Insper

Projeto 1: Construindo uma rede social

Introdução

O nosso primeiro projeto será a construção da base de dados para uma rede social para observadores de pássaros. O projeto será desenvolvido em duas etapas:

1. Implantação da base inicial e população com exemplos fictícios de dados.
2. Incorporação de novos requisitos.

O projeto será feito em duplas.

Você deverá submeter todo o trabalho via Github, envie o link pelo Blackboard.

Prazo:

- Fase 1: domingo, 28/09, 23:59.
- Fase 2: domingo, 13/10, 23:59.

Fase 1

Conversando com o cliente vocês levantaram os seguintes requisitos iniciais:

- Dados pessoais do usuário: nome, email, cidade onde mora.
- Um usuário pode também declarar várias preferências de pássaros.
- Cada post nessa rede social tem um título (obrigatorio), um texto (opcional), e uma URL de uma foto (opcional) – vamos supor que as fotos são armazenadas por algum outro sistema, que disponibiliza as URLs.
- O usuário pode apagar seus posts, mas eles não devem ser removidos fisicamente da base, apenas marcados como inativos (delete lógico).
- Os textos podem ter tags # marcando o tipo de pássaro de que se fala no post - vou querer buscar por tipo de pássaro depois.
- O texto também pode ter tags @ referenciando outros usuários (@shouts), e isso vai ser importante destacar também.

Insper

- Ao guardar informações de quem viu o post, queremos também guardar o tipo de aparelho (Android ou iOS), browser (Chrome, IE, Firefox, Safari, outros), IP, e instante da visualização.

Exemplo de post:

Titulo: Que tipo de canário é esse? Foto: (URL de uma foto de um pombo) Texto: @Juquinha @Mariazinha Gente, hoje esse canário estranho fez caca no meu carro, é impressionante! Como pode ser tão grande? #canário.

Entregáveis:

- Modelo Entidade-Relacionamento representando a informação obtida do cliente.
- Modelo Relacional
    - Schema
    - Dicionário de dados:
        - Para cada tabela:
            - O que representa?
            - Para cada campo:
                - Nome do campo
                - Descrição
                - Auto-gerada? Como? (timestamp, auto-increment, etc)
                - Chave primária?
                - Chave estrangeira? Para quem?
                - Restrições?

    - Diagrama do modelo relacional, indicando cardinalidades (“bolinha”, “risquinho”, “pé-de-galinha”) e relacionamentos identificadores/não-identificadores.
- Scripts de criação do banco de dados
- Script de criação de triggers e outros constraints para garantir a integridade do banco de dados. Por exemplo: se um usuário apaga um comentário que referenciava outros usuários (@Pedrinho, @Laurinha), devemos apagar (logicamente, não fisicamente) essa informação de referência.
- Programa em Python que testa a funcionalidade do banco de dados (ver seção abaixo)

Testando o banco de dados

Para testar o banco de dados, vamos supor que o servidor MySQL ao qual você tem acesso é um servidor de testes, e não o servidor de produção! Seu script Python de teste deverá então:

1. Rodar todos os scripts de modificação do banco de dados na sua sequência original, sem falhas.
2. Rodar testes para cobrir todas as funcionalidades descritas no enunciado do problema.

Você deve desenvolver seu programa de testes usando o framework de testes unitários padrão do Python, o `unittest`. Para rodar os scripts de criação e modificação do banco de dados, use o módulo `subprocess`.

Insper

Rubrica de avaliação da Fase 1:

Conceito	Descrição
I	Não entregou mais de um dos itens requeridos
D	Não entregou um dos itens requeridos.
C	Entregou todos os componentes requeridos. Os testes unitários rodam corretamente os scripts de criação da base de dados, e cobrem algumas das funcionalidades principais. Podem existir falhas na garantia de integridade (e.g. falta de triggers adequados). Erros menores de modelagem
B	Rubrica C, mas com cobertura completa de testes para as funcionalidades do projeto. Triggers e constraints garantem a integridade dos dados. Qualidade da entrega está aceitável, mas poderia melhorar (redação, qualidade do código). Sem erros de modelagem.
A	Rubrica B, qualidade excepcional.

Fase 2

Nesta fase o cliente percebeu que tem novos requisitos:

- Joinhas: Um usuário pode registrar um joinha ou um anti-joinha em qualquer post. O sistema não deverá permitir que o mesmo usuário dê vários joinhas ou anti-joinhas no mesmo post – deve ser possível apenas cancelar ou mudar de ideia.
- O sistema deve permitir a adição de posts, obviamente. Note que ao adicionar um post vocês devem também registrar os #tags e @shouts.
- O sistema deve permitir que um post seja removido (note que na fase 1 vocês fizeram o design que permitia que isso fosse feito, agora na fase 2 vocês estão fazendo as queries que executam essas operações).
- O sistema deve permitir várias consultas
    - Posts do usuário em ordem cronológica reversa.
    - Usuários mais popular de cada cidade.
    - Lista de usuários que referenciam um dado usuário.
    - Tabela cruzada de quantidade de aparelhos por tipo e por browser.

    - Lista com URLs de imagens e respectivos #tags de tipo de pássaro.
- Crie uma feature do seu interesse.
- Todas essas operações de interação com o banco de dados devem ser feitas através de um serviço REST escrito em Python. Use FastAPI (https://fastapi.tiangolo.com/), teste com sua ferramenta favorita de interação com serviços web.
- Obviamente a documentação do modelo (modelagem E-R, modelagem relacional, dicionário de dados) deverá ser atualizada
- Todas as alterações de schema devem ser feitas com scripts *delta*, sem alterar os scripts anteriores
- Testes unitários deverão ser feitos para os novos itens.

Rubrica de avaliação da Fase 2:

Conceito	Descrição
I	Não entregou mais de um dos itens requeridos
D	Não entregou um dos itens requeridos. Itens com bugs graves não contam como itens entregues.
C	Entregou todos os componentes requeridos, alguns problemas de qualidade ou algum pequeno bug.
B	Rubrica C, boa qualidade, sem bugs.
A	Rubrica B, usou de modo apropriado features mais avançadas do SQL como stored procedures, views, índices.

Conceito final

A nota de cada fase será convertida para um valor numérico segundo a regra:

I - zero	D – 3 pts	C – 5 pts	B – 8 pts	A – 10 pts

A nota final será dada pela média das notas de fase.