1



Sumário

**I** **PARTE 1: MODELAGEM DE DADOS (RA1)**

1. **INFORMAÇÕES GERAIS DO PROJETO . . . . . . . . . . . . . . . . . . . 4**
   1. **Título do Sistema. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** **. 4**
   2. **Integrantes da Equipe . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4**
2. **DESCRIÇÃO DO MODELO DE NEGÓCIO . . . . . . . . . . . . . . . . . .5**
   1. **Itens do Modelo de Negócio . . . . . . . . . . . . . . . . . . . . . . . . .** . **5**
3. **JUSTIFICATIVAS TÉCNICAS DA MODELAGEM . . . . . . . . . . . . . 8**
   1. **Relacionamentos . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6**
   2. **Chaves Primárias (PK)** **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6**
   3. **Chaves Estrangeiras (FK) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6**
   4. **Atributos e Domínios** **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6**
   5. **Normalização . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** **7**
   6. **Especializações/Generalizações (Herança**) **. . . . . . . . . . . . . . .7**
   7. **Decisões de Desempenho e Escalabilidade . . . . . . . . . . . . . . . .7**
   8. **Alternativas Consideradas . . . . . . . . . . . . . . . . . . . . . . . . .** **7**
4. **MODELOS E SCRIPTS . . . . . . . . . . . . . . . . . . . . . . . . .8**
   1. **Modelo Conceitual – Diagrama Entidade-Relacionamento . . . . . . 8**
   2. **Modelo Lógico e Físico - Script SQL de Criação** . . . . . . . . . . . .**9**
   3. **Evidências Adicionais** . . . . . . . . . . . . . . . . . . . . . . . . . . . .**9**

*Sumário*

**II PARTE 2: IMPLEMENTAÇÃO E CONSULTAS (RA2)**

1. **IMPLEMENTAÇÃO E POPULAÇÃO DO BANCO . . . . . . . . . . 11**
   1. **Script SQL de Inserção de Dados (População)** . . . . . . . . . . . . .**11**
2. **SCRIPTS SQL DE CONSULTAS OBRIGATÓRIAS. . . . . . . . .12**
   1. **Consultas com JOIN (Mínimo 10)** . . . . . . . . . . . . . . . . . . . . **12**
   2. **Consultas com GROUP BY e HAVING (Mínimo 5)** . . . . . . . . . .**12**
   3. **Subqueries (Mínimo 5)** . . . . . . . . . . . . . . . . . . . . . . . . . .**12**
   4. **Criação de VIEWs (Mínimo 5)** . . . . . . . . . . . . . . . . . . . . . .**13**
   5. **UPDATEs e DELETEs Justificados** . . . . . . . . . . . . . . . . . . .**13**
      1. **UPDATEs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13**
      2. **DELETE** **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13**
3. **RELATÓRIO EXPLICATIVO DAS CONSULTAS PROPOSTAS PELO** **PROFESSOR** . . . . . . . . . . **14**
   1. **Ajustes e Melhorias na Modelagem (Revisão da Parte 1)**. . . . . .**14**

**III PARTE 3: CONTROLE DE ACESSO, TRANSAÇÕES E** **OTIMIZAÇÃO(RA3) 15**

1. **SEGURANÇA E CONTROLE DE ACESSO . . . . . . . . . . . . . .16**
   1. **Criação de Usuários e Perfis de Acesso (DCL)** . . . . . . . . . . . . .**16**
   2. **Simulação de Revogação de Acesso** . . . . . . . . . . . . . . . . . . .**16**
2. **CONTROLE DE TRANSAÇÕES (COMMIT/ROLLBACK) . . . . . 18**
   1. **Simulação de Transação de Venda (Sucesso)**. . . . . . . . . . . . .**18**
   2. **Simulação de Transação com Falha (Rollback)** . . . . . . . . . . . . .**18**
3. **OTIMIZAÇÃO DE DESEMPENHO E BOAS PRÁTICAS . . . . . . 20**
   1. **Análise de Índices** . . . . . . . . . . . . . . . . . . . . . . . . . . . . .**20**
   2. **Configuração de Backup e Log** . . . . . . . . . . . . . . . . . . . . . .**20**
   3. **Declaração de Autoria . . . . . . . . . . . . . . . . . . . . . . . . . .21**

Parte I

Parte 1: Modelagem de Dados (RA1)

# 1 Informações Gerais do Projeto

1.1 Título do Sistema

Sistema de gerenciamento de um studio de Pilates

## 1.2 Integrantes da Equipe

* Cauã Brylkowski dos Santos
* Clara Miyamoto Yajima
* Estevão Rickli de Carvalho
* Leonardo Simioni Torquato

# 2 Descrição do Modelo de Negócio

## 2.1 Itens do Modelo de Negócio

* **Identificação da Organização:**   
  Master Studio de Pilates - Ramo de atuação: Atividades Físicas, porte médio.
* **Objetivo Principal:**

O sistema tem como objetivo gerenciar as turmas, professores e os alunos de maneira mais prática e objetiva, além de facilitar atividades de pagamento e administração de planos.

* **Público-alvo e Usuários:**   
  Será utilizado por funcionários e administradores da corporação.
* **Processos Principais:**

Cadastro – O aluno realiza a matrícula no sistema do studio, informando CPF, telefone e dados de saúde relevantes para a prática das atividades.

Escolha do plano – No ato da matrícula, o aluno seleciona um plano (mensal, trimestral ou anual), que define o valor, com descontos para planos mais duradouros.

Agendamento de aulas – Após a escolha do plano, o aluno define seu horário de disponibilidade para as aulas.

Montagem de turma – Conforme o horário, o sistema te posiciona em uma turma. Se a escolhida estiver lotada, é necessária a escolha de outro horário.

Pagamento – O aluno escolherá uma forma de pagamento, podendo ser PIX, cartão de débito ou crédito, boleto, dinheiro. Também escolhe um dia para a recorrência do pagamento, ou seja, seu vencimento. Após pago, será liberado as próximas etapas de seu agendamento.

Atribuição do professor – Com base nesse horário, o sistema atribui automaticamente ao aluno a um professor responsável, garantindo que cada professor gerencie os alunos nos horários em que está disponível.

*Capítulo 2.* *Descrição do Modelo de Negócio*

* **Fluxo de Informações:**

Os dados serão separados em: informados pelos alunos durante a matrícula e atribuídos aos professores durante sua contratação. São utilizados dentro do banco de dados para   
medir informações inerentes aos alunos e professores, datas e turmas. São consultados por professores e funcionários que desejem verificar informações de turmas. Os dados de alunos são excluídos após 3 meses do cancelamento de sua matrícula, e os de professores após 1 ano de seu desligamento.

* **Necessidade do Banco de Dados:**

O banco de dados será útil para organizar turmas, facilitar o fluxo de alunos, otimizar processos e resolver conflitos de agendamento.

* **Regras de Negócio:**

As aulas podem ser canceladas com direito de reagendamento em até 48h precedentes. Após esse período, é possível cancelar, mas sem direito a reposição. O cancelamento da matrícula vai ser referente ao plano escolhido, seguindo ordens de contrato: plano mensal, sem fidelidade, sem descontos; Plano Trimestral, fidelidade de 3 meses, desconto de 10%; Plano Semestral, fidelidade de 6 meses, desconto de 15%, acesso ao espaço de conforto, com cadeiras de massagem. Tanto o plano Trimestral quanto o Semestral possuem multa se quebra de contrato.

* **Entidades e Relacionamentos-chave:**

PESSOA – é ou professor, ou aluno, ou funcionário  
PROFESSOR – é pessoa, tem N turmas   
FUNCIONÁRIO – é pessoa  
ALUNO – é pessoa, está em N turmas  
TURMA – formada por N alunos, gerenciada por 1 professor  
PLANO – atrelado a 1 pagamento e 1 matrícula  
MENSALIDADE– vinculado a 1 matrícula  
SALÁRIO - vinculado a 1 funcionário ou 1 professor  
MATRÍCULA – atrelado a 1 plano e 1 pagamento e 1 aluno  
ENDEREÇO – vinculado a 1 pessoa

*Capítulo 2.* *Descrição do Modelo de Negócio*

* **Requisitos Não Funcionais Relevantes:**É pedido ao aluno, ao professor, e ao funcionário consentimento para o armazenamento e uso de dados, assegurando a privacidade e uso não comercial, seguindo a LGPD.
* **Relatórios e Consultas Esperadas:**

- Aluno X está em Y horário?  
- Turma X está cheia?  
- Professor X está em que turma?  
- O pagamento de aluno X está confirmado?

# 3 Justificativas Técnicas da Modelagem

## 3.1 Relacionamentos

PESSOA – PROFESSOR (1:1)  
*Um professor só pode ter 1 CPF*

PESSOA – FUNCIONÁRIO (1:1)  
*Um funcionário só pode ter 1 CPF*

PESSOA – ALUNO (1:1)  
*Um aluno só pode ter 1 CPF*

*PESSOA – FAXINEIRO (1:1)*  
*Um faxineiro só pode ter um CPF*

*PESSOA – SECRETARIO (1:1)*  
*Um secretario só pode ter um CPF*

PROFESSOR – TURMA (1:N)  
*Um professor pode ter N turmas, e uma turma só pode ter 1 professor*

MATRÍCULA – TURMA (N:N)  
*Um aluno pode estar em várias turmas, e uma turma pode ter vários alunos*

PROFESSOR – SALÁRIO (1:1)  
*Um professor só pode ter um salário, e o salário está atribuído a um professor*

FUNCIONÁRIO – SALÁRIO (1:1)  
*Um funcionário só pode ter um salário, e o salário está atribuído a um funcionário*

ALUNO – MATRÍCULA (1:1)  
*Um aluno pode ter 1 matrícula, e uma matrícula pertence a 1 aluno*

PLANO – MENSALIDADE (1:1)  
*Um plano prove 1 mensalidade, e uma mensalidade vem de 1 plano*

MENSALIDADE – MATRÍCULA (1:1)  
*Uma mensalidade está ligada a 1 matrícula, e uma matrícula tem 1 mensalidade*

ENDEREÇO – PESSOA (1:1)  
*Uma pessoa tem 1 endereço, e um endereço refere a 1 pessoa.*

*Capítulo 3.*  *Justificativas Técnicas da Modelagem*

## 3.2 Chaves Primárias (PK)

1) TABELA pessoa (PK = CPF) — Natural key  
**Critério de escolha:** O CPF é único para cada indivíduo e cumpre perfeitamente o papel de identificação.  
**Por que não usar PK artificial (idPessoa)?**  
Seria redundante, pois o CPF já garante unicidade.

2) TABELA aluno (PK = cpf) — Natural key herdada de pessoa  
**Critério:** Aluno é uma especialização de pessoa → usa a mesma identificação.  
**Por que não usar idAluno artificial?** Iria duplicar identificação desnecessariamente.

### 3) TABELA funcionario (PK = cpf + idFuncionario UNIQUE) Aqui existe uma mistura. **PK escolhida:** CPF (natural) Usado para manter consistência com a tabela pessoa.

### **Chave artificial adicional:** idFuncionario Criada para facilitar relacionamento com subtipos (professor, secretario, faxineiro).

**Por que não usar só idFuncionario como PK?**  
Porque funcionário precisa estar vinculado diretamente à pessoa.

### 4) Tabelas de especialização (professor, secretario, faxineiro) **PK:** idProfessor, idSecretario, idFaxineiro (artificiais)

**Critério:** As subclasses precisam de identificadores próprios numéricos para controle interno da aplicação.

**Por que não usar CPF como PK?**  
Também funciona, mas a PK artificial deixa cada papel independente e facilita referenciar turmas, cargos, etc.

5) TABELA endereco (PK = cep) — Natural key  
**Critério:** CEP identifica unicamente um endereço.  
**Por que não usar idEndereco artificial?**CEP já é único e amplamente usado.

*Capítulo 3 Justificativas Técnicas da Modelagem*

6) plano (PK = idPlano) — Artificial  
**Critério**: Planos são entidades abstratas sem identificador natural.  
**Por que não usar nome do plano como PK?**Poderia haver planos com nomes parecidos ou modificados.

7) matrícula (PK = idMatricula) — Artificial  
**Critério:** Matrícula precisa de um identificador sequencial único.  
**Por que não usar CPF + data?**  
Esses dados podem mudar e não garantem unicidade.

8) mensalidade (PK = idMensalidade) — Artificial  
**Critério:** Cada mensalidade precisa de um ID próprio para controle financeiro.  
**Por que não usar idMatricula + data?**Seria difícil garantir que não existam duplicações mensais.

9) turma (PK = idTurma) — Artificial  
**Critério:** Cada turma deve ser identificada por um número único simples.  
**Por que não usar nome/horário do professor?**  
Essas informações mudam e podem se repetir.

## 10) turmaaluno (PK composta = idTurma + idMatricula) — Natural composta **Critério:** Essa tabela representa um relacionamento N:N → PK natural composta. **Por que não usar idTurmaAluno artificial?** A chave composta já define unicamente o aluno em uma turma.

## 11) salario (PK = idPagamento) — Artificial **Critério:** Cada pagamento é um registro único e sequencial. **Por que não usar CPF + mês?** Possível, mas impediria histórico de múltiplos pagamentos no mesmo mês.

*Capítulo 3.*  *Justificativas Técnicas da Modelagem*

## 3.3 Chaves Estrangeiras (FK)

Cada entidade foi incluída para:

* evitar redundância (pessoa, endereço)
* especializar categorias (aluno, funcionário, cargos)
* registrar eventos (matrícula, mensalidade, salário)
* representar relacionamentos N:N (turma aluno)
* organizar operações (turma, plano)
* Cada FK existe para:  
  - assegurar que um registro dependa de outro válido, evitando inconsistências.  
  - garantir integridade referencial, impedindo registros órfãos.

## 3.4 Atributos e Domínios

**Pessoa** - usado para relacionar os indivíduos do banco de dados – PK: CPF(char) (NOT NULL, UNIQUE), nome(varchar)(NOT NULL), fk:CEP, telefone(char), dataNascimento (date) email(varchar)

A PK de pessoa, CPF, é uma natural key, já que se trata de um valor já existente, facilitando a otimização por ser um dado já existente e não duplicável.

**Professor** – usado para administrar horários e turmas – PK:fk:CPF, especialização(varchar) (NOT NULL), FK:idPagamento, idProfessor (char) (NOT NULL, UNIQUE)

A PK de professor, CPF, é uma natural key importada de Pessoa, se trata de um valor já existente, facilitando a otimização por ser um dado já existente e não duplicável.

**Funcionário** – usado para controle dos trabalhadores – PK:fk:CPF, idFuncionário(char) (NOT NULL, UNIQUE), cargo(varchar) (NOT NULL), FK:idPagamento

A PK de funcionário, CPF, é uma natural key importada de Pessoa, se trata de um valor já existente, facilitando a otimização por ser um dado já existente e não duplicável.

**Faxineiro –** usado para organização de funcionários - PK:fk:CPF, idFaxineiro(INT), idFuncionario (INT).

**Secretario –** usado para organização de funcionários - PK:FK:cpf(char,NOT NULL), idSecretario(INT), idFuncionario (int,NOT NULL).

**Aluno** – uso para controle da quantidade de alunos, seus pagamentos e organização das turmas–PK:fk:CPF, fk:IDMatrícula, (varchar)dadosSaúde.

A PK de aluno, CPF, é uma natural key importada de Pessoa, se trata de um valor já existente, facilitando a otimização por ser um dado já existente e não duplicável.

**Turma** – usado para organização dos horários, da disponibilidade – PK:idTurma(char) (NOT NULL, UNIQUE), Horário (time)(NOT NULL), fk:idProfessor, nível(varchar)(iniciante, intermediário, avançado), capacidadeMax (int)(NOT NULL)

A PK de turma, idTurma, é uma surrogate key, já que se trata de um valor inventado, sendo útil para a organização e classificação separada de turmas.

**Plano** – usado para controle de benefícios e recorrência da mensalidade – PK:idPlano(int)(NOT NULL, UNIQUE), tipo(varchar) (mensal/trimestral/semestral) (NOT NULL), valor (int)(NOT NULL), validade(date), regrasCancelamento(varchar)(NOT NULL), benefícios(varchar).

A PK de turma, idPlano, é uma surrogate key, já que se trata de um valor inventado, sendo útil para a organização e classificação separada de planos.

**Mensalidade** – controle de pagamento da mensalidade do aluno –PK:idMensalidade (char)(NOT NULL, UNIQUE), dataRecorrência(int)(NOT NULL), statusMensalidade(varchar) (pago/em aberto/atrasado) (NOT NULL), formaPagamento(varchar) (pix, boleto, transação bancária, cartão de crédito ou débito), mensalidade(int)

A PK de mensalidade, idMensalidade, é uma surrogate key, já que se trata de um valor inventado, sendo útil para a organização e classificação do status de pagamento.

**Salário** – controle de pagamento de servidores – PK:idPagamento(char)(NOT NULL UNIQUE),

dataDepósito(int),(NOT NULL), statusSalario(varchar),(pago/não pago),(NOT NULL),

formaDepósito(varchar),(dinheiro, transação bancaria),(NOT NULL), pagamento(int)

A PK de salário, idSalario, é uma surrogate key, já que se trata de um valor inventado, sendo útil para a organização e classificação do status de pagamento.

**Matrícula** – controle de entrada e saída de alunos – PK:idMatrícula(char) (NOT NULL, UNIQUE), fk:idMensalidade, statusMatricula(varchar)(ativa, cancelada) (NOT NULL), fk:idPlano (int) (NOT NULL)

A PK de matrícula, idMatricula, é uma surrogate key, já que se trata de um valor inventado, sendo útil para a organização e armazenamento de dados e preferências do aluno.

**Endereço** – **detalhamento da residência**– PK:CEP(char)(NOT NULL, UNIQUE), cidade(varchar),

bairro(varchar), rua(varchar), número(varchar), complemento(varchar)

A PK de endereço, CEP, é uma natural key, já se trata de um valor já existente, facilitando a otimização por ser um dado já existente e não duplicável.

TurmaAluno - usado para resolver o problema de exibir alunos pertencentes a uma turma seguindo o 1FN- Pk:fk:idTurma (char) e fk:idMatricula (char)..

*Capítulo 3.* *Justificativas Técnicas da Modelagem*

## 3.5 Normalização

Primeiramente aplicamos a 1FN, evitando atributos multivalorados, sem usar atributos compostos. Na 2FN, não fizemos dependência parcial, onde o atributo só depende de uma parte da PK, como na tabela turmaAluno em que nenhum atributo depende de uma só parte. E na 3FN tomamos cuidado para não haver dependência transitiva (um atributo depender de outro atributo não chave, como na tabela Pessoa que o CEP depende do CPF, ao invés do endereço

Utilizamos desnormalização proposital na tabela secretario, faxineiro, e professor, o idFuncionario nessas tabelas a fim de que em consultas mais simples não seja necessário utilizar joins em outras tabelas.

## 3.6 Especializações/Generalizações (Herança)

Para especializações, foi separado pessoa, em que pode ser aluno ou funcionário, a fim de otimizar o uso da chave primária CPF, com cada um tendo um atributo específico em sua sub-classe.

## 3.7 Decisões de Desempenho e Escalabilidade

Para a possibilidade de escalabilidade futura, é implementada as chaves de idMatricula, idFuncionario, idProfessor, assim possibilitando a facilidade de inscrição de novos cadastros facilmente distinguíveis por ordem crescente de ingresso no banco de dados. A otimização do sistema foi feita por meio da separação de diversas entidades e chaves que comunicam entre si por meio das chaves entrangeiras.

3.8 Alternativas Consideradas

A primeira ideia de modelagem escolhida envolvia um sistema de gerenciamento de vendas e estoque de uma indústria de derivados de milho e mandioca, previamente descartada por envolver casos muito amplos e dificilmente classificados, além da falta de conhecimento do processo geral de estoque.

# 4 Modelos e Scripts

## 4.1 Modelo Conceitual – Diagrama Entidade-Relacionamento

## 

## 

*Capítulo 4. Modelo e Scripts*

## 4.2 Modelo Lógico e Físico - Script SQL de Criação

Inclua o modelo lógico (tabelas, atributos, tipos de dados, restrições) e o script SQL de criação das tabelas.

**MODELO LÓGICO**

A computer screen shot of a diagram

AI-generated content may be incorrect.

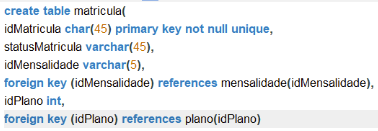
*Capítulo 4.*  *Modelo e Scripts*

**MODELO FÍSICO**

A screenshot of a computer code

AI-generated content may be incorrect.A close-up of a computer screen

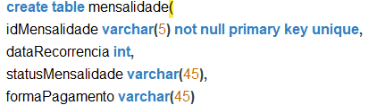
AI-generated content may be incorrect.

 A white background with blue text

AI-generated content may be incorrect.

A white background with blue text

AI-generated content may be incorrect.

A close-up of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.

*Capítulo 4.*  *Modelo e Scripts*

## 4.3 Evidências Adicionais

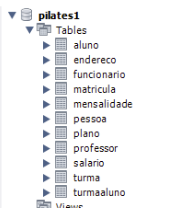
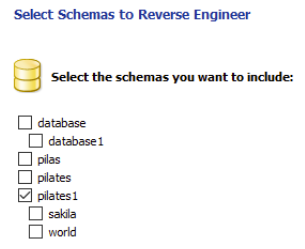
Insira diagramas, prints de telas, scripts adicionais ou complementares que foram utilizados no MySQL Workbench.

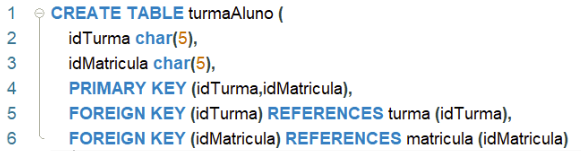










Parte II

Parte 2: Implementação e Consultas (RA2)

# 5 Implementação e População do Banco

## 5.1 Script SQL de Inserção de Dados (População)

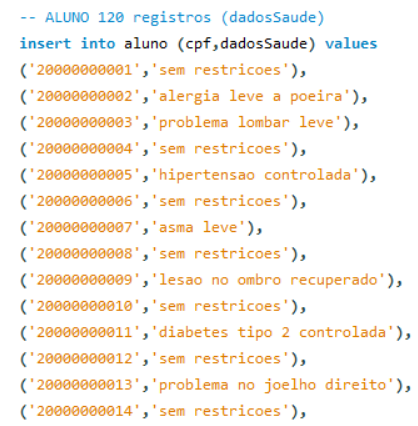
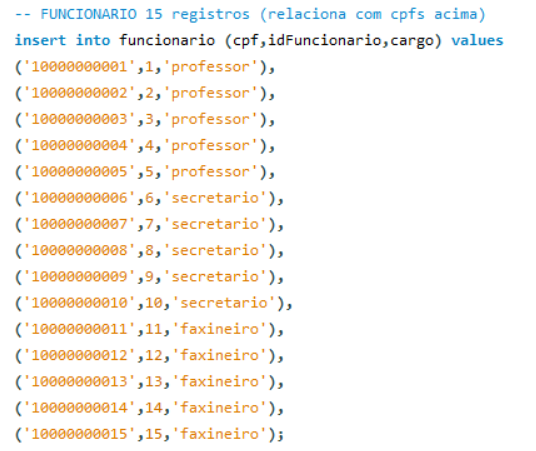
A implementação deve conter ao menos \*\*20 registros por tabela\*\*.

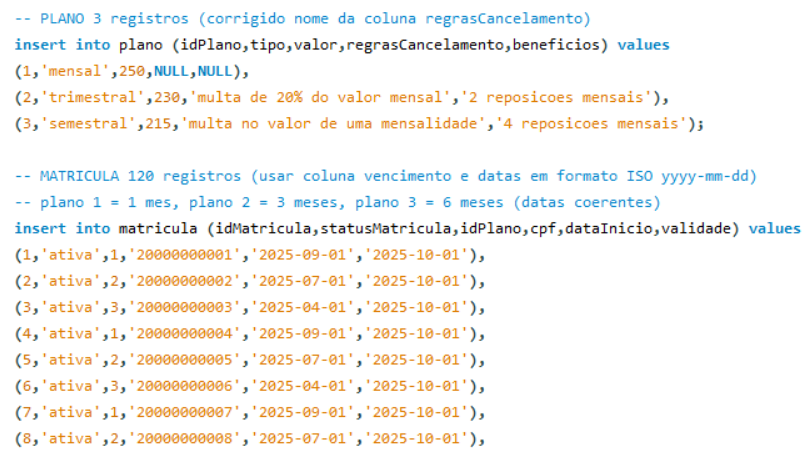
A screenshot of a computer code

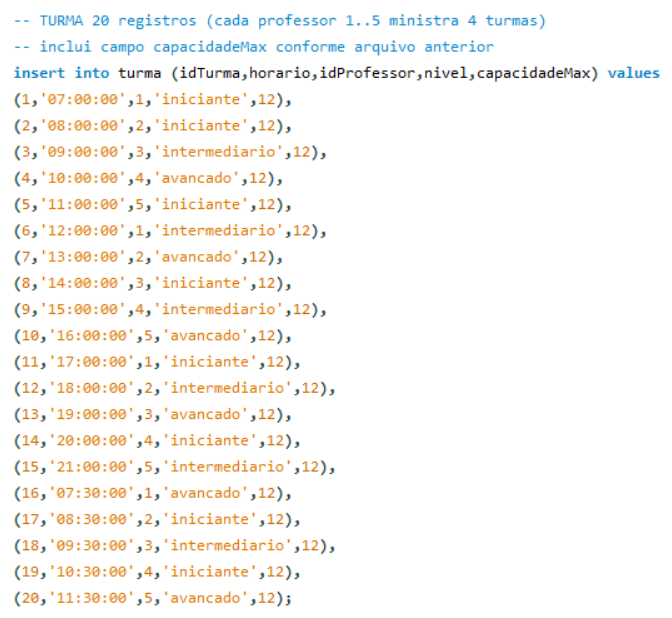
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.







A screenshot of a computer code

AI-generated content may be incorrect.

A white and orange numbers on a blue background

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.

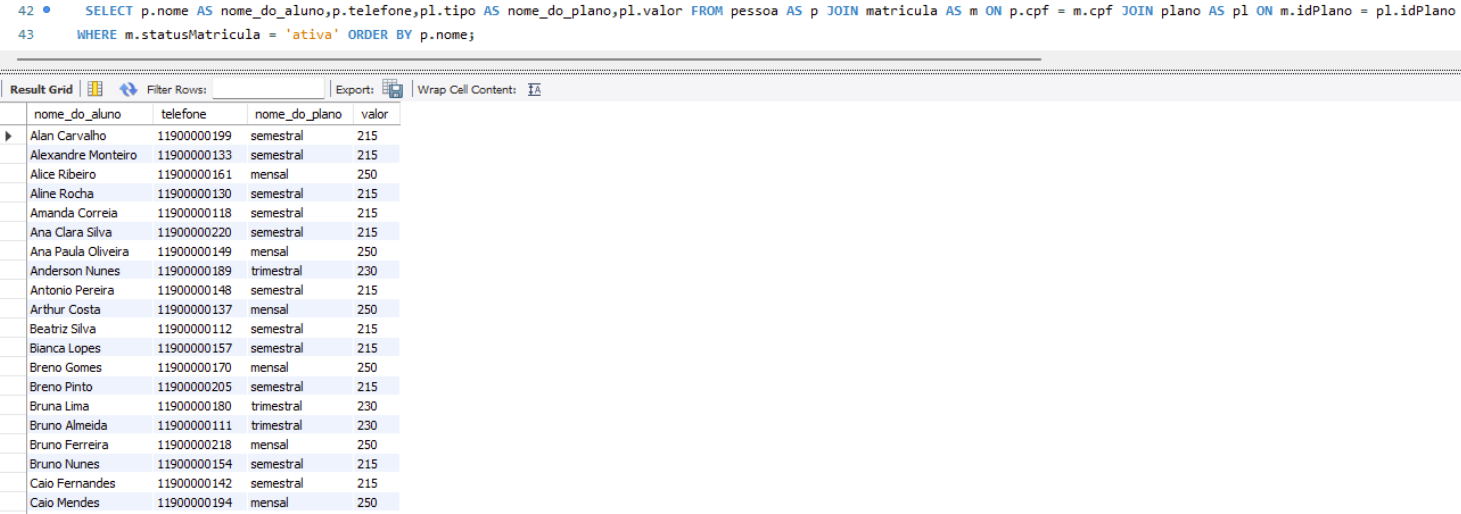
# 6 Scripts SQL de Consultas Obrigatórias

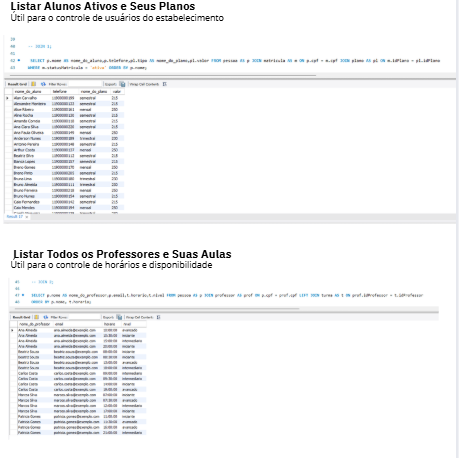
## 6.1 Consultas com JOIN (Mínimo 10)

Incluir 10 consultas com JOIN, sendo que \*\*pelo menos uma devem envolver 3 ou mais tabelas\*\*.

A screenshot of a computer

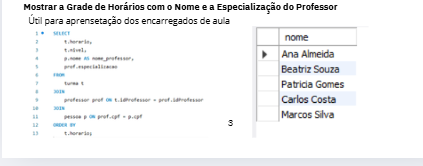
AI-generated content may be incorrect.

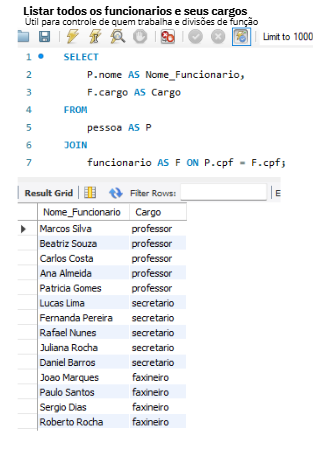


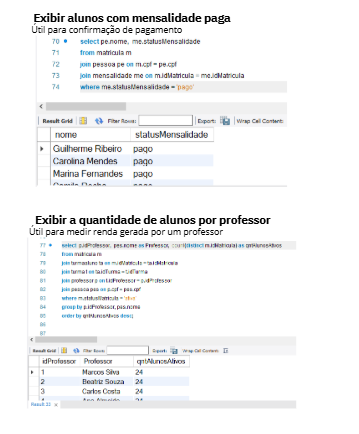


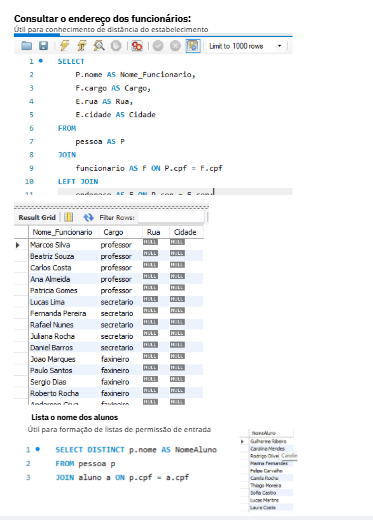






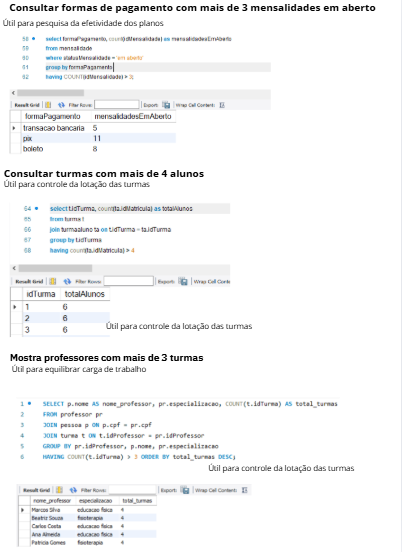






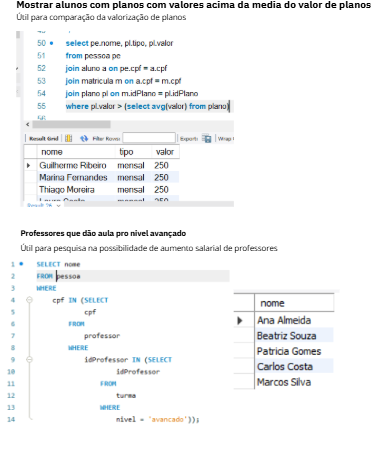
6.2 Consultas com GROUP BY e HAVING (Mínimo 5)  
Incluir 5 consultas com GROUP BY e HAVING.





6.3 Subqueries (Mínimo 5)  
Incluir 5 consultas utilizando SUBQUERY.





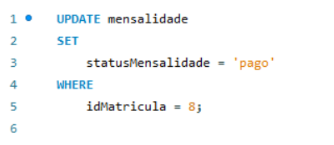
*Capítulo 6. Scripts SQL de Consultas Obrigatórias*

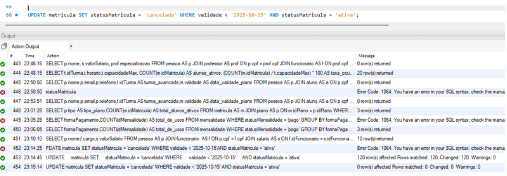
## 6.4 Criação de VIEWs (Mínimo 5)

Incluir 5 criações de VIEW útil para o contexto do negócio.



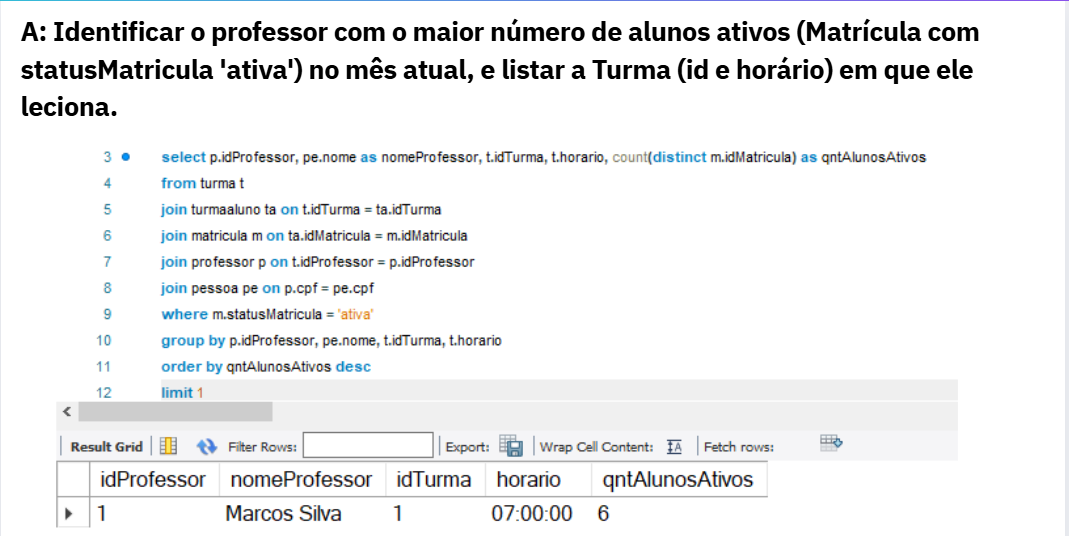
6.5 UPDATEs e DELETEs Justificados  
Incluir \*\*3 UPDATEs e 1 DELETE com cláusulas bem justificadas\*\*.

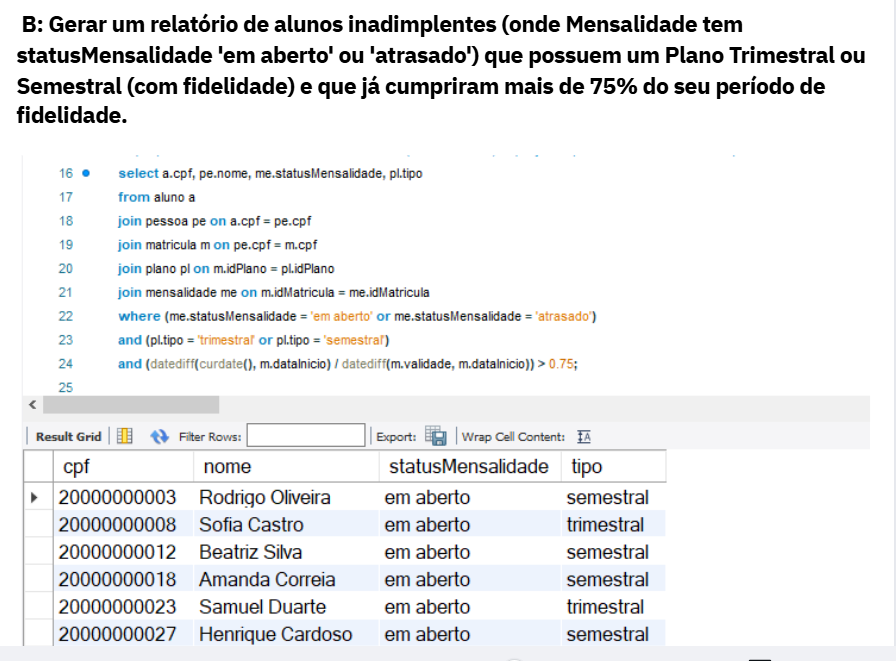
**6.5.1 UPDATEs**   
  
**Justificativa de Negócio:** Corrigir o Status de uma Mensalidade Específica

  
**Justificativa de Negócio:** Manutenção de dados para inativar as matrículas que já expiraram. Qualquer matrícula que ainda esteja com o status 'ativa', mas cuja data de validade já passou, deve ser atualizada para o status 'cancelada'. Isso garante que apenas alunos com planos válidos sejam considerados ativos no sistema.  
  
  
**Justificativa de Negócio:** Neste UPDATE há um aumento de 5% ao valor atual do salário de todos os funcionários do estabelecimento.  
  
6.5.2 DELETE  
  
**Justificativa de Negócio:** Deletar todos os registros da tabela mensalidade que estão com o status 'em aberto' ou 'atrasada' e que pertencem a matrículas que foram 'canceladas’

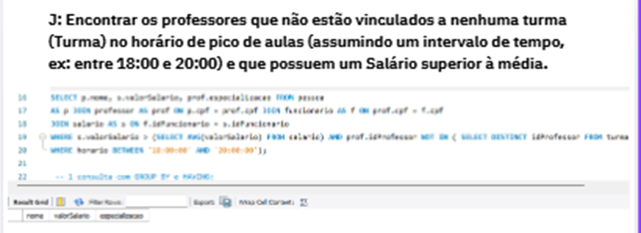
# 7 Relatório Explicativo das Consultas Propostas pelo Professor

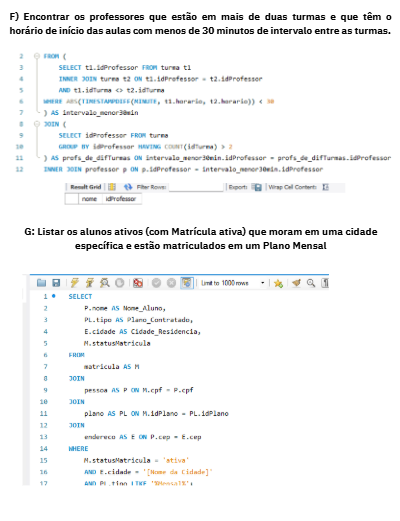
Apresentar a execução das \*\*10 consultas propostas pelo+ professor\*\* e as capturas de tela (prints) de cada uma.

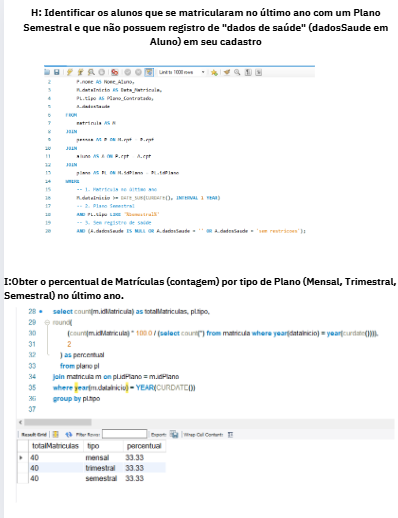












## 7. Ajustes e Melhorias na Modelagem (Revisão da Parte 1)

Detalhar e justificar os ajustes e melhorias feitos no modelo em relação à Parte 1.

**Para a melhoria do projeto em relação à versão anteriormente entregue, o grupo realizou os seguintes ajustes:**

* Nas relações de pertencimento entre Id’s e tabelas (faz as relações entre foreign keys fazerem sentido e as consultas funcionarem);
* Modificações em tipos de dados - substituição de diversos campos que apresentavam varchar por enum, e de campos que apresentavam char para int (torna a inserção e manipulação de dados mais viável);
* Adição das tabelas Faxineiro e Secretario, a fim de expandir o escopo do projeto com complemento de mais tipos de funcionário.

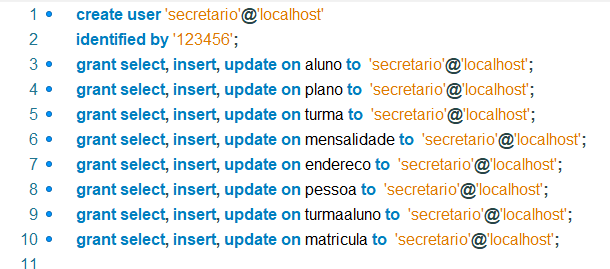
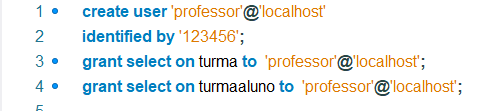
Parte III

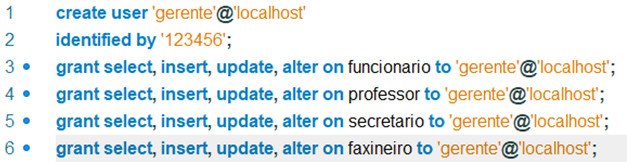
Parte 3: Controle de Acesso, Transações e

Otimização (RA3)

# 8 Segurança e Controle de Acesso O objetivo é implementar a segurança baseada em perfis e o princípio do privilégio mínimo.

## 8.1 Criação de Usuários e Perfis de Acesso (DCL) Implemente, via script SQL (DCL), a criação de pelo menos 3 perfis de acesso diferentes no banco, justificando os privilégios concedidos a cada um.

**Perfil 1: Secretario**  
 **Justificativa:** Permite alterar dados relacionados a alunos, possibilitando a organização eficiente de dados dos clientes.  
  
**Perfil 2: Professor** **Justificativa:** Permite ao professor acessar suas turmas e horários.

**Perfil 3: Gerente  
  
Justificativa:** Permite ao gerente controle em dados sobre seus funcionários, mas ainda segue o princípio do privilégio mínimo e não concede ALL.

## *Capítulo 10.* *Otimização de Desempenho e Boas Práticas*

## 8.2 Simulação de Revogação de Acesso

Simule a revogação de um privilégio importante de um dos perfis, demonstrando o script REVOKE e explicando o cenário de segurança que justifica a ação.

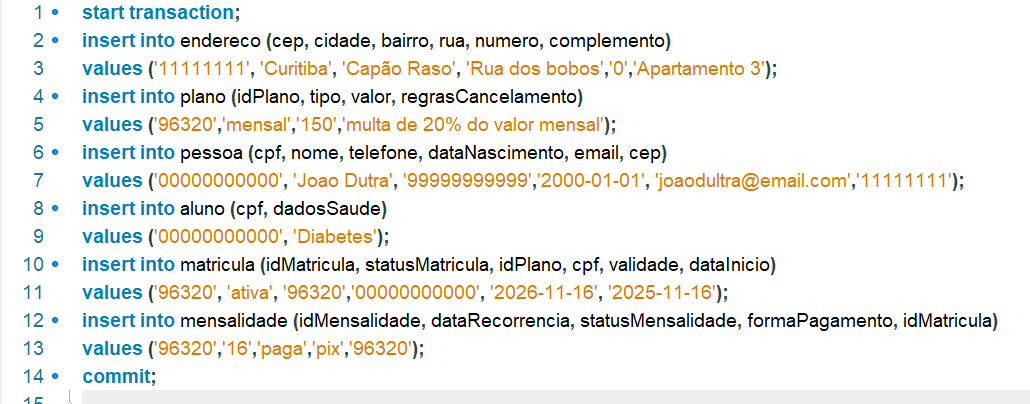


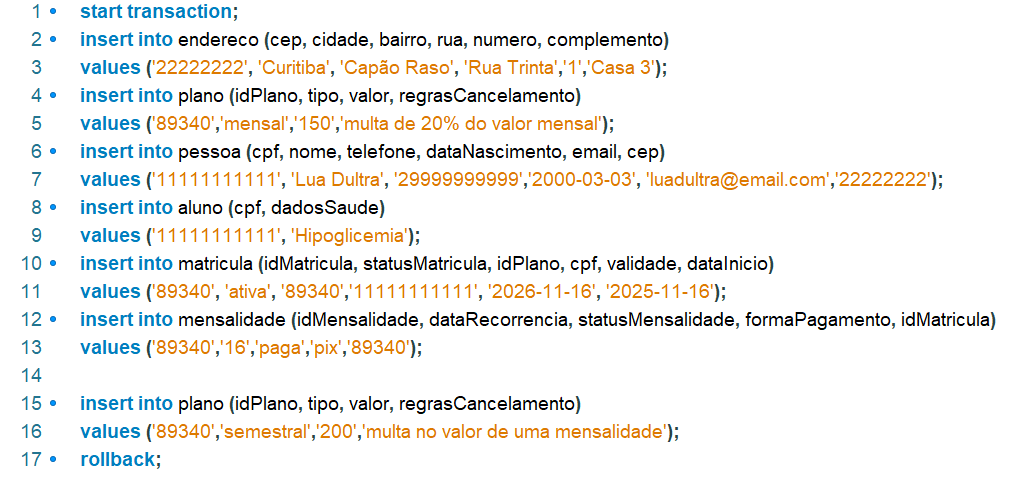
**Justificativa de segurança:**

Remove a possibilidade de um secretário adulterar dados de pagamento sobre um aluno, como fraudes de status de pagamento.

# 9 Controle de Transações (Commit/Rollback) O objetivo é garantir a atomicidade (A) e a consistência (C) dos dados.

9.1 Simulação de Transação de Venda (Sucesso)  
 Implemente um bloco de transação para simular um processo de negócio crítico (ex.:  
uma venda completa que envolve a inserção de um pedido e a atualização do estoque).  
Use START TRANSACTION e finalize com COMMIT.

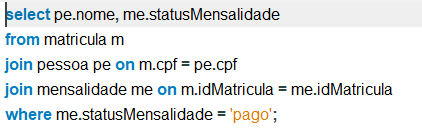


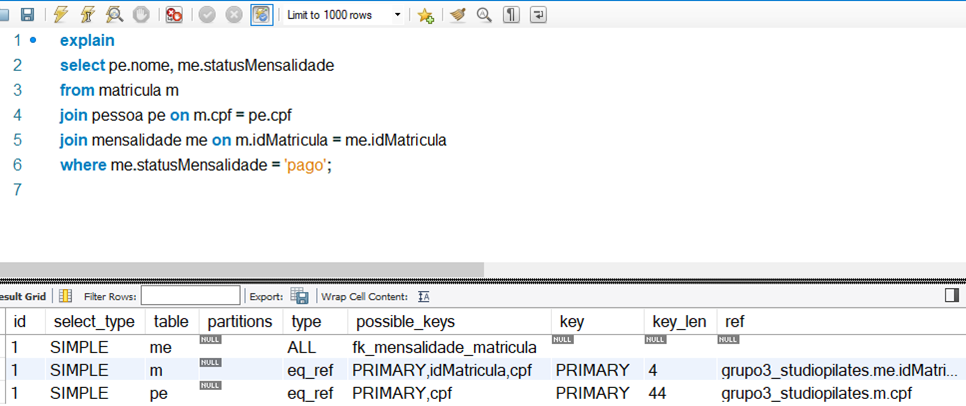
**Justificativa:** Para o sistema, os passos devem ser atômicos pois os dados de cadastro de um aluno ou funcionário dependem de chaves estrangeiras espalhadas em diversas tabelas. Sendo assim, se qualquer parte do processo falhar, é necessário cancelar toda a operação, a fim de evitar cadastros incompletos e dados órfãos no banco.  
  
9.2 Simulação de Transação com Falha (Rollback)  
Simule a mesma transação crítica, mas adicione uma condição de falha intencional  
(ex.: violação de restrição CHECK ou ERROR) e use o ROLLBACK para garantir que nenhuma  
alteração parcial seja persistida.  
  
  
**Justificativa:** O Rollback impede que sejam adicionadas 2 tabelas de plano ao mesmo cliente.

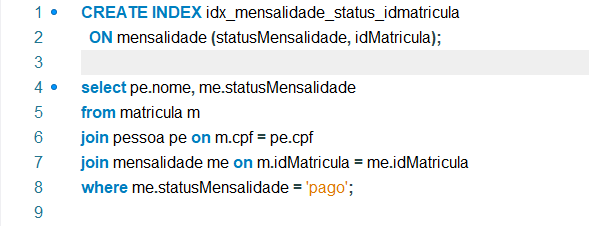
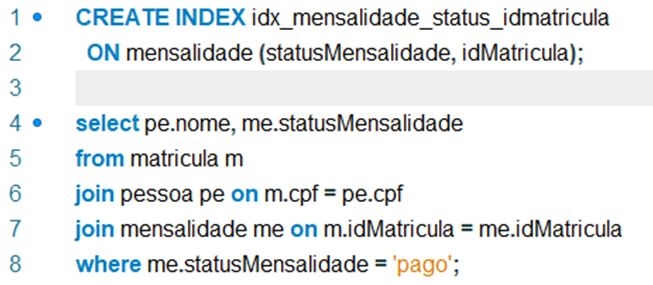
# 10 Otimização de Desempenho e Boas Práticas O objetivo é propor melhorias para a escalabilidade e velocidade do banco.

## 10.1 Análise de Índices

Escolha uma consulta complexa e utilize o comando EXPLAIN para analisar seu plano de execução

* **Consulta Escolhida:**
* **Plano de Execução (EXPLAIN):**



* **Proposta de Otimização:**  
  ****

O índice criado organiza a tabela mensalidade ordenando-a pelas colunas statusMensalidade e idMatricula, permitindo que o banco encontre os registros com statusMensalidade = ‘pago’ sem percorrer a tabela inteira. Como também possui o idMatricula, executa o join de matrícula de forma mais eficiente, reduzindo o número de leituras necessárias e acelerando o filtro.

10.2 Configuração de Backup e Log

**Frequência de Backup:**   
  
Semanal, devido ao fluxo de alunos e turmas não estar em constante mudança.

**Tipo de Backup:**   
  
Incremental, copia apenas o que mudou desde último salvamento, assim economizando memória e alterando apenas dados e horários novos.  
  
**Política de Logs:**   
  
É feita a ativação dos logs de transação, que registram todas as alterações feitas no sistema (cadastro de alunos, pagamentos, criação de turmas), assim possibilitando a realização de Point-In-Time-Recovery, onde no caso de alguma exclusão ou alteração acidental de dados é possível restaurar o banco e reaplicar os logs até o momento anterior ao erro. Desta forma, o studio possui copias completas semanais e possibilita a restauração em caso de erros.

Declaração de Autoria

Declaramos que este trabalho foi integralmente desenvolvido por nossa equipe.

|  |  |
| --- | --- |
| **Integrante 1:** Cauã Brylkowski dos Santos | **Assinatura: Cauã Brylkowski** |
| **Integrante 2:** Clara Miyamoto Yajima | **Assinatura: Clara Miyamoto** |
| **Integrante 3:** Leonardo Simioni Torquato | **Assinatura: Leonardo Simioni** |
| **Integrante 4:** Estevão Rickli de Carvalho | **Assinatura: Estevão Rickli** |