Projeto de Banco de Dados - Sistema de Bibliotecas

Discente: Allan Silva Borges

Docente: Robespierre Pita

Disciplina: MATA60 - Banco de Dados

Link do projeto no GitHub:
https://github.com/of-allan-borges/Projeto-Individual---Banco-de-dados/tree/main

Link do projeto no Google Drive:

https://drive.google.com/drive/folders/1mdNIKGGG3l74hJam3ym54_Z4asWD5CRA?usp=sharing

---

1. Contexto e Problemática

Marcelo e Ana, um casal apaixonado por livros, gerenciam três bibliotecas localizadas em diferentes pontos da cidade. Apesar do amor pela literatura, eles enfrentam alguns desafios, como a organização e o rastreamento dos livros. Durante anos, a localização dos livros, registros de empréstimos e reservas eram anotados manualmente em blocos de notas, no início até que estava funcionando, quando o negócio era pequeno, mas ao longo do tempo essa abordagem foi se tornando ineficiente, e acabou resultando em dificuldades como:

- Localizar livros específicos e identificar em qual biblioteca e local estavam armazenados.
- Lentidão na verificação da disponibilidade de livros para empréstimos ou reservas.
- Falta de um histórico consolidado sobre reservas e empréstimos, dificultando a análise do uso das bibliotecas.

Para resolver esses problemas, Marcelo e Ana decidiram implementar um sistema de informações baseado em um banco de dados relacional que centralize todas as informações relevantes.

1.2 Sobre o Sistema de Informações (SI)

O sistema desenvolvido tem como objetivo centralizar e automatizar o gerenciamento das três bibliotecas, abrangendo as seguintes funcionalidades principais:

a. Rastreamento de Livros:

- Localizar rapidamente qualquer livro, identificando sua biblioteca e local (ex.: prateleira e andar).
- Verificar a disponibilidade de exemplares para empréstimos e reservas.

b. Gerenciamento de Empréstimos e Reservas:

- Registrar empréstimos e devoluções com históricos detalhados.
- Controlar reservas, incluindo prazos de validade e status.

c. Organização de Gêneros e Autores:

- Associar livros a autores e gêneros.
- Manter catálogos atualizados para consultas.

d. Gestão de Multas:

- Calcular e registrar multas por devolução atrasada.

e. Relatórios e Dashboards:

- Exibir relatórios sobre o uso das bibliotecas, como empréstimos mais frequentes, livros mais reservados e performance dos funcionários.

1.3 Delimitação do Mini-Mundo para o Banco de Dados

Para atender às necessidades do casal, o banco de dados conta com as seguintes entidades e relacionamentos:

Entidades:

Autor

- **Descrição:** Representa os autores dos livros presentes na biblioteca.
- **Atributos:**
    - **PK_id_autor:** Identificador único do autor. (inteiro, incremental, PK)
    - **nome_autor:** Nome completo do autor. (string, 100 caracteres)
    - **biografia:** Breve biografia do autor. (texto)
    - **nacionalidade:** Nacionalidade do autor. (string, 50 caracteres)
    - **data_nascimento:** Data de nascimento do autor. (data)

Biblioteca

1. **Descrição:** Representa as bibliotecas onde os livros são armazenados e emprestados.
2. **Atributos:**
    - **PK_id_biblioteca:** Identificador único da biblioteca. (inteiro, incremental, PK)
    - **nome_biblioteca:** Nome da biblioteca. (string, 100 caracteres)
    - **endereco:** Endereço completo da biblioteca. (texto)
    - **telefone:** Telefone da biblioteca. (string, 15 caracteres)
    - **horario_abertura:** Horário de abertura da biblioteca. (hora)
    - **horario_fechamento:** Horário de fechamento da biblioteca. (hora)
    - **quantidade_funcionarios:** Número de funcionários da biblioteca. (inteiro)

Empréstimo

- **Descrição:** Representa a ação de emprestar um livro a um usuário.
- **Atributos:**
    - **PK_id_emprestimo:** Identificador único do empréstimo. (inteiro, incremental, PK)
    - **id_usuario:** Referência ao usuário que realizou o empréstimo. (inteiro, FK para tabela usuário)
    - **id_livro:** Referência ao livro emprestado. (inteiro, FK para tabela livro)
    - **data_emprestimo:** Data em que o livro foi emprestado. (data)

    - **data_devolucao:** Data prevista para a devolução do livro. (data)
    - **status:** Status atual do empréstimo (Em aberto, Devolvido, Extraviado). (string, 20 caracteres)
    - **tipo_emprestimo:** Tipo de empréstimo (Normal, Renovado, Emergencial). (string, 50 caracteres)
    - **id_funcionario_responsavel:** Referência ao funcionário que autorizou o empréstimo. (inteiro, FK para tabela funcionario)

Funcionário

- **Descrição:** Representa os funcionários da biblioteca.
- **Atributos:**
    - **PK_id_funcionario:** Identificador único do funcionário. (inteiro, incremental, PK)
    - **nome:** Nome completo do funcionário. (string, 100 caracteres)
    - **cargo:** Cargo do funcionário. (string, 50 caracteres)
    - **data_contratacao:** Data de contratação do funcionário. (data)
    - **salario:** Salário do funcionário. (decimal, 10.2)
    - **email:** Endereço de e-mail do funcionário. (string, 100 caracteres, único)
    - **telefone:** Telefone do funcionário. (string, 15 caracteres)
    - **id_biblioteca:** Referência à biblioteca em que o funcionário trabalha. (inteiro, FK para tabela biblioteca)

Gênero

- **Descrição:** Representa os gêneros literários dos livros.
- **Atributos:**
    - **PK_id_genero:** Identificador único do gênero. (inteiro, incremental, PK)
    - **nome_genero:** Nome do gênero literário. (string, 50 caracteres)
    - **tipo_genero:** Tipo de gênero (Ficção, Não-ficção, Infantil). (string, 50 caracteres)
    - **data_criacao:** Data de criação do registro do gênero. (data)

Livro_Autor

- **Descrição:** Representa a relação muitos-para-muitos entre livros e autores.
- Atributos:

    - **id_livro:** Referência ao livro. (inteiro, FK para tabela livro, parte da chave primária composta)
    - **id_autor:** Referência ao autor. (inteiro, FK para tabela autor, parte da chave primária composta)
    - **Chave Primária:** (id_livro, id_autor)

Livro

- **Descrição:** Representa os livros presentes na biblioteca.
- **Atributos:**
    - **PK_id_livro:** Identificador único do livro. (inteiro, incremental, PK)
    - **titulo:** Título do livro. (string, 200 caracteres)
    - **editora:** Editora responsável pela publicação do livro. (string, 100 caracteres)
    - **data_publicacao:** Data de publicação do livro. (data)
    - **quantidade_disponivel:** Quantidade de exemplares disponíveis do livro. (inteiro)
    - **id_local:** Referência ao local onde o livro está armazenado. (inteiro, FK para tabela local)
    - **quantidade_total:** Quantidade total de exemplares do livro. (inteiro)

Local

- **Descrição:** Representa os locais de armazenamento dos livros na biblioteca (estantes, prateleiras, etc.).
- **Atributos:**
    - **PK_id_local:** Identificador único do local. (inteiro, incremental, PK)
    - **descricao:** Descrição detalhada do local. (texto)
    - **andar:** Andar onde o local se encontra. (inteiro)
    - **prateleira:** Identificação da prateleira. (string, 20 caracteres)
    - **tipo_local:** Tipo de local (estante, caixa, depósito). (string, 50 caracteres)
    - **capacidade:** Capacidade máxima de livros do local. (inteiro)
    - **id_biblioteca:** Referência a biblioteca. (inteiro, FK para tabela biblioteca)

Multa

- **Descrição:** Representa as multas aplicadas aos usuários por atrasos na devolução de livros.
- **Atributos:**

- **PK_id_multa:** Identificador único da multa. (inteiro, incremental, PK)
- **id_funcionario_responsavel:** Referência ao funcionário que aplicou a multa. (inteiro, FK para tabela funcionario)
- **id_usuario:** Referência ao usuário que recebeu a multa. (inteiro, FK para tabela usuario)
- **valor:** Valor da multa. (decimal, 10.2)
- **data_multa:** Data em que a multa foi aplicada. (data)
- **status_multa:** Status da multa (Aberta, Paga). (string, 20 caracteres)
- **descricao:** Descrição da razão da multa. (texto)
- **data_pagamento:** Data em que a multa foi paga. (data, nulo por padrão)
- **id_emprestimo:** Referência ao empréstimo relacionado à multa. (inteiro, FK para tabela emprestimo, nulo se a multa não estiver relacionada a um empréstimo específico)

Reserva

- **Descrição:** Representa as reservas de livros feitas pelos usuários.
- **Atributos:**
  - **PK_id_reserva:** Identificador único da reserva. (inteiro, incremental, PK)
  - **id_usuario:** Referência ao usuário que realizou a reserva. (inteiro, FK para tabela usuario)
  - **id_livro:** Referência ao livro reservado. (inteiro, FK para tabela livro)
  - **data_reserva:** Data em que a reserva foi feita. (data)
  - **status_reserva:** Status da reserva (Pendente, Confirmada, Cancelada). (string, 20 caracteres)
  - **data_validade:** Data de validade da reserva. (data)
  - **tipo_reserva:** Tipo de reserva (Normal, Emergencial). (string, 50 caracteres)
  - **id_funcionario_responsavel:** Referência ao funcionário que processou a reserva. (inteiro, FK para tabela funcionario)

Usuário

- **Descrição:** Representa os usuários da biblioteca.
- **Atributos:**
  - **PK_id_usuario:** Identificador único do usuário. (inteiro, incremental, PK)
  - **nome:** Nome completo do usuário. (string, 100 caracteres)
  - **email:** Endereço de e-mail do usuário. (string, 100 caracteres, único)
  - **telefone:** Telefone do usuário. (string, 15 caracteres)

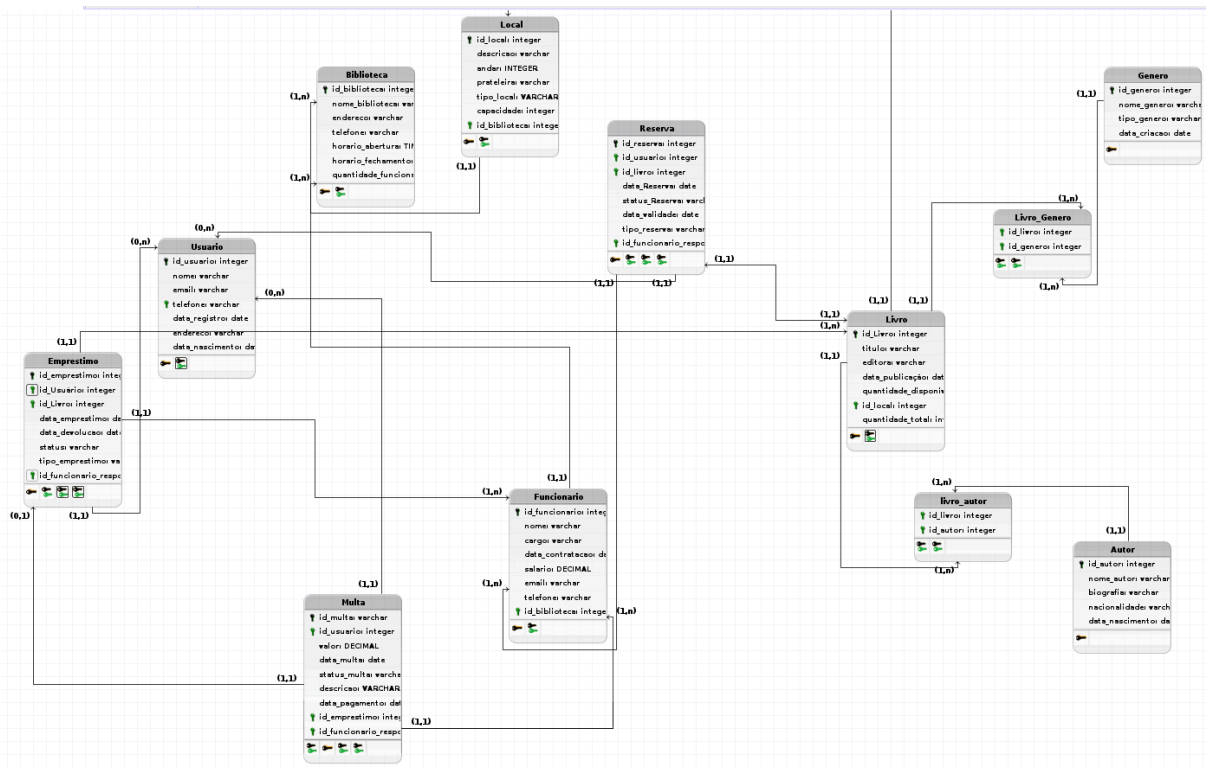
- **data_registro:** Data de registro do usuário. (data)
- **endereco:** Endereço completo do usuário. (texto)
- **data_nascimento:** Data de nascimento do usuário. (data)

---

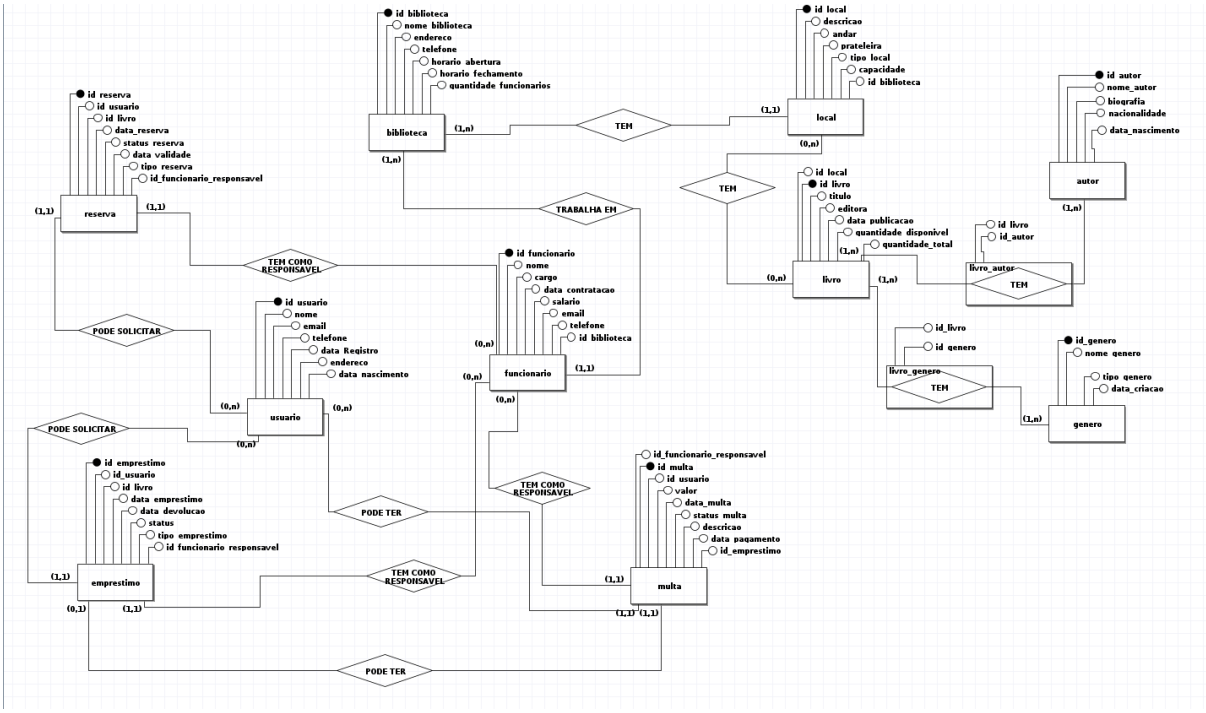
2. Modelo Conceitual e Lógico

Os modelos conceitual e lógico do banco de dados foram elaborados para garantir a integridade referencial e a normalização das tabelas, minimizando redundâncias e assegurando um bom desempenho nas consultas.

2.1. Modelo Lógico:



2.2. Modelo Conceitual:



---

3. Criação do banco de dados

O código SQL usado para a criação das tabelas do banco de dados se encontra neste link: https://github.com/of-allan-borges/Projeto-Individual---Banco-de-dados/blob/main/CriarTabelas/criacaoTabelas.sql

---

4. População de banco de dados:

O código SQL usado para gerar a população do banco de dados se encontra neste link: https://github.com/of-allan-borges/Projeto-Individual---Banco-de-dados/blob/main/populandoTabelas/populandoTabelas.sql.

---

5. Criando Indexação:

5.1. Índice em titulo: CREATE INDEX idx_livro_titulo ON livro(titulo);

- **Buscas por título serão bastante comuns na biblioteca porque quando o usuário for pegar um livro emprestado, ele vai identificar o livro pelo título, então assim o funcionário vai procurar a disponibilidade deste livro através do seu título.**
- **Exemplo de consulta: SELECT titulo, editora, quantidade_disponivel FROM livro WHERE titulo = 'Livro 1';**

5.2. Índice em id_local na tabela livro: CREATE INDEX idx_livro_local ON livro(id_local);

5.2.1. Índice em id_local e id_biblioteca na tabela local: CREATE INDEX idx_local_biblioteca ON local(id_local, id_biblioteca);

- **O primeiro índice ajuda a acelerar a busca dos locais dos livros, o segundo ajuda a otimizar o JOIN entre livro e local, além de agilizar a pesquisa pela biblioteca associada a um local. Os funcionários precisam pesquisar o local do livro sempre que um usuário solicitar o empréstimo de algum livro, pois os livros estão associados a um local específico e cada local pertence a uma das três bibliotecas. Então esses índices vão agilizar esse processo.**
- **Exemplo de consulta: SELECT livro.titulo, local.andar, local.prateleira, biblioteca.nome_biblioteca FROM livro JOIN local ON livro.id_local = local.id_local JOIN biblioteca ON local.id_biblioteca = biblioteca.id_biblioteca WHERE livro.id_livro = 2; --Digite o id do livro aqui**

5.3. Índice para Busca de Livros com Base em Quantidade Total e Quantidade Disponível: CREATE INDEX idx_livro_quantidades ON livro(quantidade_total, quantidade_disponivel);

- **Essa consulta será feita frequentemente para verificar se há livros disponíveis para empréstimo com base na quantidade total e controlar o estoque de livros.**
- **Exemplo de consulta: SELECT titulo, quantidade_total, quantidade_disponivel FROM livro WHERE quantidade_disponivel > 0;**

6.1. Consultas Simples:

1. Gerenciamento de Livros

- **Inserir um novo livro na tabela:**

  INSERT INTO livro (titulo, editora, data_publicacao, quantidade_disponivel, id_local, quantidade_total) VALUES ('Dom Casmurro', 'Editora A', '1899-01-01', 5, 1, 10);

- **Alterar a quantidade disponível de um livro específico:**

  UPDATE livro SET quantidade_disponivel = 3 WHERE id_livro = 1;

- **Excluir um livro da tabela livro:**

  DELETE FROM livro WHERE id_livro = 1;

2. Gerenciamento de Usuários

- **Inserir um novo usuário:**

  INSERT INTO usuario (nome, email, telefone, data_registro, endereco, data_nascimento) VALUES ('Maria Silva', 'maria@example.com', '(11) 98765-4321', '2025-01-12', 'Rua das Flores, 123', '1995-02-25');

- **Alterar o telefone de um usuário:**

UPDATE usuario SET telefone = '(11) 99876-5432' WHERE id_usuario = 1;

- **Excluir um usuário:**

  DELETE FROM usuario WHERE id_usuario = 1;

3. **Gerenciamento de Multas**
   - **Inserir uma nova multa:**

     INSERT INTO multa (id_funcionario_responsavel, id_usuario, valor, data_multa, status_multa, descricao) VALUES (2, 3, 50.00, '2025-01-12', 'Pendente', 'Multa por atraso no retorno do livro');

   - **Alterar o status de uma multa:**

     UPDATE multa SET status_multa = 'Pago', data_pagamento = '2025-01-13' WHERE id_multa = 1;

   - **Excluir uma multa:**

     DELETE FROM multa WHERE id_multa = 1;

4. **Gerenciamento de Funcionários**

   - **Inserir um novo funcionário:**

     INSERT INTO funcionario (nome, cargo, data_contratacao, salario, email, telefone, id_biblioteca) VALUES ('Carlos Oliveira', 'Assistente', '2025-01-01', 3000.00, 'carlos@biblioteca.com', '(11) 99876-5432', 1);

   - **Alterar o salário de um funcionário:**

UPDATE funcionario SET salario = 3200.00 WHERE id_funcionario = 1;

- ## Excluir um funcionário:

  DELETE FROM funcionario WHERE id_funcionario = 1;

6.2. Consultas Intermediárias:

1. ## Buscar os livros disponíveis para empréstimo de uma biblioteca específica:

   SELECT livro.titulo, livro.quantidade_disponivel, biblioteca.nome_biblioteca

   FROM livro

   JOIN local ON livro.id_local = local.id_local

   JOIN biblioteca ON local.id_biblioteca = biblioteca.id_biblioteca

   WHERE biblioteca.id_biblioteca = 1

   AND livro.quantidade_disponivel > 0;

2. ## Buscar todos os empréstimos de um usuário específico:

   SELECT emprestimo.id_emprestimo, livro.titulo, emprestimo.data_emprestimo, emprestimo.data_devolucao, emprestimo.status FROM emprestimo JOIN livro ON emprestimo.id_livro = livro.id_livro WHERE emprestimo.id_usuario = 1;

3. ## Buscar livros e seus autores:

   SELECT livro.titulo, autor.nome_autor FROM livro JOIN livro_autor ON livro.id_livro = livro_autor.id_livro JOIN autor ON livro_autor.id_autor = autor.id_autor;

6.3. Consultas Avançadas:

1. **Buscar o total de livros emprestados por um usuário e o valor total das multas:**

   SELECT usuario.nome, COUNT(emprestimo.id_emprestimo) AS total_emprestimos, SUM(multa.valor) AS total_multas FROM usuario JOIN emprestimo ON usuario.id_usuario = emprestimo.id_usuario JOIN multa ON emprestimo.id_emprestimo = multa.id_emprestimo WHERE usuario.id_usuario = 1 GROUP BY usuario.nome;

2. **Buscar o número de livros disponíveis em cada biblioteca, separados por tipo de local:**

   SELECT biblioteca.nome_biblioteca, local.tipo_local, SUM(livro.quantidade_disponivel) AS total_disponivel FROM livro JOIN local ON livro.id_local = local.id_local JOIN biblioteca ON local.id_biblioteca = biblioteca.id_biblioteca GROUP BY biblioteca.nome_biblioteca, local.tipo_local ORDER BY biblioteca.nome_biblioteca, total_disponivel DESC;

3. **Buscar todos os livros, seus autores e o número de empréstimos realizados:**

   SELECT livro.titulo, autor.nome_autor, COUNT(emprestimo.id_emprestimo) AS total_emprestimos FROM livro

   JOIN livro_autor ON livro.id_livro = livro_autor.id_livro

   JOIN autor ON livro_autor.id_autor = autor.id_autor

   LEFT JOIN emprestimo ON livro.id_livro = emprestimo.id_livro

   GROUP BY livro.titulo, autor.nome_autor

   ORDER BY total_emprestimos DESC;

---

7 Política de Backup:

A biblioteca é responsável por gerenciar uma grande quantidade de dados essenciais para o funcionamento adequado das suas operações diárias. As informações sobre livros, usuários, empréstimos, reservas, multas, gêneros e autores são fundamentais para garantir que os serviços prestados à comunidade funcionem de maneira eficiente. A perda de dados ou falhas no sistema podem resultar em prejuízos financeiros e operacionais, afetando diretamente a experiência dos usuários e a eficiência dos serviços oferecidos pela biblioteca. Portanto, a implementação de uma política de backup robusta é crucial para proteger a integridade dessas informações e assegurar a continuidade das operações, especialmente em caso de falhas inesperadas.

7.1 Tipos de Backup

O sistema adotará três tipos principais de backup para garantir a proteção completa dos dados:

- **Backup Completo (Full Backup): Realizado semanalmente, esse tipo de backup captura todos os dados do sistema, criando uma cópia completa e independente de todas as tabelas críticas.**
- **Backup Incremental: Esse backup será realizado diariamente, capturando apenas as mudanças feitas desde o último backup, seja ele completo ou incremental.**
- **Backup Diferencial: Caso o backup completo precise ser restaurado, um backup diferencial será realizado a cada dois dias, garantindo que todas as alterações feitas desde o último full backup sejam capturadas, sem a necessidade de múltiplos backups incrementais para restaurar os dados.**

7.2 Tabelas Críticas

As tabelas mais críticas para o funcionamento contínuo do Sistema de Informações da biblioteca, que terão sua proteção priorizada, incluem: Usuário, Livro, Empréstimo, Reserva, Multa.

Essas tabelas serão armazenadas e protegidas de forma prioritária, com backups realizados no servidor em nuvem, garantindo que dados essenciais estejam disponíveis e possam ser restaurados rapidamente em caso de falhas.

7.3 Frequência de Backup

- **Backup Completo: Será realizado semanalmente, garantindo que todos os dados críticos sejam copiados em um único ponto no tempo.**
- **Backup Incremental: Será realizado diariamente para capturar todas as alterações feitas nas tabelas críticas desde o último backup, com ênfase em atualizações feitas no empréstimo, reserva e multa.**
- **Backup Diferencial: A cada dois dias, um backup diferencial será realizado para garantir a captura de todas as alterações desde o último backup completo.**

7.4 Tempo de Retenção

Os backups serão mantidos por um período de 90 dias. Após esse período, backups antigos serão removidos para liberar espaço, com exceção do último backup completo, que será mantido por um ano, como uma medida extra de segurança.

7.5 Local de Armazenamento

Todos os backups, tanto completos quanto incrementais e diferenciais, serão armazenados em um servidor na nuvem. Isso garante que os dados estejam protegidos contra falhas de hardware locais e que possam ser acessados de forma remota, proporcionando alta disponibilidade e facilidade de recuperação em qualquer situação de emergência.

7.6 Tabelas Menos Críticas

Além das tabelas críticas, também existem outras tabelas no sistema que, embora importantes, não têm o mesmo nível de urgência para garantir a continuidade imediata das operações. Essas tabelas incluem: Autor, Genero, Funcionário, Biblioteca.

Embora essas tabelas não sejam críticas para a operação imediata do sistema, elas ainda são vitais para o funcionamento geral e a gestão de dados. Os backups dessas tabelas serão feitos de forma semelhante aos dados críticos, mas com uma frequência reduzida. O backup completo para essas tabelas será realizado a cada 15 dias, com backups incrementais diários.

7.7 Conclusão

A estratégia de backup adotada para o Sistema de Informações da biblioteca assegura que os dados essenciais, como empréstimos, reservas e multas, estarão sempre protegidos e poderão ser rapidamente restaurados em caso de falha. A combinação de backups completos, incrementais e diferenciais, com armazenamento em nuvem, garante a segurança e a integridade dos dados enquanto mantém a performance do sistema. Com uma política de retenção clara e uma frequência bem definida de backups, a biblioteca estará preparada para qualquer imprevisto, garantindo a continuidade dos seus serviços e a proteção das informações de seus usuários.

---

8. Materialized View

- **View para empréstimos e reservas de um Usuário**

Permite consultar todos os empréstimos e reservas feitas por um determinado usuário, com detalhes sobre o status, datas e livros envolvidos.

```
CREATE MATERIALIZED VIEW mv_emprestimos_reservas_usuario AS
SELECT
    u.id_usuario AS usuario_id,
    u.nome AS usuario_nome,
    l.titulo AS livro_titulo,
    e.data_emprestimo AS data_emprestimo,
```

```
        e.data_devolucao AS data_devolucao,
        e.status AS status_emprestimo,
        r.data_reserva AS data_reserva,
        r.status_reserva AS status_reserva,
        r.data_validade AS data_validade_reserva
    FROM
        usuario u
    JOIN emprestimo e ON u.id_usuario = e.id_usuario
    JOIN livro l ON e.id_livro = l.id_livro
    LEFT JOIN reserva r ON u.id_usuario = r.id_usuario AND l.id_livro = r.id_livro;
```

- **View para Livros disponíveis por Biblioteca**

Exibe todos os livros disponíveis em cada biblioteca, com suas respectivas localizações (prateleira, andar), e a quantidade disponível para empréstimo.

```
CREATE MATERIALIZED VIEW mv_livros_disponiveis_por_biblioteca AS

SELECT

    b.id_biblioteca AS biblioteca_id,

    b.nome_biblioteca AS biblioteca_nome,

    l.descricao AS local_descricao,

    l.andar AS local_andar,

    l.prateleira AS local_prateleira,

    l.tipo_local AS tipo_local,

    l.capacidade AS capacidade_local,

    livro.titulo AS livro_titulo,

    livro.quantidade_disponivel AS quantidade_disponivel

FROM

    biblioteca b

JOIN local l ON b.id_biblioteca = l.id_biblioteca

JOIN livro ON l.id_local = livro.id_local

WHERE
```

```
    livro.quantidade_disponivel > 0;
```

- **View para multas aplicadas por Usuário**

Exibe todas as multas aplicadas aos usuários, com informações sobre o valor, status e a razão da multa.

```
CREATE MATERIALIZED VIEW mv_multas_por_usuario AS
SELECT
    u.id_usuario AS usuario_id,
    u.nome AS usuario_nome,
    m.valor AS valor_multa,
    m.status_multa AS status_multa,
    m.descricao AS descricao_multa,
    m.data_multa AS data_multa,
    m.data_pagamento AS data_pagamento
FROM
    multa m
JOIN usuario u ON m.id_usuario = u.id_usuario;
```

---

9. Stores Procedures

- **Atualizar a view de Empréstimos e Reservas de um Usuário**

```
CREATE OR REPLACE PROCEDURE
sp_atualizar_mv_emprestimos_reservas_usuario()
LANGUAGE plpgsql
AS $$
BEGIN
    REFRESH MATERIALIZED VIEW mv_emprestimos_reservas_usuario;
END;
$$;
```

- **Atualizar a view de Livros disponíveis por Biblioteca**

```
CREATE OR REPLACE PROCEDURE
sp_atualizar_mv_livros_disponiveis_por_biblioteca()
LANGUAGE plpgsql
```

```
AS $$
BEGIN
        REFRESH MATERIALIZED VIEW mv_livros_disponiveis_por_biblioteca;
END;
$$;
```

- **Atualizar a view de Multas aplicadas por Usuário**

```
CREATE OR REPLACE PROCEDURE sp_atualizar_mv_multas_por_usuario()
LANGUAGE plpgsql
AS $$
BEGIN
        REFRESH MATERIALIZED VIEW mv_multas_por_usuario;
END;
$$;
```

---

10. Perguntas Analíticas

Descrição da Query	Pergunta Analítica	Query	Justificativa da Query
Obter quantidade de livros por biblioteca	"Quantos livros cada biblioteca possui?"	SELECT id_biblioteca, COUNT(*) AS num_livros FROM livro GROUP BY id_biblioteca;	Visão geral sobre a distribuição de livros entre as bibliotecas.
Verificar empréstimos em atraso	"Quais empréstimos estão atrasados?"	SELECT * FROM emprestimo WHERE data_devolucao < CURRENT_DATE AND status = 'Em andamento';	Identifica empréstimos em atraso para tomada de ação.
Listar livros mais reservados	"Quais são os livros mais reservados?"	SELECT id_livro, COUNT(*) AS num_reservas FROM reserva GROUP BY id_livro	Ajuda a identificar os livros mais populares entre os usuários.

		ORDER BY num_reservas DESC;	
Consultar multas em aberto por usuário	"Quais multas um usuário possui em aberto?"	SELECT * FROM multa WHERE id_usuario = X AND status_multa = 'Aberta';	Auxilia na cobrança e regularização de pendências.
Gêneros mais populares	"Quais gêneros literários são mais populares?"	SELECT id_genero, COUNT(*) AS num_livros FROM livro_genero GROUP BY id_genero ORDER BY num_livros DESC;	Fornece insights sobre a preferência dos usuários por gêneros.
Livros em locais com alta capacidade	"Quais livros estão em locais com capacidade acima de 100?"	SELECT l.* FROM livro l INNER JOIN local loc ON l.id_local = loc.id_local WHERE loc.capacidade > 100;	Avalia o uso do espaço nas bibliotecas.
Empréstimos por funcionário	"Quantos empréstimos cada funcionário realizou?"	SELECT id_funcionario_responsavel, COUNT(*) AS num_emprestimos FROM emprestimo GROUP BY id_funcionario_responsavel;	Mede a performance dos funcionários no atendimento ao público.
Livros disponíveis por gênero	Quantos livros estão disponíveis por gênero?	SELECT g.nome_genero, COUNT(lg.id_livro) AS num_livros_disponiveis FROM genero g INNER JOIN livro_genero lg ON g.id_genero = lg.id_genero INNER JOIN livro l ON lg.id_livro = l.id_livro WHERE l.quantidade_disponivel > 0 GROUP BY g.nome_genero	Esta consulta fornece insights sobre a disponibilidade de livros por gênero, ajudando a identificar quais gêneros possuem mais ou menos opções

		ORDER BY num_livros_disponiveis DESC;	disponíveis para os usuários.
Funcionários que mais processam reservas	Quais funcionários processam mais reservas?	SELECT f.nome, COUNT(r.id_reserva) AS num_reservas FROM funcionario f INNER JOIN reserva r ON f.id_funcionario = r.id_funcionario_responsavel GROUP BY f.nome ORDER BY num_reservas DESC;	Ajuda a avaliar a performance dos funcionários no processamento de reservas e a identificar quem está mais ativo nesta atividade.
Usuários mais ativos no sistema	Quais são os usuários mais ativos no sistema?	SELECT u.nome, COUNT(DISTINCT e.id_emprestimo) AS num_emprestimos, COUNT(DISTINCT r.id_reserva) AS num_reservas FROM usuario u LEFT JOIN emprestimo e ON u.id_usuario = e.id_usuario LEFT JOIN reserva r ON u.id_usuario = r.id_usuario GROUP BY u.nome ORDER BY (num_emprestimos + num_reservas) DESC;	Identifica os usuários mais engajados, considerando tanto os empréstimos quanto as reservas, oferecendo insights sobre o comportamento e o uso do sistema.

11. Telas de Aplicações

Tela 1: Controle de Usuários

Finalidade: Esta tela é destinada ao gerenciamento de usuários da biblioteca. O administrador pode visualizar os dados dos usuários, alterar suas informações, excluir usuários (por exemplo, usuários com multas pendentes), e incluir novos usuários.

Funcionalidades:

1. **Inclusão de Usuário:** O administrador pode adicionar um novo usuário à biblioteca.
2. **Alteração de Dados do Usuário:** O administrador pode alterar as informações dos usuários.
3. **Exclusão de Usuário:** O administrador pode excluir um usuário caso ele tenha 5 ou mais multas, ou em outras condições.
4. **Busca de Usuários:** O administrador pode buscar usuários por nome, e-mail ou telefone.

Queries:

1. **Inclusão de Usuário:**

   Interface da tela:

   Campos:

- **Nome do Usuário: [_____]**
- **Data de Nascimento: [//_____]**
- **Telefone: [_____]**
- **E-mail: [_____]**
- **Endereço: [_____]**

  Botões:

- **Adicionar Usuário (Botão para salvar os dados do novo usuário).**
- **Cancelar (Botão para cancelar a ação de inclusão).**

INSERT INTO usuario (nome, email, telefone, data_registro, endereco, data_nascimento) VALUES ('João Silva', 'joao.silva@email.com', '123456789', '2025-01-12', 'Rua A, 123', '1990-05-15');

2. **Alteração de Dados do Usuário:**

   Interface da tela:

   Campos:

- **ID do Usuário: [_____]**
- **Telefone: [_____]**
- **Endereço: [_____]**

   Botões:

- **Atualizar Dados (Botão para salvar as alterações feitas).**
- **Cancelar (Botão para cancelar as alterações).**

```
UPDATE usuario
SET telefone = '987654321', endereco = 'Rua B, 456'
WHERE id_usuario = 1;
```

3. **Exclusão de Usuário (caso tenha 5 ou mais multas):**

   Interface da tela:

- **Campos:**
  - **Nome do Usuário: [_____]**
  - **ID do Usuário: [____] (identificador único do usuário).**
- **Botões:**
  - **Excluir Usuário (Botão para excluir o usuário).**
  - **Cancelar (Botão para cancelar a exclusão).**

```
DELETE FROM usuario
WHERE id_usuario = 1
AND (SELECT COUNT(*) FROM multa WHERE id_usuario = 1) >= 5;
```

4. **Busca de Usuário por Nome:**

Interface da tela:

Campos:

- **Nome: [_____]**

Botões:

- **Pesquisar (Botão para realizar a busca).**
- **Limpar Filtros (Botão para limpar os filtros da busca).**

```
SELECT * FROM usuario
WHERE nome LIKE '%Usuario50%';
```

Tela 2: Controle de Livros

Finalidade: Esta tela é usada para adicionar novos livros ao acervo, alterar detalhes dos livros e excluir livros do sistema.

Funcionalidades:

1. **Inclusão de Livro:** O administrador pode adicionar novos livros ao catálogo.
2. **Alteração de Dados do Livro:** O administrador pode atualizar informações do livro, como título, editora, ou quantidade disponível.
3. **Exclusão de Livro:** O administrador pode excluir livros que não estão mais disponíveis ou não são mais necessários.
4. **Busca de Livros:** O administrador pode buscar livros pelo título, autor ou gênero.

Queries:

1. Inclusão de Livro:

Interface da tela:

Campos:

- **Título do Livro: [_____]**
- **Editora: [_____]**

- **Data de Publicação: [//_____]**
- **Quantidade Disponível: [____]**
- **Local do Livro: [_____] (Selecionar local de prateleira/andar).**
- **Quantidade Total: [____]**

Botões:

- **Adicionar Livro (Botão para salvar o novo livro).**
- **Cancelar (Botão para cancelar a inclusão).**

```
INSERT INTO livro (titulo, editora, data_publicacao, quantidade_disponivel, id_local,
quantidade_total)
VALUES ('O Grande Livro', 'Editora X', '2023-08-20', 10, 1, 20);
```

2. Alteração de Dados do Livro:

Interface da tela:

- **Campos:**
  - **ID do Livro: [_____]**
  - **Quantidade Disponível: [____]**
- **Botões:**
  - **Atualizar Dados (Botão para salvar as alterações feitas).**
  - **Cancelar (Botão para cancelar as alterações).**

```
UPDATE livro
SET quantidade_disponivel = 8
WHERE id_livro = 1;
```

3. Exclusão de Livro:

Interface da tela:

Campos:

- **Título do Livro: [_____]**

- **ID do Livro: [____] (identificador único do livro).**

  Botões:

- **Excluir Livro (Botão para excluir o livro).**
- **Cancelar (Botão para cancelar a exclusão)**

```
DELETE FROM livro
WHERE id_livro = 1;
```

4. Busca de Livro por Título:

Interface da tela:

Campos:

- **Título: [_____]**

  Botões:

- **Pesquisar (Botão para realizar a busca).**
- **Limpar Filtros (Botão para limpar os filtros da busca).**

```
SELECT * FROM livro
WHERE titulo LIKE '%Livro_1%';
```

Tela 3: Controle de Funcionários

Finalidade: Esta tela permite que o administrador gerencie os funcionários da biblioteca. O administrador pode adicionar, alterar e excluir dados de funcionários.

Funcionalidades:

1. **Inclusão de Funcionário:** O administrador pode incluir novos funcionários.

2. **Alteração de Dados do Funcionário:** O administrador pode alterar informações como salário ou cargo.
3. **Exclusão de Funcionário:** O administrador pode excluir funcionários da biblioteca.
4. **Busca de Funcionários:** O administrador pode buscar funcionários por nome ou cargo.

Queries:

1. **Inclusão de Funcionário:**

   Interface da tela:

   Campos:

- **Nome do Funcionário: [_____]**
- **Cargo: [_____] (Selecione o cargo: ex: Bibliotecário, Assistente, etc.)**
- **Data de Contratação: [//_____]**
- **Salário: [_____]**
- **E-mail: [_____]**
- **Telefone: [_____]**
- **Biblioteca: [_____] (Selecione a biblioteca em que o funcionário vai atuar).**

  Botões:

- **Adicionar Funcionário (Botão para salvar o novo funcionário).**
- **Cancelar (Botão para cancelar a inclusão).**

```
INSERT INTO funcionario (nome, cargo, data_contratacao, salario, email, telefone,
id_biblioteca)
VALUES ('Funcionario_101, 'Bibliotecário', '2024-02-01', 2500.00,
'funcionario_101@email.com', '987654321', 1);
```

2. **Alteração de Dados do Funcionário:**

   Interface da tela:

- **Campos:**
    - **ID do Funcionário: [_____]**
    - **Cargo: [_____]**
    - **Salário: [_____]**
    - **E-mail: [_____]**
    - **Telefone: [_____]**
- **Botões:**
    - **Atualizar Dados (Botão para salvar as alterações feitas).**
    - **Cancelar (Botão para cancelar as alterações).**

```
UPDATE funcionario
SET salario = 3000.00
WHERE id_funcionario = 1;
```

3. **Exclusão de Funcionário:**

   Interface da tela:

   Campos:

- **Nome do Funcionário: [_____]**
- **ID do Funcionário: [____] (identificador único do funcionário).**

  Botões:

- **Excluir Funcionário (Botão para excluir o funcionário).**
- **Cancelar (Botão para cancelar a exclusão).**

```
DELETE FROM funcionario
WHERE id_funcionario = 1;
```

4. **Busca de Funcionário por Cargo:**

   Interface da tela:

   Campos:

- **Cargo: [_____] (Exemplo: "Bibliotecário").**
- **Nome: [_____] (Exemplo: "Carlos Oliveira").**

Botões:

- **Pesquisar (Botão para realizar a busca).**
- **Limpar Filtros (Botão para limpar os filtros da busca).**

```
SELECT * FROM funcionario
WHERE cargo LIKE '%Bibliotecário%';
```

Tela 4: Controle de Multas

Finalidade: Esta tela é utilizada para visualizar, calcular, alterar e excluir multas aplicadas aos usuários da biblioteca.

Funcionalidades:

1. **Inclusão de Multa:** O administrador pode adicionar multas para um usuário.
2. **Alteração de Multa:** O administrador pode alterar o status da multa ou adicionar informações.
3. **Exclusão de Multa:** O administrador pode excluir uma multa.
4. **Busca de Multas:** O administrador pode buscar multas por status ou por usuário.

Queries:

1. **Inclusão de Multa:**

   Interface da tela:

   Campos:

- **ID do Funcionário Responsável: [_____] (Seleção do funcionário que aplicou a multa).**
- **ID do Usuário: [_____] (Seleção do usuário que recebeu a multa).**
- **Valor da Multa: [_____]**

- **Descrição: [_____] (Motivo da multa, ex: "Atraso no empréstimo").**
- **Data da Multa: [//_____]**
- **Status da Multa: [Pendente] (Seleção do status: Pendente, Pago).**
- **Data de Pagamento: [//_____] (Somente preenchido se o status for "Pago").**
- **ID do Empréstimo: [_____] (Seleção do empréstimo relacionado).**

Botões:

- **Registrar Multa (Botão para salvar a nova multa).**
- **Cancelar (Botão para cancelar a inclusão da multa).**

```
INSERT INTO multa (id_funcionario_responsavel, id_usuario, valor, data_multa,
status_multa, descricao, data_pagamento, id_emprestimo)
VALUES (1, 1, 15.00, '2025-01-12', 'Pendente', 'Multa por atraso no livro', NULL, 1);
```

2. **Alteração de Status de Multa:**

   Interface da tela:

   Campos:

- **ID da Multa: [____] (Identificador único da multa).**
- **Status Atual: [Pendente/Pago] (Campo de leitura para o status atual).**
- **Novo Status: [Pago] (Alteração do status).**
- **Data de Pagamento: [//_____] (Somente habilitado se o status for alterado para "Pago").**

Botões:

- **Alterar Status (Botão para salvar a alteração do status da multa).**
- **Cancelar (Botão para cancelar a alteração do status).**

```
UPDATE multa
SET status_multa = 'Pago', data_pagamento = '2025-01-15'
WHERE id_multa = 1;
```

3. **Exclusão de Multa:**

Interface da tela:

Campos:

- **ID da Multa: [____] (Identificador único da multa).**
- **Nome do Usuário: [_____]**

Botões:

- **Excluir Multa (Botão para excluir a multa).**
- **Cancelar (Botão para cancelar a exclusão).**

```
DELETE FROM multa
WHERE id_multa = 1;
```

4. **Busca de Multas por Status:**

   Interface da tela:

   Campos:

- **Status da Multa: [Pendente] (Seleção do status das multas a serem consultadas).**
- **Data da Multa: [//_____] (Filtragem por data).**
- **Nome do Usuário: [_____] (Campo opcional para busca pelo nome do usuário).**

Botões:

- **Pesquisar (Botão para realizar a busca).**
- **Limpar Filtros (Botão para limpar os filtros da busca).**

```
SELECT * FROM multa
WHERE status_multa = 'Pendente';
```

Tela 5: Controle de Reservas

Finalidade: Esta tela é usada para gerenciar as reservas de livros feitas pelos usuários da biblioteca.

Funcionalidades:

1. **Inclusão de Reserva:** O administrador pode criar reservas para os usuários.
2. **Alteração de Reserva:** O administrador pode alterar a data de validade ou o status da reserva.
3. **Exclusão de Reserva:** O administrador pode excluir reservas quando necessário.
4. **Busca de Reservas:** O administrador pode buscar reservas por usuário ou status.

Queries:

1. **Inclusão de Reserva:**

   Interface da tela:

   Campos:

- **ID do Usuário: [_____] (Seleção do usuário que fez a reserva).**
- **ID do Livro: [_____] (Seleção do livro que foi reservado).**
- **Data da Reserva: [//_____] (Data em que a reserva foi realizada).**
- **Status da Reserva: [Ativa] (Seleção do status da reserva, ex: Ativa, Cancelada).**
- **Data de Validade: [//_____] (Data em que a reserva expira).**
- **Tipo de Reserva: [Normal] (Seleção do tipo de reserva, ex: Normal, Prioritária).**
- **ID do Funcionário Responsável: [_____] (Seleção do funcionário que processou a reserva).**

  Botões:

- **Registrar Reserva (Botão para salvar a nova reserva).**
- **Cancelar (Botão para cancelar a inclusão da reserva).**

INSERT INTO reserva (id_usuario, id_livro, data_reserva, status_reserva, data_validade, tipo_reserva, id_funcionario_responsavel)

```
VALUES (1, 2, '2025-01-12', 'Ativa', '2025-02-12', 'Normal', 1);
```

2. **Alteração de Status de Reserva:**

   Interface da tela:

   Campos:

- **ID da Reserva: [____] (Identificador único da reserva).**
- **Status Atual: [Ativa/Cancelada] (Campo de leitura para o status atual).**
- **Novo Status: [Cancelada] (Alteração do status da reserva).**
- **Data de Validade: [//_____] (Se o status for alterado, habilita campo para alteração da data de validade).**

   Botões:

- **Alterar Status (Botão para salvar a alteração do status da reserva).**
- **Cancelar (Botão para cancelar a alteração do status).**

```
UPDATE reserva
SET status_reserva = 'Cancelada'
WHERE id_reserva = 1;
```

3. **Exclusão de Reserva:**

   Interface da tela:

   Campos:

- **ID da Reserva: [____] (Identificador único da reserva).**
- **Nome do Usuário: [_____]**

   Botões:

- **Excluir Reserva (Botão para excluir a reserva).**
- **Cancelar (Botão para cancelar a exclusão).**

```
DELETE FROM reserva
WHERE id_reserva = 1;
```

4. **Busca de Reservas por Status:**

   Interface da tela:

   Campos:

- **Status da Reserva: [Ativa] (Seleção do status das reservas a serem consultadas).**
- **Nome do Usuário: [_____] (Campo opcional para busca pelo nome do usuário).**
- **Livro Reservado: [_____] (Campo opcional para busca pelo título do livro reservado).**

Botões:

- **Pesquisar (Botão para realizar a busca).**
- **Limpar Filtros (Botão para limpar os filtros da busca).**

```
SELECT * FROM reserva
WHERE status_reserva = 'Ativa';
```
