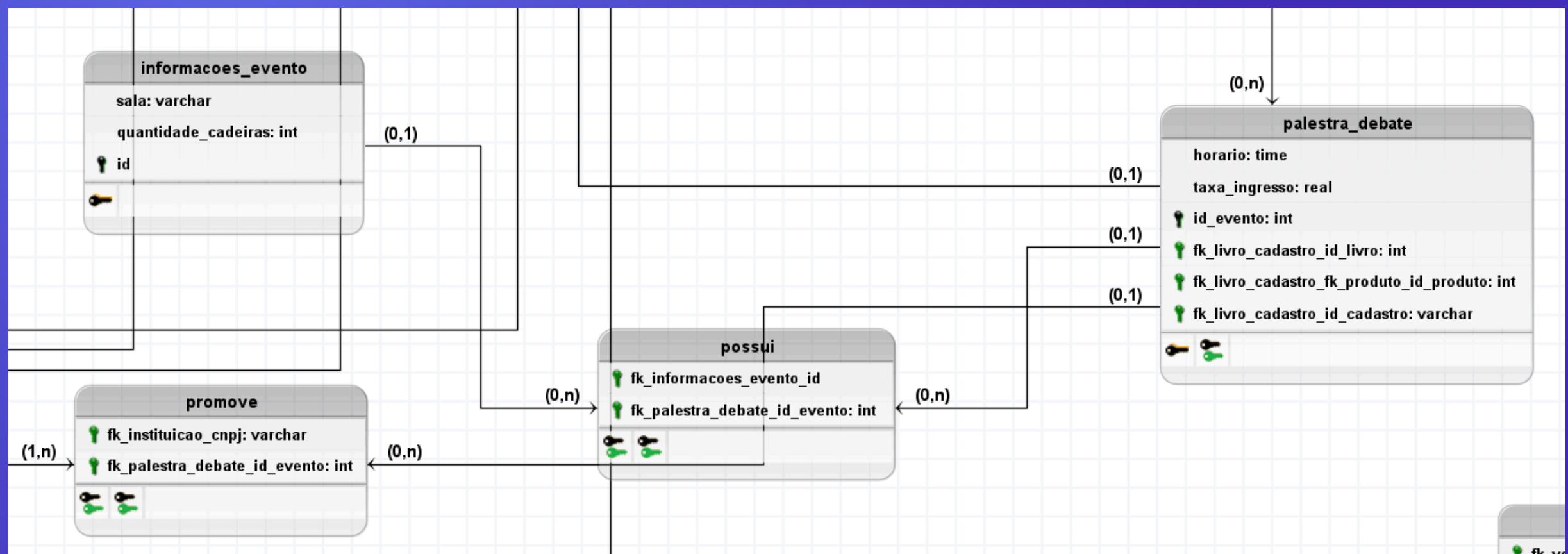
fk_livro_id_livro	livro_tema
0923	O Senhor dos Anéis
0954	Dom Casmurro

3FN:

sala	lotacao
T-02	40
A-04	80

Normalização

Aplicação no Br Modelo e no DDL



```
304 CREATE TABLE informacoes_evento (
305     sala varchar,
306     quantidade_cadeiras int,
307     id int PRIMARY KEY
308 );
```

Normalização

tabela material_papelaria



fk_produto_id_produto	fk_instituicao_cnpj	fk_fornecedor_cnpj	num_registro_inmetro	situacao_registro	certificado_inmetro	tipo_teste
0001	234567	789873	678	ativo	conformidade	físico
0230	654321	541234	543	suspensão	meio ambiente	químico

2FN:

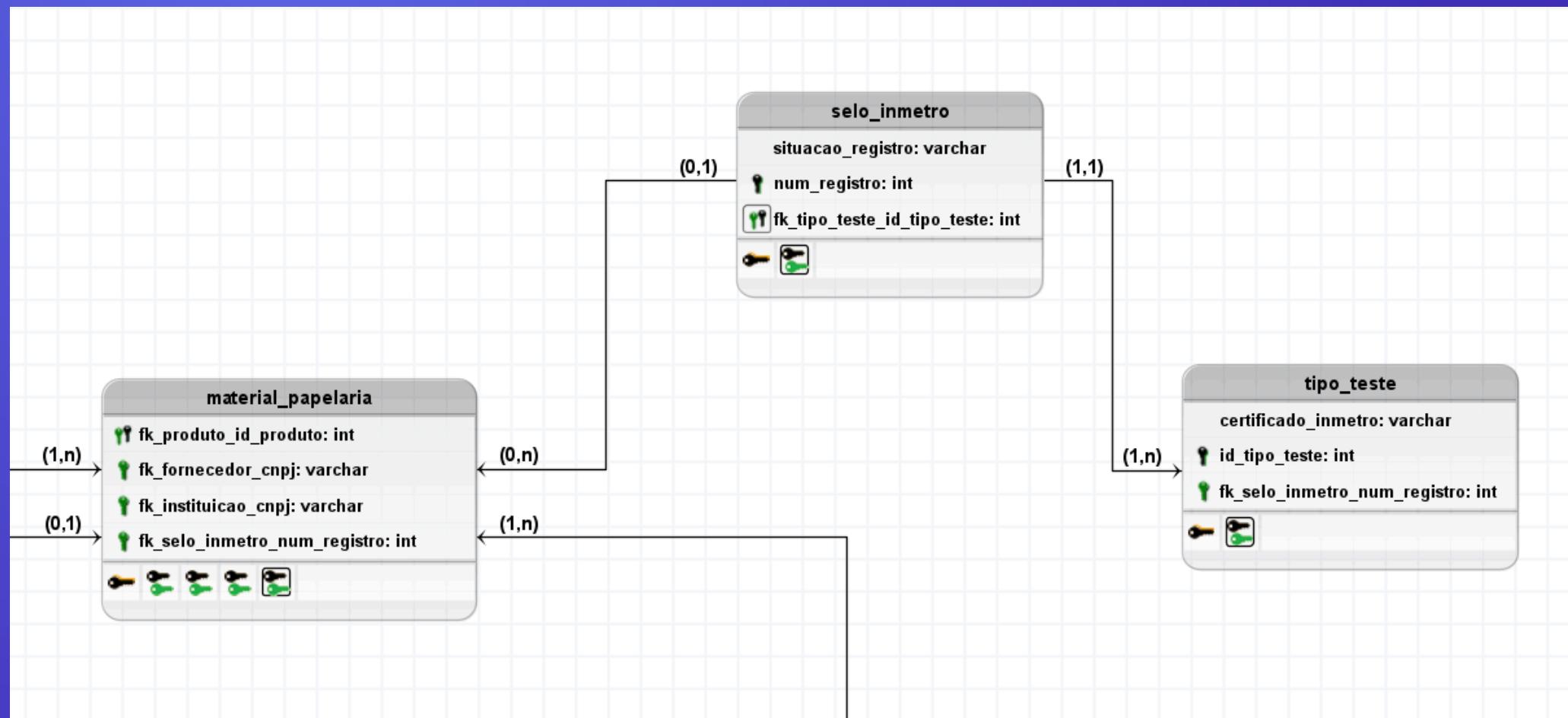
num_registro_inmetro	situacao_registro	certificado_inmetro	tipo_teste
678	ativo	conformidade	físico

3FN:

certificado_inmetro	tipo_teste
conformidade	físico
meio ambiente	químico

Normalização

Aplicação no Br Modelo e DDL



```
320 CREATE TABLE selo_inmetro (
321     situacao_registro varchar,
322     num_registro int PRIMARY KEY
323 );
```

```
330 CREATE TABLE tipo_teste (
331     certificado_inmetro varchar,
332     id_tipo_teste varchar PRIMARY KEY,
333     fk_selo_inmetro_num_registro int
334 );
```

Normalização

tabela aluno

id_matricula	id_matriz_escolar	série	turma	turno	mensalidade	qtd_faltas	qtd_livros_comprados	saldo_cantina
202401	00	1ª serie	A	manhã	1500	2	5	10
202402	02	3º ano	B	tarde	2000	7	2	10

2FN:

id_matriz_escolar	série	turma
00	1ª serie	A
02	3º ano	B

3FN:

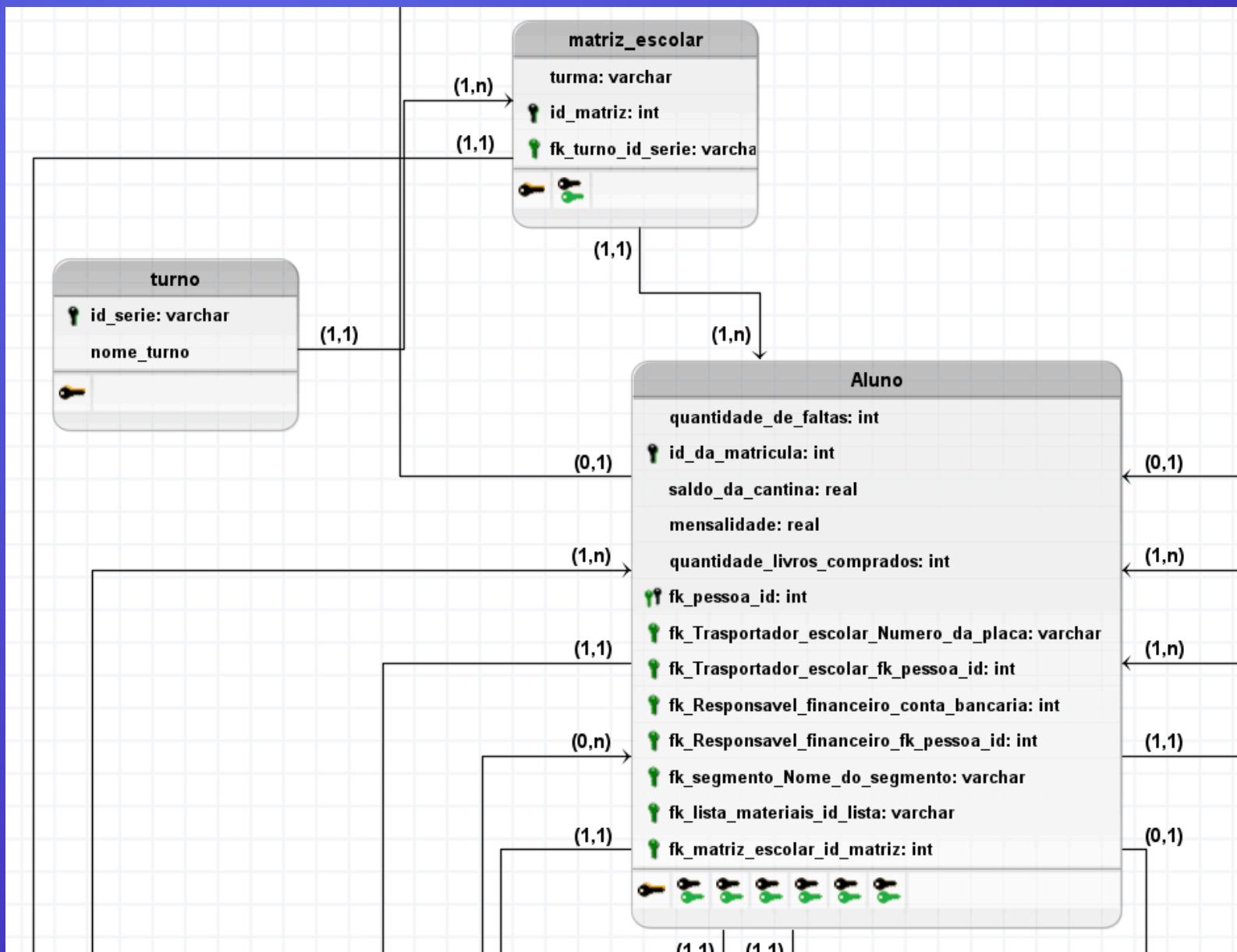
serie	turno
1ª serie	manhã
3º ano	tarde



Normalização



Aplicação no Br Modelo e DDL



```
325 CREATE TABLE turno (
326     id_serie varchar PRIMARY KEY,
327     nome_turno varchar
328 );
```

```
298 CREATE TABLE matriz_escolar (
299     turma varchar,
300     id_matriz int PRIMARY KEY,
301     fk_turno_id_serie varchar
302 );
```

CRUD



Framework



ORM



Banco utilizado



Tabelas utilizadas para o CRUD



AUTOR

INSTITUIÇÃO

SERVIÇOS

FUNCIONÁRIO

EDITORIA

USUÁRIOS

Obrigado!

Equipe:

Gabriel Torres

Isadora Xavier

Marcelo Henrique

Maria Eduarda Santana