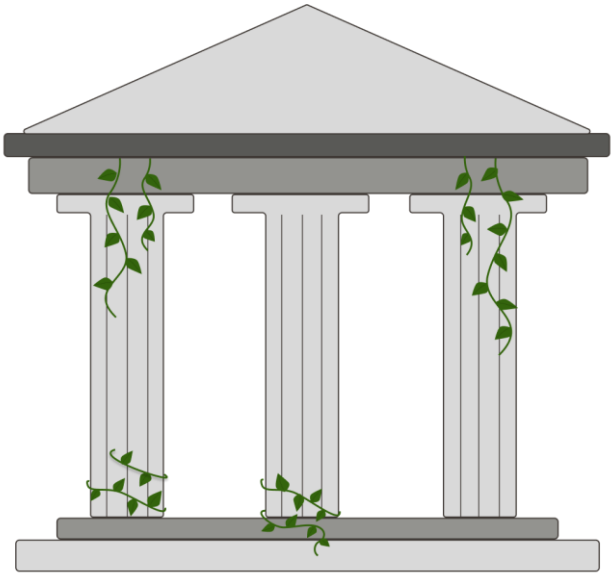
ATHLETA

Relatório modelagem de dados: Normalização para o aplicativo ATHLETA

PANATEM
Raika Raiane Sousa da Silva - 16
Stephany Andreina de Carvalho da Silva - 28



PANATEM

Introdução

Visão geral do aplicativo

Nosso aplicativo tem como principal objetivo promover união e bem-estar através dos esportes, incentivando sua prática, especialmente de modalidades menos conhecidas, para ampliar opções e contribuir para o crescimento e diversificação do cenário esportivo nacional, tornando-o acessível, inclusivo e prazeroso para todos. Nosso compromisso é com a formação de valores, o respeito ao próximo, e a promoção de uma cultura esportiva que valorize a diversidade e o espírito de equipe, proporcionando oportunidades para que todos possam participar e se desenvolver em um ambiente acolhedor e inspirador.

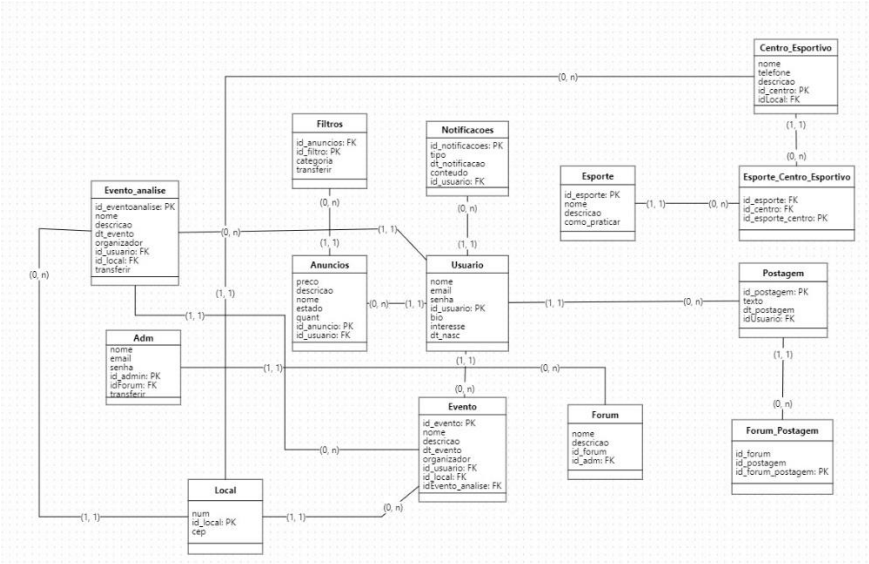
Objetivo do nosso banco de dados

O objetivo do nosso banco de dados é organizar, armazenar e gerenciar informações de forma eficiente e segura, permitindo o acesso, a manipulação e a recuperação dos dados de maneira rápida e precisa. É essencial para suportar operações, facilitar a tomada de decisões e garantir a integridade e a consistência das informações, especialmente em ambientes onde grandes volumes de dados precisam ser gerenciados e utilizados para diversas finalidades, como consultas, relatórios e análises.

Normalização do banco de dados

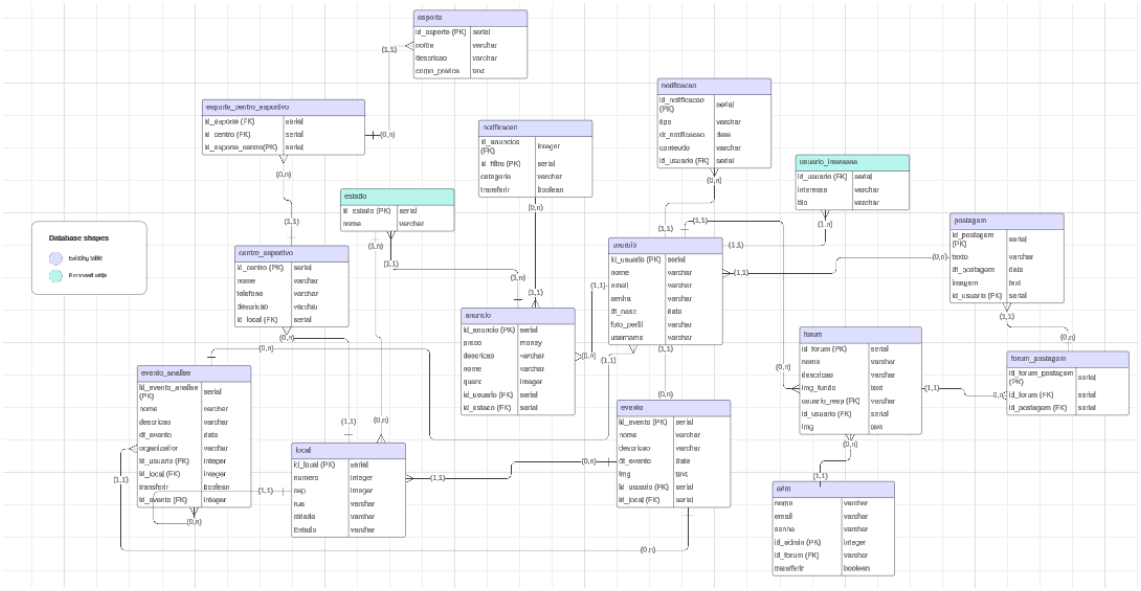
Normalizamos o banco de dados para estruturar os dados de forma eficiente, eliminando redundâncias e inconsistências. Isso melhora a integridade e a eficiência das operações, facilitando a manutenção e a escalabilidade do banco de dados, além de garantir que os dados sejam armazenados de maneira lógica e otimizada.

Nosso primeiro modelo, que não foi normalizado, contém 14 tabelas: Usuário, Anúncios, Notificações, Evento, Local, Centro_Esportivo, Esporte_Centro_Esportivo, Esporte, Postagem, Forum_Postagem, Forum, Adm, Evento_analise e Filtros.



Modelo sem normalização

No nosso modelo normalizado, adicionamos duas novas tabelas: Estado e Usuario_Interesse. Reestruturamos as tabelas, redistribuindo alguns campos para otimizar a organização dos dados. Por exemplo, o campo nome_estado, que antes estava na tabela Usuario, agora foi movido para a nova tabela Estado, que se relaciona tanto com a tabela Usuario quanto com a tabela Local. Além disso, os campos interesse e bio foram transferidos para a tabela Usuario_Interesse, que estabelece uma ligação com a tabela Usuario.



Modelo normalizado

Procedure, Trigger e Log

Procedure: inserir_usuario_e_interesse

Essa procedure faz duas coisas:

- Insere um novo usuário na tabela Usuario, com os dados fornecidos como parâmetros: nome, email, senha e data de nascimento.
- Insere o interesse do usuário recém-criado na tabela Usuario_Interesse, utilizando a sequência currval para obter o ID do usuário inserido e associá-lo ao interesse e à bio informados.

Essa procedure facilita a criação de novos usuários e o vínculo imediato de seus interesses no sistema.

Procedure: inserir_anuncio_e_notificacao

Essa procedure realiza as seguintes ações:

- Insere um anúncio na tabela Anuncio, utilizando os parâmetros fornecidos como preço, descrição, nome do produto, quantidade, ID do usuário e ID do estado.
- Cria uma notificação para o usuário que está publicando o anúncio, inserindo uma nova entrada na tabela Notificacao, com o tipo, data atual e conteúdo da notificação.

Ela automatiza a criação de um anúncio e a geração de uma notificação correspondente para o usuário.

Procedure: inserir_evento_com_centro_esportivo

Esta procedure executa duas operações:

- Insere um evento na tabela Evento, com o nome, descrição, data, organizador, imagem e ID do local fornecidos.
- Insere um novo centro esportivo relacionado ao evento, associando o nome, telefone, descrição e o ID do local na tabela Centro_Esportivo.

Essa procedure é usada para garantir que, ao criar um evento, o centro esportivo correspondente também seja registrado.

Tabela de Log: Log_Usuario

Essa tabela captura as seguintes informações sobre as operações realizadas na tabela Usuario:

- ID do usuário, nome, email, senha, data de nascimento.
- A operação realizada (inserção, atualização ou exclusão).
- Data e hora em que a operação ocorreu.

Serve como um registro histórico das alterações feitas nos usuários, garantindo rastreabilidade.

Tabela de Log: Log_Anuncio

Semelhante à Log_Usuario, essa tabela registra operações realizadas na tabela Anuncio, armazenando informações como:

- ID do anúncio, preço, descrição, nome do produto, quantidade, ID do usuário e ID do estado.
- Tipo de operação (inserção, atualização ou exclusão).
- Data e hora da operação.

Tabela de Log: Log_Evento

Armazena as operações realizadas na tabela Evento:

- ID do evento, nome, descrição, data, organizador, imagem e ID do local.
- Operação (inserção, atualização ou exclusão).
- Data e hora da operação.

Trigger: usuario_log_trigger

Essa trigger aciona a função log_usuario toda vez que um registro da tabela Usuario é inserido, atualizado ou deletado. Ela insere as informações correspondentes na tabela Log_Usuario, garantindo que qualquer modificação seja registrada.

Trigger: anuncio_log_trigger

Essa trigger funciona da mesma maneira que a usuario_log_trigger, mas é aplicada à tabela Anuncio. Ela chama a função log_anuncio sempre que um anúncio é criado, modificado ou excluído, registrando o evento na tabela Log_Anuncio.

Trigger: evento_log_trigger

De modo semelhante, essa trigger é associada à tabela Evento. Ela chama a função log_evento em caso de inserções, atualizações ou exclusões, registrando o evento na tabela Log_Evento.

Trigger: atualizar_quantidade_anuncio

Essa trigger é ativada antes de qualquer atualização na tabela Anuncio. Se a quantidade do anúncio for reduzida (i.e., o novo valor for menor que o antigo), uma mensagem de aviso é gerada usando o comando RAISE NOTICE.
Ela é útil para alertar o sistema ou os administradores sobre reduções na quantidade de produtos disponíveis nos anúncios.

Tabela de Log: Log_Postagem

Registra as operações realizadas na tabela Postagem, incluindo:

- ID da postagem, texto, data de postagem, ID do usuário, e imagem.
- Operação realizada (inserção, atualização ou exclusão).

- Data e hora do evento.

Tabela de Log: Log_Forum

Captura informações sobre as operações na tabela Forum, como:

- ID do fórum, nome, descrição, responsável pelo fórum, ID do usuário e imagem de fundo.
- Operação realizada (inserção, atualização ou exclusão).
- Data e hora do evento.

Tabela de Log: Log_Esporte_Centro_Esportivo

Registra as operações feitas na tabela Esporte_Centro_Esportivo, que associa esportes a centros esportivos. Armazena o ID da relação, o ID do esporte, o ID do centro esportivo e a operação realizada (inserção, atualização ou exclusão).

Trigger: postagem_log_trigger

Acionada para registrar operações na tabela Postagem. Quando uma postagem é inserida, atualizada ou excluída, a função log_postagem é chamada, que insere essas informações na tabela Log_Postagem.

Trigger: forum_log_trigger

Essa trigger é responsável por registrar as operações realizadas na tabela Forum, chamando a função log_forum para gravar os dados na tabela de log associada (Log_Forum).

Trigger: esporte_centro_log_trigger

Essa trigger é executada quando há mudanças na tabela Esporte_Centro_Esportivo. Ela chama a função log_esporte_centro_esportivo para registrar inserções, atualizações ou exclusões de registros, permitindo rastreamento dessas operações.