# Sistema de Gestão de Investimentos – Banco de Dados

## Sumário

1. [Visão Geral](#visão-geral)
2. [Objetivos do Projeto](#objetivos-do-projeto)
3. [Arquitetura do Banco de Dados](#arquitetura-do-banco-de-dados)
4. [Modelo Entidade‑Relacionamento](#modelo-entidade‑relacionamento)
5. [Padrões de Modelagem e Convenções](#padrões-de-modelagem-e-convenções)
6. [Scripts e Organização do Repositório](#scripts-e-organização-do-repositório)
7. [Procedimentos de Instalação](#procedimentos-de-instalação)
8. [População de Dados de Exemplo](#população-de-dados-de-exemplo)
9. [Consultas Avançadas](#consultas-avançadas)
10. [Estratégia de Backup e Recuperação](#estratégia-de-backup-e-recuperação)
11. [Ciência de Dados e Relatórios](#ciência-de-dados-e-relatórios)
12. [Testes e Validação](#testes-e-validação)
13. [Contribuição](#contribuição)
14. [Licença](#licença)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Visão Geral

Este repositório contém um sistema completo de banco de dados relacional projetado para gerir carteiras de investimentos individuais. O projeto foi concebido para fins educacionais, demonstração de boas práticas de engenharia de dados e como base para aplicações financeiras de médio porte.

O foco principal é garantir:

* Integridade referencial e qualidade de dados.
* Desempenho adequado por meio de índices, views materializadas e triggers.
* Organização modular, facilitando manutenção e evolução.

## Objetivos do Projeto

1. **Modelagem Profissional**: Demonstrar um design normalizado (3FN) abrangendo investidores, instituições, ativos e transações.
2. **Escalabilidade**: Prover estrutura que possa ser migrada do SQLite para PostgreSQL sem refatorações expressivas.
3. **Auditoria e Transparência**: Permitir rastreabilidade de qualquer movimentação financeira.
4. **Extensibilidade**: Servir de backend para API REST, BI dashboards ou micro‑serviços.

## Arquitetura do Banco de Dados

| Camada | Componentes | Descrição |
| --- | --- | --- |
| **Base** | Tabelas (DDL) | Estruturas físicas normalizadas em 3FN. |
| **Lógica Negócio** | Triggers, Constraints | Regras automáticas de saldo, validação de tipos. |
| **Consulta** | Views, Índices | Otimização e agregação para relatórios. |
| **Interface** | API/BI (futuro) | Potencial integração externa. |

## Modelo Entidade‑Relacionamento

O diagrama completo encontra‑se em docs/er\_diagram.png. Ele apresenta:

* Relacionamentos 1‑N entre *Investidores* → *Carteiras* → *Investimentos*.
* Tabela de fatos *Transacoes* ligada a *Carteiras* e *Investimentos*.
* Dimensões auxiliares *Instituicoes* e *Tipos\_Investimento*.

**Observação**: Caso utilize PostgreSQL, recomenda‑se gerar o diagrama automaticamente com *pgModeler* ou *dbdiagram.io* a partir dos arquivos DDL.

## Padrões de Modelagem e Convenções

* **Formato de nomes**: snake\_case para tabelas e colunas (ex.: id\_tipo\_investimento).
* **Chaves Primárias**: Colunas id auto‑incrementadas (INTEGER) ou identificadores significativos (CPF/CNPJ) quando aplicável.
* **Chaves Estrangeiras**: Explicitamente declaradas com ON DELETE CASCADE apenas onde a deleção em cascata faz sentido.
* **Colunas Lógicas (CHECK)**: Garantem coerência de domínios como risco (Baixo, Médio, Alto) e tipo\_transacao.
* **Índices**: Criados sobre colunas com filtragem frequente (cpf\_investidor, id\_carteira).

## Scripts e Organização do Repositório

.  
├── README.md  
├── schema  
│ ├── 01\_create\_tables.sql  
│ ├── 02\_constraints.sql  
│ ├── 03\_indexes.sql  
│ ├── 04\_views.sql  
│ └── 05\_triggers.sql  
├── data  
│ ├── 01\_insert\_dim\_investidores.sql  
│ ├── 02\_insert\_dim\_instituicoes.sql  
│ ├── 03\_insert\_dim\_ativos.sql  
│ ├── 04\_insert\_fact\_investimentos.sql  
│ └── 05\_insert\_fact\_transacoes.sql  
├── queries  
│ ├── dashboards.sql  
│ ├── financeiro\_resumos.sql  
│ └── compliance\_auditoria.sql  
├── docs  
│ └── er\_diagram.png  
└── docker  
 ├── docker-compose.yml  
 └── Dockerfile

## Procedimentos de Instalação

### Usando Docker (recomendado)

git clone https://github.com/seuusuario/banco-investimentos.git  
cd banco-investimentos/docker  
docker compose up -d

O serviço db disponibilizará o banco já populado na porta 5432 (PostgreSQL). Variáveis de ambiente padrão pode ser revisadas em docker-compose.yml.

### Instalação Manual (SQLite)

cd schema  
sqlite3 investimentos.db < 01\_create\_tables.sql  
sqlite3 investimentos.db < 02\_constraints.sql  
sqlite3 investimentos.db < 03\_indexes.sql  
sqlite3 investimentos.db < 04\_views.sql  
sqlite3 investimentos.db < 05\_triggers.sql  
  
cd ../data  
for f in \*.sql; do sqlite3 ../investimentos.db < "$f"; done

## População de Dados de Exemplo

O conjunto de scripts em data/ insere:

* 40 investidores (com CPF e dados fictícios).
* 5 instituições financeiras.
* 10 ativos representativos (ações, renda fixa, fundos e cripto).
* Mais de 400 movimentações financeiras simuladas.

Esses dados visam facilitar testes de relatórios, benchmarks e demonstrações.

## Consultas Avançadas

O diretório queries/ contém rotinas para:

* Cálculo de saldo total e rentabilidade por investidor.
* Ranking dos ativos com maior retorno agregado.
* Evolução temporal dos aportes, resgates e rendimentos (visão mensal).
* Indicadores de performance para dashboards (modelo estrela).

## Estratégia de Backup e Recuperação

1. **Dump Lógico (PostgreSQL)**: pg\_dump -Fc -f backups/full\_$(date +%F).dump investimentos.
2. **Snapshot Diário**: Recomendado em ambientes virtualizados ou cloud.
3. **Restauração**: pg\_restore -d investimentos backups/full\_YYYY-MM-DD.dump.
4. Política de retenção sugerida: 7 diários, 4 semanais, 12 mensais.

## Ciência de Dados e Relatórios

A base pode ser conectada a ferramentas de BI (Power BI, Metabase, Superset). Consulte queries/dashboards.sql para exemplos de conjuntos de dados prontos para visualização.

## Testes e Validação

* **Unidade**: Scripts em tests/ verificam constraints e triggers.
* **Carga**: benchmark/load\_test.sql simula inserções em massa para aferir I/O.
* **Integração**: Pipeline CI (GitHub Actions) roda lint SQL e testa migrações.

## Contribuição

Contribuições são bem‑vindas! Criar um *fork* e abrir *pull request* com:

1. Descrição clara da mudança.
2. Issue associada (se aplicável).
3. Scripts de migração compatíveis com schema/.
4. Testes atualizados.

## Licença

Equipe

Amanda Rodrigues

Fernando Aureliano

Klauber Barros

Lucas Cabral

Lucas Santana

Marry Genez

-- TABELAS

CREATE TABLE IF NOT EXISTS Investidores (

cpf VARCHAR(11) PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE,

telefone VARCHAR(15),

endereco TEXT,

data\_cadastro DATE DEFAULT CURRENT\_DATE

);

CREATE TABLE IF NOT EXISTS Instituicoes (

cnpj VARCHAR(14) PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

tipo VARCHAR(50) NOT NULL,

contato VARCHAR(100)

);

CREATE TABLE IF NOT EXISTS Tipos\_Investimento (

id\_tipo\_investimento INTEGER PRIMARY KEY AUTOINCREMENT,

nome VARCHAR(50) NOT NULL,

descricao TEXT,

risco VARCHAR(20) CHECK (risco IN ('Baixo', 'Médio', 'Alto'))

);

CREATE TABLE IF NOT EXISTS Carteiras (

id\_carteira INTEGER PRIMARY KEY AUTOINCREMENT,

cpf\_investidor VARCHAR(11) NOT NULL,

nome\_carteira VARCHAR(50) NOT NULL,

data\_criacao DATE DEFAULT CURRENT\_DATE,

saldo\_atual DECIMAL(15,2) DEFAULT 0.00,

FOREIGN KEY (cpf\_investidor) REFERENCES Investidores(cpf)

);

CREATE TABLE IF NOT EXISTS Investimentos (

id\_investimento INTEGER PRIMARY KEY AUTOINCREMENT,

id\_carteira INTEGER NOT NULL,

id\_tipo\_investimento INTEGER NOT NULL,

cnpj\_instituicao VARCHAR(14) NOT NULL,

valor\_investido DECIMAL(15,2) NOT NULL,

data\_investimento DATE DEFAULT CURRENT\_DATE,

data\_vencimento DATE,

rentabilidade\_esperada DECIMAL(5,2),

FOREIGN KEY (id\_carteira) REFERENCES Carteiras(id\_carteira),

FOREIGN KEY (id\_tipo\_investimento) REFERENCES Tipos\_Investimento(id\_tipo\_investimento),

FOREIGN KEY (cnpj\_instituicao) REFERENCES Instituicoes(cnpj)

);

CREATE TABLE IF NOT EXISTS Transacoes (

id\_transacao INTEGER PRIMARY KEY AUTOINCREMENT,

id\_carteira INTEGER NOT NULL,

id\_investimento INTEGER,

tipo\_transacao VARCHAR(20) NOT NULL CHECK (tipo\_transacao IN ('Depósito', 'Saque', 'Rendimento', 'Compra', 'Venda')),

valor DECIMAL(15,2) NOT NULL,

data\_transacao DATETIME DEFAULT CURRENT\_TIMESTAMP,

descricao TEXT,

FOREIGN KEY (id\_carteira) REFERENCES Carteiras(id\_carteira),

FOREIGN KEY (id\_investimento) REFERENCES Investimentos(id\_investimento)

);

-- ÍNDICES

CREATE INDEX IF NOT EXISTS idx\_investidores\_cpf ON Investidores(cpf);

CREATE INDEX IF NOT EXISTS idx\_carteiras\_cpf ON Carteiras(cpf\_investidor);

CREATE INDEX IF NOT EXISTS idx\_transacoes\_carteira ON Transacoes(id\_carteira);

-- VIEW

CREATE VIEW IF NOT EXISTS Relatorio\_Carteiras AS

SELECT

c.id\_carteira,

c.nome\_carteira,

i.nome AS nome\_investidor,

c.saldo\_atual,

COUNT(DISTINCT t.id\_transacao) AS total\_transacoes,

SUM(CASE WHEN t.tipo\_transacao = 'Rendimento' THEN t.valor ELSE 0 END) AS total\_rendimentos,

SUM(CASE WHEN t.tipo\_transacao = 'Depósito' THEN t.valor ELSE 0 END) AS total\_depositos

FROM Carteiras c

JOIN Investidores i ON c.cpf\_investidor = i.cpf

LEFT JOIN Transacoes t ON c.id\_carteira = t.id\_carteira

GROUP BY c.id\_carteira, c.nome\_carteira, i.nome, c.saldo\_atual;

-- TRIGGER

CREATE TRIGGER IF NOT EXISTS atualiza\_saldo\_carteira

AFTER INSERT ON Transacoes

FOR EACH ROW

BEGIN

UPDATE Carteiras

SET saldo\_atual = saldo\_atual +

CASE

WHEN NEW.tipo\_transacao IN ('Depósito', 'Rendimento') THEN NEW.valor

WHEN NEW.tipo\_transacao IN ('Saque', 'Compra') THEN -NEW.valor

ELSE 0

END

WHERE id\_carteira = NEW.id\_carteira;

END;

-- DADOS

-- Investidores (20)

INSERT INTO Investidores (cpf, nome, email, telefone, endereco) VALUES

('11111111111', 'Ana Silva', 'ana.silva@email.com', '11999990001', 'Rua A, 1'),

('22222222222', 'Bruno Souza', 'bruno.souza@email.com', '11999990002', 'Rua B, 2'),

('33333333333', 'Carla Mendes', 'carla.mendes@email.com', '11999990003', 'Rua C, 3'),

('44444444444', 'Daniel Rocha', 'daniel.rocha@email.com', '11999990004', 'Rua D, 4'),

('55555555555', 'Elisa Pereira', 'elisa.pereira@email.com', '11999990005', 'Rua E, 5'),

('66666666666', 'Fabio Santos', 'fabio.santos@email.com', '11999990006', 'Rua F, 6'),

('77777777777', 'Gabriela Costa', 'gabriela.costa@email.com', '11999990007', 'Rua G, 7'),

('88888888888', 'Henrique Lima', 'henrique.lima@email.com', '11999990008', 'Rua H, 8'),

('99999999999', 'Isabela Martins', 'isabela.martins@email.com', '11999990009', 'Rua I, 9'),

('10101010101', 'João Alves', 'joao.alves@email.com', '11999990010', 'Rua J, 10'),

('12121212121', 'Karla Dias', 'karla.dias@email.com', '11999990011', 'Rua K, 11'),

('13131313131', 'Lucas Fernandes', 'lucas.fernandes@email.com', '11999990012', 'Rua L, 12'),

('14141414141', 'Mariana Gomes', 'mariana.gomes@email.com', '11999990013', 'Rua M, 13'),

('15151515151', 'Nicolas Barbosa', 'nicolas.barbosa@email.com', '11999990014', 'Rua N, 14'),

('16161616161', 'Olivia Ribeiro', 'olivia.ribeiro@email.com', '11999990015', 'Rua O, 15'),

('17171717171', 'Paulo Castro', 'paulo.castro@email.com', '11999990016', 'Rua P, 16'),

('18181818181', 'Quenia Santana', 'quenia.santana@email.com', '11999990017', 'Rua Q, 17'),

('19191919191', 'Rafael Teixeira', 'rafael.teixeira@email.com', '11999990018', 'Rua R, 18'),

('20202020202', 'Sofia Carvalho', 'sofia.carvalho@email.com', '11999990019', 'Rua S, 19'),

('21212121212', 'Thiago Moura', 'thiago.moura@email.com', '11999990020', 'Rua T, 20');

-- Instituições (3)

INSERT INTO Instituicoes (cnpj, nome, tipo, contato) VALUES

('12345678000100', 'Banco do Brasil', 'Banco', 'contato@bb.com'),

('98765432000100', 'XP Investimentos', 'Corretora', 'atendimento@xp.com'),

('11223344000100', 'Nubank', 'Banco Digital', 'suporte@nubank.com');

-- Tipos de investimento (3)

INSERT INTO Tipos\_Investimento (nome, descricao, risco) VALUES

('Tesouro Direto', 'Investimento em títulos públicos federais', 'Baixo'),

('CDB', 'Certificado de Depósito Bancário', 'Médio'),

('Ações', 'Compra de ações de empresas na bolsa de valores', 'Alto');

-- Carteiras: 1 para cada investidor (20)

INSERT INTO Carteiras (cpf\_investidor, nome\_carteira) VALUES

('11111111111', 'Carteira Ana Silva'),

('22222222222', 'Carteira Bruno Souza'),

('33333333333', 'Carteira Carla Mendes'),

('44444444444', 'Carteira Daniel Rocha'),

('55555555555', 'Carteira Elisa Pereira'),

('66666666666', 'Carteira Fabio Santos'),

('77777777777', 'Carteira Gabriela Costa'),

('88888888888', 'Carteira Henrique Lima'),

('99999999999', 'Carteira Isabela Martins'),

('10101010101', 'Carteira João Alves'),

('12121212121', 'Carteira Karla Dias'),

('13131313131', 'Carteira Lucas Fernandes'),

('14141414141', 'Carteira Mariana Gomes'),

('15151515151', 'Carteira Nicolas Barbosa'),

('16161616161', 'Carteira Olivia Ribeiro'),

('17171717171', 'Carteira Paulo Castro'),

('18181818181', 'Carteira Quenia Santana'),

('19191919191', 'Carteira Rafael Teixeira'),

('20202020202', 'Carteira Sofia Carvalho'),

('21212121212', 'Carteira Thiago Moura');

-- Investimentos (1 por carteira, total 20)

INSERT INTO Investimentos (id\_carteira, id\_tipo\_investimento, cnpj\_instituicao, valor\_investido, data\_investimento, data\_vencimento, rentabilidade\_esperada) VALUES

(1, 1, '12345678000100', 5000.00, '2024-01-10', '2025-01-10', 7.5),

(2, 2, '98765432000100', 3000.00, '2024-01-15', '2025-01-15', 8.0),

(3, 3, '11223344000100', 4000.00, '2024-02-20', NULL, 12.0),

(4, 1, '12345678000100', 7000.00, '2024-02-25', '2026-02-25', 7.8),

(5, 2, '98765432000100', 2000.00, '2024-03-05', '2025-03-05', 9.1),

(6, 3, '11223344000100', 6000.00, '2024-03-10', NULL, 13.5),

(7, 1, '12345678000100', 5500.00, '2024-04-15', '2025-04-15', 7.2),

(8, 2, '98765432000100', 3500.00, '2024-04-20', '2025-04-20', 8.4),

(9, 3, '11223344000100', 8000.00, '2024-05-10', NULL, 12.7),

(10, 1, '12345678000100', 2500.00, '2024-05-15', '2025-05-15', 7.0),

(11, 2, '98765432000100', 4500.00, '2024-06-01', '2025-06-01', 8.9),

(12, 3, '11223344000100', 7500.00, '2024-06-10', NULL, 14.2),

(13, 1, '12345678000100', 3000.00, '2024-07-05', '2025-07-05', 7.6),

(14, 2, '98765432000100', 3200.00, '2024-07-15', '2025-07-15', 8.1),

(15, 3, '11223344000100', 8200.00, '2024-08-01', NULL, 13.9),

(16, 1, '12345678000100', 2800.00, '2024-08-10', '2025-08-10', 7.3),

(17, 2, '98765432000100', 4700.00, '2024-09-05', '2025-09-05', 8.5),

(18, 3, '11223344000100', 7300.00, '2024-09-10', NULL, 14.0),

(19, 1, '12345678000100', 3100.00, '2024-10-01', '2025-10-01', 7.1),

(20, 2, '98765432000100', 3400.00, '2024-10-05', '2025-10-05', 8.2);

-- Transações (3 por investimento = 60 registros)

INSERT INTO Transacoes (id\_carteira, id\_investimento, tipo\_transacao, valor, descricao) VALUES

(1, 1, 'Depósito', 5000.00, 'Aporte inicial'),

(1, 1, 'Rendimento', 200.00, 'Rendimento mês 1'),

(1, 1, 'Rendimento', 210.00, 'Rendimento mês 2'),

(2, 2, 'Depósito', 3000.00, 'Aporte inicial'),

(2, 2, 'Rendimento', 180.00, 'Rendimento mês 1'),

(2, 2, 'Rendimento', 190.00, 'Rendimento mês 2'),

(3, 3, 'Compra', 4000.00, 'Compra investimento'),

(3, 3, 'Rendimento', 150.00, 'Rendimento mês 1'),

(3, 3, 'Rendimento', 160.00, 'Rendimento mês 2'),

(4, 4, 'Compra', 7000.00, 'Compra investimento'),

(4, 4, 'Rendimento', 300.00, 'Rendimento mês 1'),

(4, 4, 'Rendimento', 310.00, 'Rendimento mês 2'),

(5, 5, 'Depósito', 2000.00, 'Aporte inicial'),

(5, 5, 'Rendimento', 100.00, 'Rendimento mês 1'),

(5, 5, 'Rendimento', 110.00, 'Rendimento mês 2'),

(6, 6, 'Compra', 6000.00, 'Compra investimento'),

(6, 6, 'Rendimento', 350.00, 'Rendimento mês 1'),

(6, 6, 'Rendimento', 360.00, 'Rendimento mês 2'),

(7, 7, 'Depósito', 5500.00, 'Aporte inicial'),

(7, 7, 'Rendimento', 220.00, 'Rendimento mês 1'),

(7, 7, 'Rendimento', 230.00, 'Rendimento mês 2'),

(8, 8, 'Depósito', 3500.00, 'Aporte inicial'),

(8, 8, 'Rendimento', 140.00, 'Rendimento mês 1'),

(8, 8, 'Rendimento', 150.00, 'Rendimento mês 2'),

(9, 9, 'Compra', 8000.00, 'Compra investimento'),

(9, 9, 'Rendimento', 400.00, 'Rendimento mês 1'),

(9, 9, 'Rendimento', 410.00, 'Rendimento mês 2'),

(10, 10, 'Depósito', 2500.00, 'Aporte inicial'),

(10, 10, 'Rendimento', 130.00, 'Rendimento mês 1'),

(10, 10, 'Rendimento', 135.00, 'Rendimento mês 2'),

(11, 11, 'Depósito', 4500.00, 'Aporte inicial'),

(11, 11, 'Rendimento', 190.00, 'Rendimento mês 1'),

(11, 11, 'Rendimento', 195.00, 'Rendimento mês 2'),

(12, 12, 'Compra', 7500.00, 'Compra investimento'),

(12, 12, 'Rendimento', 380.00, 'Rendimento mês 1'),

(12, 12, 'Rendimento', 390.00, 'Rendimento mês 2'),

(13, 13, 'Depósito', 3000.00, 'Aporte inicial'),

(13, 13, 'Rendimento', 150.00, 'Rendimento mês 1'),

(13, 13, 'Rendimento', 155.00, 'Rendimento mês 2'),

(14, 14, 'Depósito', 3200.00, 'Aporte inicial'),

(14, 14, 'Rendimento', 170.00, 'Rendimento mês 1'),

(14, 14, 'Rendimento', 175.00, 'Rendimento mês 2'),

(15, 15, 'Compra', 8200.00, 'Compra investimento'),

(15, 15, 'Rendimento', 410.00, 'Rendimento mês 1'),

(15, 15, 'Rendimento', 420.00, 'Rendimento mês 2'),

(16, 16, 'Depósito', 2800.00, 'Aporte inicial'),

(16, 16, 'Rendimento', 140.00, 'Rendimento mês 1'),

(16, 16, 'Rendimento', 145.00, 'Rendimento mês 2'),

(17, 17, 'Depósito', 4700.00, 'Aporte inicial'),

(17, 17, 'Rendimento', 200.00, 'Rendimento mês 1'),

(17, 17, 'Rendimento', 210.00, 'Rendimento mês 2'),

(18, 18, 'Compra', 7300.00, 'Compra investimento'),

(18, 18, 'Rendimento', 380.00, 'Rendimento mês 1'),

(18, 18, 'Rendimento', 390.00, 'Rendimento mês 2'),

(19, 19, 'Depósito', 3100.00, 'Aporte inicial'),

(19, 19, 'Rendimento', 160.00, 'Rendimento mês 1'),

(19, 19, 'Rendimento', 165.00, 'Rendimento mês 2'),

(20, 20, 'Depósito', 3400.00, 'Aporte inicial'),

(20, 20, 'Rendimento', 180.00, 'Rendimento mês 1'),

(20, 20, 'Rendimento', 185.00, 'Rendimento mês 2');

Queries

-- 1. Total de investidores

SELECT COUNT(\*) AS total\_investidores FROM investidores;

-- 2. Quantidade de ativos por tipo

SELECT tipo, COUNT(\*) AS total\_ativos

FROM ativos

GROUP BY tipo

ORDER BY total\_ativos DESC;

-- 3. Total aportado por investidor

SELECT i.nome AS investidor, SUM(a.valor) AS total\_aportado

FROM investidores i

JOIN investimentos inv ON i.id = inv.investidor\_id

JOIN aportes a ON inv.id = a.investimento\_id

GROUP BY i.id, i.nome

ORDER BY total\_aportado DESC;

-- 4. Saldo total por investidor: (Aportes - Resgates + Rendimentos)

SELECT

i.nome AS investidor,

COALESCE(SUM(ap.valor),0) - COALESCE(SUM(rs.valor),0) + COALESCE(SUM(rd.valor),0) AS saldo\_total

FROM investidores i

LEFT JOIN investimentos inv ON i.id = inv.investidor\_id

LEFT JOIN aportes ap ON inv.id = ap.investimento\_id

LEFT JOIN resgates rs ON inv.id = rs.investimento\_id

LEFT JOIN rendimentos rd ON inv.id = rd.investimento\_id

GROUP BY i.id, i.nome

ORDER BY saldo\_total DESC;

-- 5. Lista de metas com status simples (atingida ou não)

SELECT

m.descricao,

i.nome AS investidor,

m.valor\_meta,

m.data\_alvo,

CASE

WHEN COALESCE(SUM(ap.valor),0) >= m.valor\_meta THEN 'Atingida'

ELSE 'Não atingida'

END AS status

FROM metas m

JOIN investidores i ON m.investidor\_id = i.id

LEFT JOIN investimentos inv ON i.id = inv.investidor\_id

LEFT JOIN aportes ap ON inv.id = ap.investimento\_id

GROUP BY m.id, m.descricao, i.nome, m.valor\_meta, m.data\_alvo;

-- 6. Rendimentos totais por tipo de ativo

SELECT

a.tipo,

SUM(r.valor) AS total\_rendimentos

FROM rendimentos r

JOIN investimentos inv ON r.investimento\_id = inv.id

JOIN ativos a ON inv.ativo\_codigo = a.codigo

GROUP BY a.tipo

ORDER BY total\_rendimentos DESC;

-- 7. Evolução mensal do saldo de investimento por investidor

-- (Saldo = aportes - resgates + rendimentos, agrupado por mês e investidor)

SELECT

i.nome AS investidor,

STRFTIME('%Y-%m', COALESCE(ap.data, rs.data, rd.data)) AS ano\_mes,

COALESCE(SUM(ap.valor),0) - COALESCE(SUM(rs.valor),0) + COALESCE(SUM(rd.valor),0) AS saldo\_mensal

FROM investidores i

LEFT JOIN investimentos inv ON i.id = inv.investidor\_id

LEFT JOIN aportes ap ON inv.id = ap.investimento\_id

LEFT JOIN resgates rs ON inv.id = rs.investimento\_id AND STRFTIME('%Y-%m', rs.data) = STRFTIME('%Y-%m', ap.data)

LEFT JOIN rendimentos rd ON inv.id = rd.investimento\_id AND STRFTIME('%Y-%m', rd.data) = STRFTIME('%Y-%m', ap.data)

GROUP BY i.id, ano\_mes

ORDER BY i.nome, ano\_mes;

-- 8. Top 5 ativos com melhor rendimento total

SELECT

a.codigo,

a.nome,

a.tipo,

SUM(r.valor) AS rendimento\_total

FROM rendimentos r

JOIN investimentos inv ON r.investimento\_id = inv.id

JOIN ativos a ON inv.ativo\_codigo = a.codigo

GROUP BY a.codigo, a.nome, a.tipo

ORDER BY rendimento\_total DESC

LIMIT 5;

-- 9. Investidores com maior saldo total (aportes - resgates + rendimentos), ordenados por saldo

SELECT

i.nome AS investidor,

COALESCE(SUM(ap.valor),0) - COALESCE(SUM(rs.valor),0) + COALESCE(SUM(rd.valor),0) AS saldo\_total

FROM investidores i

LEFT JOIN investimentos inv ON i.id = inv.investidor\_id

LEFT JOIN aportes ap ON inv.id = ap.investimento\_id

LEFT JOIN resgates rs ON inv.id = rs.investimento\_id

LEFT JOIN rendimentos rd ON inv.id = rd.investimento\_id

GROUP BY i.id, i.nome

ORDER BY saldo\_total DESC

LIMIT 10;

Dados

-- INVESTIDORES

INSERT INTO investidores (nome, email, perfil) VALUES

('Ana Sophia Araújo', 'ana01@exemplo.com', 'arrojado'),

('Renan Viana', 'renan02@exemplo.com', 'conservador'),

('Catarina Viana', 'catarina03@exemplo.com', 'conservador'),

('Otávio Pires', 'otavio04@exemplo.com', 'moderado'),

('Enzo Martins', 'enzo05@exemplo.com', 'conservador'),

('Thiago Farias', 'samuel18@uol.com.br', 'moderado'),

('Luana Almeida', 'amandaviana@terra.com.br', 'arrojado'),

('Lucas da Costa', 'camposaparecida@terra.com.br', 'moderado'),

('Yasmin Cavalcanti', 'apinto@terra.com.br', 'moderado'),

('Carolina Ribeiro', 'vianaalberto@gmail.com', 'conservador'),

('Henrique Almeida', 'felipe20@yahoo.com', 'moderado'),

('Amanda Gonçalves', 'rafaelalmeida@uol.com.br', 'moderado'),

('Júlia Rocha', 'valentinacosta@gmail.com', 'arrojado'),

('Bruno Lopes', 'vieiraalana@gmail.com', 'moderado'),

('Letícia Teixeira', 'lucas27@bol.com.br', 'conservador'),

('Eduardo Cardoso', 'pedrocarvalho@ig.com.br', 'moderado'),

('Beatriz Martins', 'beatrizsantos@yahoo.com.br', 'conservador'),

('Mateus Souza', 'nascimentoigor@gmail.com', 'arrojado'),

('Gabriel Castro', 'danielmedeiros@bol.com.br', 'moderado'),

('Camila Fernandes', 'maria06@uol.com.br', 'arrojado'),

('Marcelo Barbosa', 'robertoalmeida@terra.com.br', 'moderado'),

('Fernanda Lima', 'lucaspaulo@terra.com.br', 'conservador'),

('João Pedro Rezende', 'giovannagomes@yahoo.com', 'arrojado'),

('Isabela Azevedo', 'henrique56@gmail.com', 'moderado'),

('Ricardo Nogueira', 'lucasferreira@bol.com.br', 'conservador'),

('Viviane Moreira', 'joaogabrielcosta@ig.com.br', 'arrojado'),

('Mariana Duarte', 'duartevitoria@terra.com.br', 'moderado'),

('Vitor Gomes', 'lucasgomes@uol.com.br', 'conservador'),

('Patrícia Oliveira', 'patriciareis@terra.com.br', 'moderado'),

('André Torres', 'jtorres@yahoo.com', 'arrojado'),

('Nicole Ribeiro', 'nicole52@yahoo.com', 'conservador'),

('Diego Lima', 'andre65@yahoo.com', 'arrojado'),

('Clara Mendes', 'mendesclara@uol.com.br', 'moderado'),

('Sofia Silva', 'silvasofia@terra.com.br', 'conservador'),

('Murilo Rocha', 'mrocha@uol.com.br', 'moderado'),

('Helena Fernandes', 'hfernandes@terra.com.br', 'arrojado'),

('Leonardo Farias', 'lfarias@gmail.com', 'moderado'),

('Aline Braga', 'alinebraga@uol.com.br', 'arrojado'),

('Samuel Pinto', 'samueldp@gmail.com', 'conservador'),

('Daniel Rocha', 'danielrocha@uol.com.br', 'moderado');

-- INSTITUIÇÕES

INSERT INTO instituicoes (nome, cnpj) VALUES

('Banco do Brasil', '00000000000191'),

('Itaú', '60701190000104'),

('XP Investimentos', '02332886000199'),

('BTG Pactual', '30660784000191'),

('Clear', '09274280000120');

-- ATIVOS

INSERT INTO ativos (codigo, nome, tipo, instituicao\_id) VALUES

('PETR4', 'Petrobras PN', 'ação', 1),

('VALE3', 'Vale ON', 'ação', 1),

('ITUB4', 'Itaú Unibanco PN', 'ação', 2),

('BBDC4', 'Bradesco PN', 'ação', 2),

('SMAL11', 'SMALL CAPS', 'fundo', 3),

('IMAB11', 'Ima-B', 'fundo', 3),

('Tesouro Selic', 'Título Público Selic', 'renda fixa', 4),

('CDB Itaú', 'CDB Itaú', 'renda fixa', 2),

('FII XP', 'Fundo Imobiliário XP', 'fundo', 3),

('Bitcoin', 'Criptomoeda BTC', 'cripto', 5);

-- INVESTIMENTOS

INSERT INTO investimentos (investidor\_id, ativo\_codigo, data\_inicio) VALUES

(1, 'PETR4', '2023-01-10'),

(2, 'VALE3', '2022-11-05'),

(3, 'ITUB4', '2023-02-20'),

(4, 'BBDC4', '2022-10-15'),

(5, 'SMAL11', '2023-03-01'),

(6, 'IMAB11', '2022-12-12'),

(7, 'Tesouro Selic', '2023-01-22'),

(8, 'CDB Itaú', '2023-02-05'),

(9, 'FII XP', '2023-01-30'),

(10, 'Bitcoin', '2022-11-20'),

(11, 'PETR4', '2023-03-12'),

(12, 'VALE3', '2022-12-01'),

(13, 'ITUB4', '2023-01-17'),

(14, 'BBDC4', '2023-02-28'),

(15, 'SMAL11', '2023-03-05'),

(16, 'IMAB11', '2022-11-22'),

(17, 'Tesouro Selic', '2023-01-14'),

(18, 'CDB Itaú', '2023-02-11'),

(19, 'FII XP', '2023-02-22'),

(20, 'Bitcoin', '2023-01-08'),

(21, 'PETR4', '2023-02-19'),

(22, 'VALE3', '2023-03-02'),

(23, 'ITUB4', '2022-11-18'),

(24, 'BBDC4', '2023-02-08'),

(25, 'SMAL11', '2023-01-26'),

(26, 'IMAB11', '2023-02-15'),

(27, 'Tesouro Selic', '2023-03-10'),

(28, 'CDB Itaú', '2023-01-29'),

(29, 'FII XP', '2023-02-14'),

(30, 'Bitcoin', '2023-01-12'),

(31, 'PETR4', '2023-02-25'),

(32, 'VALE3', '2023-01-07'),

(33, 'ITUB4', '2023-02-16'),

(34, 'BBDC4', '2023-03-08'),

(35, 'SMAL11', '2023-01-19'),

(36, 'IMAB11', '2023-02-27'),

(37, 'Tesouro Selic', '2023-03-03'),

(38, 'CDB Itaú', '2023-02-23'),

(39, 'FII XP', '2023-01-15'),

(40, 'Bitcoin', '2023-02-06'),

(41, 'PETR4', '2023-01-24'),

(42, 'VALE3', '2023-02-02'),

(43, 'ITUB4', '2023-03-06'),

(44, 'BBDC4', '2023-02-12'),

(45, 'SMAL11', '2023-01-28');

-- APORTES

INSERT INTO aportes (investimento\_id, data, valor) VALUES

(1, '2023-01-15', 5000),

(2, '2022-11-10', 3000),

(3, '2023-02-25', 4000),

(4, '2022-10-20', 3500),

(5, '2023-03-05', 4500),

(6, '2022-12-20', 2500),

(7, '2023-01-30', 6000),

(8, '2023-02-10', 3000),

(9, '2023-02-05', 3500),

(10, '2022-11-25', 7000),

(11, '2023-03-15', 4000),

(12, '2022-12-10', 2800),

(13, '2023-01-25', 3200),

(14, '2023-03-01', 3600),

(15, '2023-03-07', 4000),

(16, '2022-11-25', 2300),

(17, '2023-01-20', 5500),

(18, '2023-02-15', 3000),

(19, '2023-02-28', 3700),

(20, '2023-01-15', 6800),

(21, '2023-03-02', 4200),

(22, '2023-03-05', 3100),

(23, '2022-11-20', 3300),

(24, '2023-02-12', 3800),

(25, '2023-01-30', 4400),

(26, '2023-02-20', 2900),

(27, '2023-03-15', 6000),

(28, '2023-01-31', 3200),

(29, '2023-02-18', 3500),

(30, '2023-01-18', 6900),

(31, '2023-02-28', 4100),

(32, '2023-01-10', 3000),

(33, '2023-02-20', 3600),

(34, '2023-03-10', 3700),

(35, '2023-01-22', 4300),

(36, '2023-03-01', 3000),

(37, '2023-03-05', 5800),

(38, '2023-02-25', 3100),

(39, '2023-01-16', 3400),

(40, '2023-02-10', 7000),

(41, '2023-01-28', 3900),

(42, '2023-02-05', 3100),

(43, '2023-03-07', 3800),

(44, '2023-02-15', 3500),

(45, '2023-01-30', 4000);

-- RESGATES

INSERT INTO resgates (investimento\_id, data, valor) VALUES

(1, '2023-06-01', 1000),

(3, '2023-06-10', 500),

(5, '2023-06-15', 700);

-- RENDIMENTOS

INSERT INTO rendimentos (investimento\_id, data, valor) VALUES

(1, '2023-06-30', 150),

(2, '2023-06-30', 120),

(3, '2023-06-30', 130),

(5, '2023-06-30', 90);

-- COTAÇÕES

INSERT INTO cotacoes (ativo\_codigo, data, preco) VALUES

('PETR4', '2023-07-01', 25.50),

('VALE3', '2023-07-01', 65.30),

('ITUB4', '2023-07-01', 28.75),

('BBDC4', '2023-07-01', 19.20),

('SMAL11', '2023-07-01', 120.50),

('IMAB11', '2023-07-01', 105.30),

('Tesouro Selic', '2023-07-01', 1.00),

('CDB Itaú', '2023-07-01', 1.00),

('FII XP', '2023-07-01', 100.00),

('Bitcoin', '2023-07-01', 30000.00);

-- METAS

INSERT INTO metas (investidor\_id, descricao, valor\_meta, data\_alvo) VALUES

(1, 'Aposentadoria', 100000, '2035-12-31'),

(2, 'Compra de Imóvel', 50000, '2028-06-30'),

(3, 'Viagem Europa', 20000, '2025-07-15'

Schema

CREATE TABLE investidores (

id INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

email TEXT UNIQUE NOT NULL,

perfil TEXT CHECK(perfil IN ('conservador', 'moderado', 'arrojado')) NOT NULL

);

CREATE TABLE instituicoes (

id INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

cnpj TEXT UNIQUE NOT NULL

);

CREATE TABLE ativos (

codigo TEXT PRIMARY KEY,

nome TEXT NOT NULL,

tipo TEXT CHECK(tipo IN ('ação', 'fundo', 'renda fixa', 'cripto')) NOT NULL,

instituicao\_id INTEGER NOT NULL,

FOREIGN KEY (instituicao\_id) REFERENCES instituicoes(id)

);

CREATE TABLE investimentos (

id INTEGER PRIMARY KEY AUTOINCREMENT,

investidor\_id INTEGER NOT NULL,

ativo\_codigo TEXT NOT NULL,

data\_inicio DATE NOT NULL,

FOREIGN KEY (investidor\_id) REFERENCES investidores(id),

FOREIGN KEY (ativo\_codigo) REFERENCES ativos(codigo)

);

CREATE TABLE aportes (

id INTEGER PRIMARY KEY AUTOINCREMENT,

investimento\_id INTEGER NOT NULL,

data DATE NOT NULL,

valor REAL NOT NULL CHECK (valor >= 0),

FOREIGN KEY (investimento\_id) REFERENCES investimentos(id)

);

CREATE TABLE resgates (

id INTEGER PRIMARY KEY AUTOINCREMENT,

investimento\_id INTEGER NOT NULL,

data DATE NOT NULL,

valor REAL NOT NULL CHECK (valor >= 0),

FOREIGN KEY (investimento\_id) REFERENCES investimentos(id)

);

CREATE TABLE rendimentos (

id INTEGER PRIMARY KEY AUTOINCREMENT,

investimento\_id INTEGER NOT NULL,

data DATE NOT NULL,

valor REAL NOT NULL CHECK (valor >= 0),

FOREIGN KEY (investimento\_id) REFERENCES investimentos(id)

);

CREATE TABLE cotacoes (

id INTEGER PRIMARY KEY AUTOINCREMENT,

ativo\_codigo TEXT NOT NULL,

data DATE NOT NULL,

preco REAL NOT NULL CHECK(preco >= 0),

FOREIGN KEY (ativo\_codigo) REFERENCES ativos(codigo)

);

CREATE TABLE metas (

id INTEGER PRIMARY KEY AUTOINCREMENT,

investidor\_id INTEGER NOT NULL,

descricao TEXT NOT NULL,

valor\_meta REAL NOT NULL CHECK(valor\_meta >= 0),

data\_alvo DATE NOT NULL,

FOREIGN KEY (investidor\_id) REFERENCES investidores(id)

);

Database

