

Relatório sobre o Sistema de Gerenciamento Esportivo

1. Introdução

O Sistema de Gerenciamento Esportivo foi concebido para atender às demandas intrincadas e diversificadas inerentes ao universo esportivo. Desde o controle preciso de transações financeiras até a gestão eficaz de atletas, treinadores e funcionários, cada aspecto do sistema foi meticulosamente elaborado para garantir uma administração eficiente e transparente. Este relatório oferece uma visão abrangente da estrutura e do funcionamento do banco de dados subjacente ao sistema, destacando tabelas-chave, gatilhos (triggers), visões (views) e proporcionando uma compreensão detalhada do ciclo de trabalho.

1.1 Objetivo do Sistema

O Sistema de Gerenciamento Esportivo tem como principal objetivo fornecer uma plataforma unificada para a gestão integrada de informações relacionadas a atletas, transações financeiras, treinadores e demais elementos essenciais no contexto esportivo. A estrutura do banco de dados é projetada para facilitar a tomada de decisões informadas, promover a eficiência operacional e garantir uma experiência contínua e bem gerenciada para todos os envolvidos.

1.2 Estrutura Modular

A modularidade é uma característica fundamental do sistema, permitindo uma fácil expansão e personalização conforme as necessidades evoluem. Cada entidade no sistema é representada por tabelas independentes, com relacionamentos cuidadosamente estabelecidos para refletir a dinâmica complexa do mundo esportivo. Isso proporciona flexibilidade para acomodar diferentes modalidades esportivas, categorias e tipos de transações, garantindo que o sistema seja adaptável a diferentes contextos.

1.3 Ciclo de Trabalho

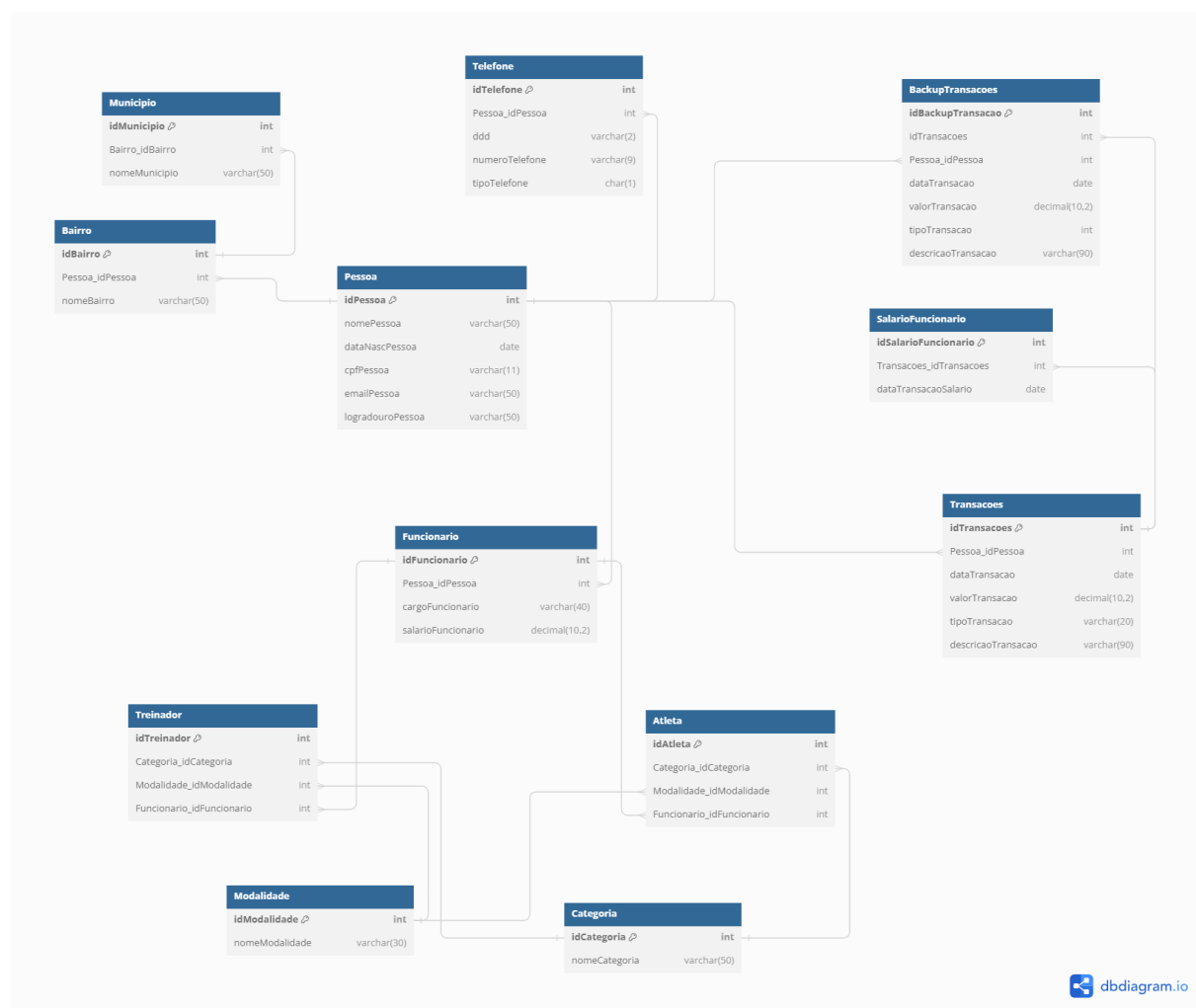
O ciclo de trabalho no Sistema de Gerenciamento Esportivo engloba desde a inserção de dados iniciais até a execução de ações automáticas por meio de gatilhos. A interação dos usuários, seja na adição de informações de transações financeiras, na inclusão de novos atletas ou na atualização de dados de treinadores, desencadeia uma série de eventos no banco de dados. Gatilhos e procedimentos armazenados são essenciais para aplicar regras de negócios, garantindo a integridade dos dados e a consistência das informações.

O relatório seguirá detalhando gatilhos, visões e oferecendo uma análise aprofundada dos componentes fundamentais do Sistema de Gerenciamento Esportivo.

2. Estrutura do Banco de Dados

O banco de dados do Sistema de Gerenciamento Esportivo foi projetado para refletir a complexidade e as inter-relações presentes no ambiente esportivo. A estrutura é composta por diversas tabelas, cada uma desempenhando um papel específico na gestão abrangente de informações. Abaixo, são detalhadas as principais tabelas do banco de dados:

DER:



Pessoa

```
CREATE TABLE Pessoa (
    idPessoa SERIAL PRIMARY KEY,
    nomePessoa VARCHAR(50),
    dataNascPessoa DATE,
    cpfPessoa VARCHAR(11),
    emailPessoa VARCHAR(50),
    logradouroPessoa VARCHAR(50)
);
```

Armazena informações básicas sobre indivíduos, como nome, data de nascimento, CPF, email e endereço.

Modalidade

```
CREATE TABLE Modalidade (
    idModalidade SERIAL PRIMARY KEY,
    nomeModalidade VARCHAR(30)
);
```

Contém dados sobre as modalidades esportivas existentes, como Futebol, Vôlei, Natação, Basquete, Handebol e Judô.

Categoria

```
CREATE TABLE Categoria (
    idCategoria SERIAL PRIMARY KEY,
    nomeCategoria VARCHAR(50)
);
```

Registra as categorias esportivas, tais como Sub13, Sub15, Sub18, Sub23 e Adulto.

Transações

```
CREATE TABLE Transacoes (
    idTransacoes SERIAL PRIMARY KEY,
    Pessoa_idPessoa INTEGER,
    dataTransacao DATE,
    valorTransacao DECIMAL(10,2),
    tipoTransacao INTEGER,
    descricaoTransacao VARCHAR(90),
    FOREIGN KEY (Pessoa_idPessoa)
        REFERENCES Pessoa(idPessoa)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);
```

Rastreia transações financeiras associadas a pessoas, incluindo data, valor, tipo e descrição.

Telefone

```
CREATE TABLE Telefone (  
    idTelefone SERIAL PRIMARY KEY  
    Pessoa_idPessoa INTEGER,  
    ddd VARCHAR(2),  
    numeroTelefone VARCHAR(9),  
    tipoTelefone CHAR(1),  
    FOREIGN KEY (Pessoa_idPessoa)  
        REFERENCES Pessoa(idPessoa)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
);
```

Mantém registros de números de telefone associados a pessoas.

Funcionario

```
CREATE TABLE Funcionario (  
    idFuncionario SERIAL PRIMARY KEY,  
    Pessoa_idPessoa INTEGER,  
    cargoFuncionario VARCHAR(40),  
    salarioFuncionario DECIMAL(10,2),  
    FOREIGN KEY (Pessoa_idPessoa)  
        REFERENCES Pessoa(idPessoa)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
);
```

Armazena dados sobre funcionários, como cargo e salário.

Bairro

```
CREATE TABLE Bairro (  
    idBairro SERIAL PRIMARY KEY,  
    Pessoa_idPessoa INTEGER,  
    nomeBairro VARCHAR(50),  
    FOREIGN KEY (Pessoa_idPessoa)  
        REFERENCES Pessoa(idPessoa)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
);
```

Relaciona pessoas a bairros específicos.

Treinador

```

CREATE TABLE Treinador (
  idTreinador SERIAL PRIMARY KEY,
  Categoria_idCategoria INTEGER,
  Modalidade_idModalidade INTEGER,
  Funcionario_idFuncionario INTEGER,
  FOREIGN KEY (Funcionario_idFuncionario)
    REFERENCES Funcionario(idFuncionario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (Modalidade_idModalidade)
    REFERENCES Modalidade(idModalidade)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (Categoria_idCategoria)
    REFERENCES Categoria(idCategoria)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

Associa treinadores a modalidades e categorias.

Atleta

```

CREATE TABLE Atleta (
  idAtleta SERIAL PRIMARY KEY,
  Categoria_idCategoria INTEGER,
  Modalidade_idModalidade INTEGER,
  Funcionario_idFuncionario INTEGER,
  FOREIGN KEY (Funcionario_idFuncionario)
    REFERENCES Funcionario(idFuncionario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (Modalidade_idModalidade)
    REFERENCES Modalidade(idModalidade)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (Categoria_idCategoria)
    REFERENCES Categoria(idCategoria)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

Relaciona atletas a modalidades e categorias.

SalarioFuncionario

```
CREATE TABLE SalarioFuncionario (
  idSalarioFuncionario SERIAL PRIMARY KEY,
  Transacoes_idTransacoes INTEGER,
  dataTransacaoSalario DATE,
  FOREIGN KEY (Transacoes_idTransacoes)
    REFERENCES Transacoes(idTransacoes)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

Registra informações sobre salários de funcionários.

Município

```
CREATE TABLE Municipio (
  idMunicipio SERIAL PRIMARY KEY,
  Bairro_idBairro INTEGER,
  nomeMunicipio VARCHAR(50),
  FOREIGN KEY (Bairro_idBairro)
    REFERENCES Bairro(idBairro)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

Associa bairros a municípios.

backupTransacoes:

```
CREATE TABLE BackupTransacoes (
  idBackup SERIAL PRIMARY KEY, |
  idTransacoes INTEGER,
  Pessoa_idPessoa INTEGER,
  dataTransacao DATE,
  valorTransacao DECIMAL(10,2),
  tipoTransacao VARCHAR(50),
  descricaoTransacao VARCHAR(90),
  FOREIGN KEY (Pessoa_idPessoa)
    REFERENCES Pessoa(idPessoa)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (idTransacoes)
    REFERENCES Transacoes(idTransacoes)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

Realiza o backup de informações adicionais sobre as transações feitas a partir de um trigger.

3. Ciclo de trabalho

O Sistema de Gerenciamento Esportivo é estruturado com tabelas que abrangem desde informações básicas de pessoas até registros financeiros e categorias esportivas. O ciclo de trabalho envolve a inserção de dados, a atualização de funcionários, a realização de transações financeiras e a consulta de informações por meio de visões especializadas. A seguir, serão detalhadas cada etapa do ciclo:

3.1 Inserção de Dados Iniciais

Tabelas Envolvidas: Modalidade, Pessoa, Categoria, Transacoes, Telefone, Funcionario, Bairro, Treinador, Atleta, SalarioFuncionario, Municipio.

Descrição:

- A inserção de dados é iniciada com informações fundamentais, como modalidades esportivas, dados pessoais, categorias e registros financeiros.
- Os dados iniciais incluem detalhes sobre pessoas, como nome, data de nascimento, CPF, e endereço.
- As categorias são inseridas, representando faixas etárias ou grupos específicos.
- A tabela Transacoes registra movimentações financeiras, incluindo dados como valor, tipo de transação e descrição.
- Informações adicionais, como telefones, funcionários, bairros, treinadores e atletas, são inseridas para criar uma base completa.

Atualização de Funcionários e Cargos:

Tabelas Envolvidas: Funcionario, Atleta, Treinador.

Descrição: Os funcionários que possuírem o cargo atleta ou treinador devem ser inseridos como atleta ou treinador ligados a sua modalidade e categoria.

Registro de Transações Financeiras:

Tabelas Envolvidas: Transacoes, SalarioFuncionario.

Descrição:

- Transações financeiras, como pagamentos de salários, compras de equipamentos ou taxas de inscrição, são registradas na tabela Transacoes.
- Gatilhos garantem que valores não negativos sejam inseridos e validam a consistência das informações.

```

CREATE OR REPLACE FUNCTION verificar_valor_transacao()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.valorTransacao < 0 THEN
        RAISE EXCEPTION 'O valor da transação não pode ser negativo';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER trigger_verificar_valor_transacao
BEFORE INSERT ON Transacoes
FOR EACH ROW
EXECUTE FUNCTION verificar_valor_transacao();

```

Atualização Automática da Categoria de Atletas:

Tabelas Envolvidas: Atleta.

Descrição:

- Um gatilho automático associado à tabela Atleta atualiza dinamicamente a categoria de atletas com base em sua data de nascimento.
- A categoria é determinada conforme faixas etárias predefinidas.


```

CREATE OR REPLACE FUNCTION verificar_categoria_atleta()
RETURNS TRIGGER AS $$
DECLARE
    idade_atleta INTEGER;
    categoria_atleta INTEGER;
BEGIN

    idade_atleta := EXTRACT(YEAR FROM AGE((
        SELECT dataNascPessoa
        FROM Pessoa
        WHERE idPessoa = (SELECT Pessoa_idPessoa FROM Funcionario WHERE idFuncionario =
    )));
    -- Determinar a categoria com base na idade
    IF idade_atleta <= 13 THEN
        categoria_atleta := 11; -- Substitua com o ID correto para 'Sub13'
    ELSIF idade_atleta <= 15 THEN
        categoria_atleta := 12; -- Substitua com o ID correto para 'Sub15'
    ELSIF idade_atleta <= 18 THEN
        categoria_atleta := 13; -- Substitua com o ID correto para 'Sub18'
    ELSIF idade_atleta <= 23 THEN
        categoria_atleta := 14; -- Substitua com o ID correto para 'Sub23'
    ELSE
        categoria_atleta := 15; -- Substitua com o ID correto para 'Adulto'
    END IF;

    END IF;
    -- Verificar se a categoria está correta e gerar uma exceção se não estiver
    IF categoria_atleta <> NEW.Categoria_idCategoria THEN
        RAISE EXCEPTION 'A categoria do atleta está incorreta. Esperada: %, /',
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

```

```

CREATE TRIGGER trigger_verificar_categoria_atleta
BEFORE INSERT OR UPDATE ON atleta
FOR EACH ROW
EXECUTE FUNCTION verificar_categoria_atleta();

```

Backup automático de transações:

Tabelas Envolvidas: Transacoes, backupTransacoes.

Descrição: um gatilho atualiza a tabela backupTransacoes a partir de uma transação realizada.

```

CREATE OR REPLACE FUNCTION realizar_backup()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO BackupTransacoes (idTransacoes, Pessoa_idPessoa, dataTransacao)
    VALUES (NEW.idTransacoes, NEW.Pessoa_idPessoa, NEW.dataTransacao, NEW.dataTransacao);

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER backup_transacoes_trigger
AFTER INSERT ON Transacoes
FOR EACH ROW
EXECUTE FUNCTION realizar_backup();

```

Consultas e Visualizações:

Visões Disponíveis:

view_atletas_categorias

view_funcionarios_salarios

view_time

Descrição:

- Usuários podem realizar consultas e visualizações por meio de visões especializadas.
- A visão visao_atletas_categorias permite visualizar os atletas agrupados por suas categorias, proporcionando uma visão clara da distribuição etária no sistema.
- A visão visao_funcionarios_salarios oferece uma visão consolidada dos funcionários e seus salários.
- Adicionada a nova view visao_time, que associa treinadores e atletas com base em uma categoria e modalidade em comum, proporcionando uma perspectiva mais abrangente das equipes no sistema.

Este ciclo de trabalho abrange desde a inserção inicial de dados até as atualizações automáticas e consultas especializadas, garantindo um gerenciamento eficiente e consistente no contexto esportivo.

4. Conclusão

O Sistema de Gerenciamento Esportivo demonstra ser uma ferramenta robusta e abrangente, proporcionando uma gestão eficaz de informações

relacionadas a atletas, treinadores, transações financeiras e categorias esportivas. Suas utilidades são vastas, indo desde a organização de eventos esportivos até o acompanhamento financeiro e categorização dinâmica dos atletas.

Ao longo deste ciclo de trabalho, destacamos a importância das atualizações automáticas, gatilhos e consultas especializadas, proporcionando aos usuários uma experiência consistente e completa no gerenciamento esportivo. A introdução de visões como "Visao_Time" ampliou a capacidade de visualização das equipes, associando treinadores e atletas com base em categorias e modalidades comuns.

Possíveis Implementações Futuras:

Controle de Mensalidades:

- Integração de um sistema de controle de mensalidades para atletas, permitindo o registro e acompanhamento de pagamentos mensais. Possibilidade de automatizar lembretes e notificações para atletas em atraso, garantindo uma gestão financeira mais eficiente.

Introdução de Membros:

- Implementação de uma tabela de Membros para representar indivíduos associados ao clube ou equipe esportiva.
- Atribuição de papéis específicos, como membro comum, torcedor, ou apoiador, ampliando as possibilidades de engajamento.

Controle de Times:

- Expansão do sistema para incluir um módulo de controle de times, possibilitando a criação e gestão de diferentes equipes dentro das modalidades esportivas.
- Associação dinâmica de atletas e treinadores aos times, proporcionando uma visão mais detalhada e estruturada das competições.

Aprimoramento da Interface:

- Desenvolvimento de interfaces mais amigáveis e intuitivas para facilitar a interação dos usuários, tornando o sistema mais acessível a diferentes perfis, desde gestores esportivos até atletas.

Relatórios Avançados:

- Implementação de relatórios mais avançados e personalizáveis, permitindo aos usuários gerar análises detalhadas sobre o desempenho financeiro, categorias esportivas e a participação de atletas em diferentes eventos.

O Sistema de Gerenciamento Esportivo, ao evoluir com essas implementações, tem o potencial de se tornar uma solução ainda mais abrangente e adaptável às necessidades dinâmicas do cenário esportivo, proporcionando uma gestão eficiente e integrada para todos os envolvidos.