



UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO

Projeto de Banco de Dados - 2025/2
Prof. Marcos Bedo

TRABALHO FINAL DE DISCIPLINA

Avisos Importantes:

- Trabalho vale 50% da Nota Final!
- Trabalho a ser realizado em grupo!
- Não compartilhem suas respostas. Haverá verificação de cópia entre os trabalhos (inclusive implementação). Vocês podem ser os maiores prejudicados em casos de plágio!
- Não deixem de enviar o relatório por e-mail no prazo especificado em aula!

Especificação do Trabalho - Projeto de BD incrementando Princípios de BD

Implementar um projeto de banco de dados relacional **já mapeado para o relacional** onde será necessário:

1. Definir a construção de Visões Virtuais e Materializadas
2. Definir a construção de índices para melhorar o desempenho de buscas
3. Definir triggers para a movimentação de dados
4. Definir rotinas em PL/SQL para consultas aos dados via Linguagem Procedural

Deverá ser entregue um .zip com as implementações realizadas em Postgres (`$ docker install postgres`), um arquivo README.md com o passo-a-passo para reproduzir a solução do grupo e um **pequeno relatório justificando as escolhas de implementação baseados nos conceitos vistos em aula.**

Requisitos do Banco de Dados

Deseja-se projetar uma base de dados para catalogar informações sobre *streamers* de jogos *online* para responder questões sobre o alcance de público, quantidade de propagandas

realizadas, arrecadação direta (doações), arrecadação indireta (membros inscritos), número de interações diretas (comentários *online*) e número de interações (comentários *online* + *offline*).

Construa o modelo ER referente à base de dados. A base de dados não deve conter redundância de dados. O modelo ER pode ser representado com a notação vista nas aulas. Mapeie o modelo ER para o modelo Relacional. Normalize o modelo relacional mapeado.

A base de dados deve incluir plataforma de *streaming* (ex., *Youtube* e *Twitch*), que são identificadas por um número sequencial único e possuem um nome, uma empresa fundadora, uma empresa responsável, uma data de fundação e uma quantidade de usuários.

Além disso, a base de dados deve incluir *streamers*, identificados por um único *nick*, número de usuário (que pode ser diferente em cada plataforma), endereço de email, data de nascimento e endereço postal, número de passaporte e país de passaporte. A base de dados também deve conter informações de usuários, identificados por um único *nick*, endereço de email, data de nascimento, um telefone para validação em duas vias, um endereço postal e um país de residência.

Cada país deve ser identificado por um número (DDI) e possuir um nome e uma moeda. Além disso, para cada moeda deve estar associada uma fator de conversão em dólar padrão (USD - dólares americanos) para conversão do valor da moeda estrangeira para dólares americanos.

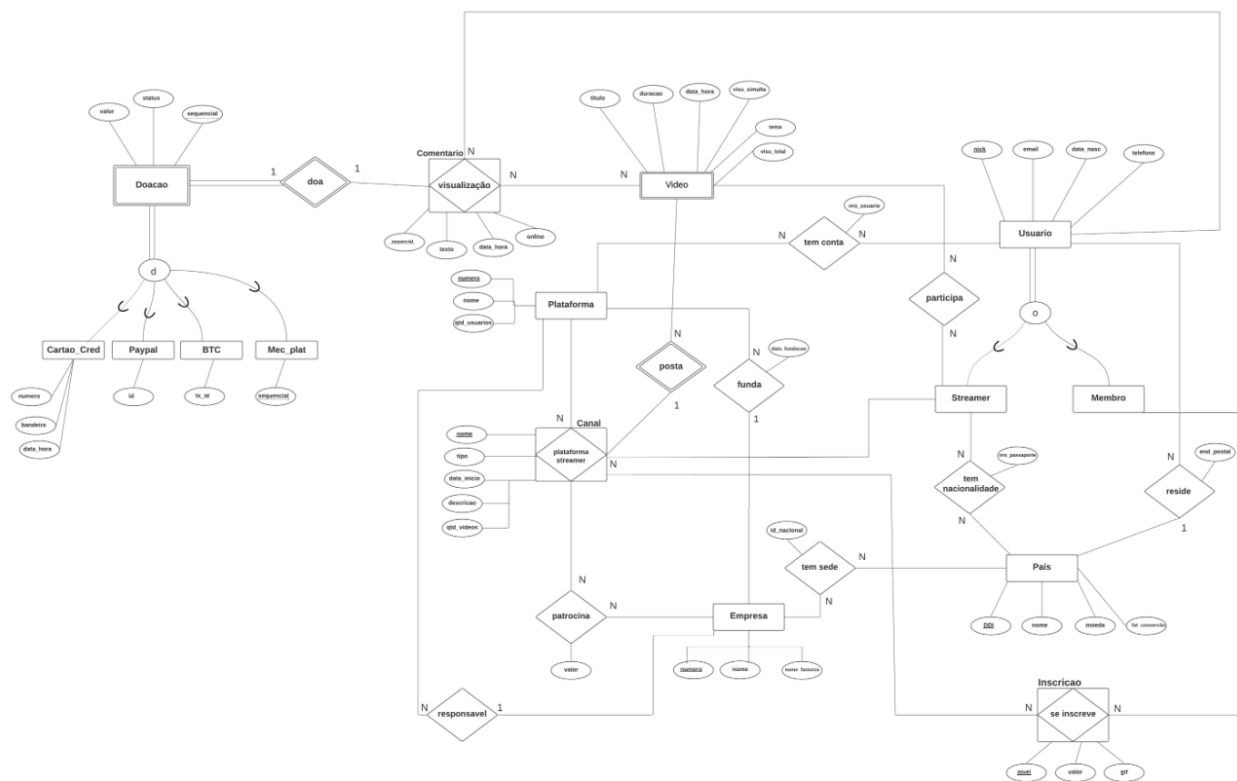
Um ou mais canais estão associados a cada *streamers* em diferentes plataformas. Cada canal é identificado por seu nome único, data de início, descrição, quantidade de vídeos postados e tipo do canal que deve ser um entre {privado, público ou misto}. Um canal pode ser patrocinado por uma empresa, que é identificada um número sequencial único, um nome, um identificador nacional (ex: CNPJ), país de origem e um nome fantasia. Cada patrocínio tem um valor (em dólares americanos) e o sistema não armazena o histórico de patrocínios, ou seja, apenas os patrocinadores com patrocínios vigentes devem aparecer nos sistema. Um canal pode ter vários membros inscritos. Cada membro inscrito contribui com um valor mensal que depende do seu nível de membro. Existem cinco níveis de membros e os nomes e os valores de cada nível são definidos para cada canal e para cada nível existe um arquivo de imagem .gif associado. O sistema não armazena o histórico de membros, ou seja, apenas os membros vigentes devem aparecer no sistema.

Um vídeo postado é identificado por seu título e data/hora de postagem. Cada vídeo tem uma duração e está associado a um único canal, mas pode ter a participação de mais *streamers* do que apenas o dono do canal. Cada vídeo tem um tema, uma quantidade máxima de visualizações simultâneas e a quantidade total de visualizações. Cada vídeo pode ter comentários *online* e *offline*. Cada comentário tem um número sequencial que depende do vídeo e tem um usuário

(que fez o comentário) associado. Cada comentário tem uma data/hora e pode trazer consigo uma doação ou não. Cada doação é identificada por um número sequencial que depende do vídeo, do usuário e do comentário. A doação pode ser feita de quatro formas: cartão de crédito, paypal, bitcoin ou mecanismo próprio da plataforma (ex.: Google SuperChat).

No caso da doação em bitcoin deve ser armazenada também a TxID (um código *hash*), além do valor associado à doação. Para doações via PayPal, deve-se armazenar o IDPayPal de pagamento, além do valor associado à doação. No caso de cartão de crédito, deve ser armazenado o número do cartão, a bandeira do cartão e a data e hora. Nos casos de mecanismo próprio da plataforma, deve-se ser associado um número sequencial que depende do usuário, do comentário e da plataforma, além do valor da doação. Para cada doação é necessário armazenar o *status* que só pode ser um dos três {recusado, recebido ou lido}.

Modelo ER



Modelo Relacional

Empresa(nro, nome, nome_fantasia)

-- Quantidade de usuários qtd_users é atributo derivado e requer atualização periódica

Plataforma(nro, nome, qtd_users, empresa_fund, empresa_respo, data_fund)

empresa_fund referencia Empresa(nro)

empresa_respo referencia Empresa(nro)

-- fator_conver é o fator de conversão para dólar

Conversao(moeda, nome, fator_conver)

Pais(DDI, nome, moeda)

moeda referencia Conversao(moeda)

Usuario(nick, email, data_nasc, telefone, end_postal, pais_residencia)

pais_residencia referencia Pais(ddi)

PlataformaUsuario(nro_plataforma, nick_usuario, nro_usuario)

nro_plataforma referencia Plataforma(nro)

nick_usuario referencia Usuario(nick)

StreamerPais(nick_streamer, ddi_pais, nro_passaporte)

nick_streamer referencia Usuario(nick)

ddi_pais referencia Pais(ddi)

EmpresaPais(nro_empresa, ddi_pais, id_nacional)

-- Quantidade de visualizações qtd_visualizações é atributo derivado e requer atualização

Canal(nome, tipo, data, desc, qtd_visualizacoes, nick_streamer, nro_plataforma)

nro_plataforma referencia Plataforma(nro)

nick_streamer referencia Usuario(nick)

Patrocionio(nro_empresa, nome_canal, nro_plataforma, valor)

nro_empresa referencia Empresa(nro)

nome_canal, nro_plataforma referencia Canal(nome, nro_plataforma)

NivelCanal(nome_canal, nro_plataforma, nivel, valor, gif)

nome_canal, nro_plataforma referencia Canal(nome, nro_plataforma)

Inscrição(nome_canal, nro_plataforma, nick_membro, nivel)

nick_membro referencia Usuario(nick)

(nome_canal, nro_plataforma, nivel)

referencia NivelCanal(nome_canal, nro_plataforma, nivel)

-- Verificar a possibilidade do uso de identificador artificial (toda ou parte da chave)

Video(nome_canal, nro_plataforma, titulo, dataH, tema, duracao, visu_simul, visu_total)

(nome_canal, nro_plataforma) referencia Canal(nome, nro_plataforma)

Participa(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_streamer)

(nome_canal, nro_plataforma, titulo_video, dataH_video) referencia Video(nome_canal, nro_plataforma, titulo_video, dataH_video)

nick_streamer referencia Usuario(nick)

Comentario(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq, texto, dataH, coment_on)

(nome_canal, nro_plataforma, titulo_video, dataH_video) referencia Video(nome_canal, nro_plataforma, titulo, dataH)

nick_usuario referencia Usuario(nick)

Doacao(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_pg, valor, status)

(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario) referencia

Comentario(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq)

BitCoin(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao, TxID)

(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao)

referencia

Doacao(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_pg)

PayPal(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao, IdPayPal)

(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao)

referencia

Doacao(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_pg)

CartaoCredito(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao, nro, bandeira)

(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao)

referencia Doacao(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_pg)

MecanismoPlat(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao, seq_plataforma)

(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_doacao)

referencia

Doacao(nome_canal, nro_plataforma, titulo_video, dataH_video, nick_usuario, seq_comentario, seq_pg)

Itens Mandatórios

1. Inserir entre 100 e 1000 tuplas por tabela (dados artificiais serão aceitos).
2. Construção criteriosa de identificadores artificiais e *constraints* UNIQUE e NOT NULL.
3. Construção de visões para otimizar o desempenho de buscas. Até 05 visões. Justifique suas escolhas (cada visão).
4. Construção de índices de apoio. Definir e criar índices de apoios. Considerar o *overhead* de inserção. Até 05 índices. Justificar suas escolhas (cada índice).
5. Definir triggers para manter a consistência do BD (todas as necessárias via requisitos).
6. Usar functions (ou stored procedures) para responder todas as consultas abaixo. As functions devem ser parametrizadas.

Consultas a serem respondidas (na forma de functions/procedures)

1. Identificar quais são os canais patrocinados e os valores de patrocínio pagos por empresa. Dar a opção de filtrar os resultados por empresa como um parâmetro opcional na forma de uma stored procedure.
2. Descobrir de quantos canais cada usuário é membro e a soma do valor desembolsado por usuário por mês. Dar a opção de filtrar os resultados por usuário como um parâmetro opcional na forma de uma stored procedure.
3. Listar e ordenar os canais que já receberam doações e a soma dos valores recebidos em doação. Dar a opção de filtrar os resultados por canal como um parâmetro opcional na forma de uma stored procedure.
4. Listar a soma das doações geradas pelos comentários que foram lidos por vídeo. Dar a opção de filtrar os resultados por vídeo como um parâmetro opcional na forma de uma stored procedure.
5. Listar e ordenar os k canais que mais recebem patrocínio e os valores recebidos.
6. Listar e ordenar os k canais que mais recebem aportes de membros e os valores recebidos.
7. Listar e ordenar os k canais que mais receberam doações considerando todos os vídeos.
8. Listar os k canais que mais faturam considerando as três fontes de receita: patrocínio, membros inscritos e doações.