INSTITUTO UNIVERSITÁRIO DE LISBOA

LEI, LETI e LIGE
BASES DE DADOS

"MUSISYS"

Ano Letivo 2023-2024

Projeto – Parte 2

A partir da Base de Dados criada para dar suporte ao Sistema de Informação para gerir festivais musicais, a segunda parte do projeto pede que sejam desenvolvidas: otimizações (1) e, sobre o modelo relacional otimizado, a criação de automatismos (2), um conjunto de pesquisas (3) e um protótipo web-based (HTML/PHP) para demonstração do sistema (4).

Para esta parte do projeto deverá utilizar a base de dados disponibilizada pela equipa docente. O objetivo consiste em ter uma solução comum que permita a mesma base de trabalho para todos, e assegurar a avaliação desta parte do projeto sem interferência de decisões anteriores.

1 OTIMIZAÇÃO DA BD

Otimize o esquema relacional fornecido de modo a obter um esquema mais eficiente tendo em conta as linhas de orientação mencionadas nas aulas teóricas. Recordando resumidamente:

- Fundir tabelas relacionadas a um-para-um – aquelas que no diagrama de classes UML são ligadas por associações 1-para-1 e por generalizações – para eliminar a necessidade de realizar *joins* ao consultar os dados;
- Introduzir chaves primárias numéricas nas tabelas que considera terem chaves primárias ineficientes. Nas chaves que introduzir, considere a conveniência de as especificar como *auto-increment*.
- Introduzir colunas com valores pré-calculados para evitar que sejam calculados repetidamente a cada select. Por exemplo, num outro caso, pretende-se manter o custo total de uma encomenda sempre atualizado e disponível na respetiva tabela, para que não tenha de ser calculado de cada vez que a encomenda é acedida.

No relatório, Para cada otimização realizada indique:

(i) o que foi melhorado e como foi melhorado;

(ii) a justificação das alterações realizadas;

(iii) as limitações em termos de validação dos dados que surgiram quando alterou a estrutura das tabelas;

(iv) os mecanismos que criou para contornar as limitações anteriores.

Nos últimos dois pontos, indique "Nenhuma" se for o caso.

2 AUTOMATISMOS: TRIGGERS, STORED PROCEDURES E FUNÇÕES

2.1 TRIGGERS

T 1. Confirmar que os roadies apenas montam o palco onde o seu artista atua. Cancelar se assim não for.

T 2. Pretende-se registar a qtd_espetadores em cada dia de festival. Introduza essa coluna e faça com que:

a) Seja automaticamente atualizada com a criação de bilhetes e a sua devolução. Os jornalistas não contam;

b) A base de dados não permita inserir mais bilhetes se a lotação diária do recinto – na coluna lotação da edição – já foi excedida.

2.2 STORED PROCEDURES

P 1. Clonar uma edição. O procedimento recebe a identificação da edição a clonar e a data de início para o clone. Pretende-se que clone os palcos e os dias do festival. Para estes últimos, as suas datas deverão estar igualmente separadas tal como na edição original.

P 2. Criar uma edição e os palcos.

2.3 FUNÇÕES

F 1. Crie uma função para calcular a média referente ao lucro por edição (por ano): Escreva uma função chamada CalcularMedia que calcula a média referente ao lucro por edição.

F 2. Crie uma função para calcular o número de participantes da última edição do festival.

3 PESQUISA DE DADOS: SQL QUERIES E VIEWS

Elabore comandos select necessários para responder aos seguintes pedidos de informação, criando views onde indicado. Atribua a cada, um nome com o número da pergunta (Q1, Q2…).

Q 1. Produza o cartaz de uma determinada edição do festival. O cartaz deve ser uma lista dos participantes, mostrando o nome dos artistas e o dia em que atuam, sendo essa lista ordenada pelo dia ascendentemente e, dentro de cada dia, pelo cachet de forma descendente. Se preferir, coloque a sua instrução dentro de uma stored procedure chamada Cartaz.

Q 2. Elabore uma view chamada Resultados_diarios que mostra a quantidade de espetadores e faturação em cada dia de festival; a faturação é a soma do valor dos bilhetes, excluindo os que foram devolvidos.

Q 3. Qual a quantidade de espetadores em cada dia numa determinada edição.

Se preferir coloque a sua instrução dentro de uma stored function chamada Qtd_espetadores_no_dia; esta possuirá um parâmetro que fornece a data do dia à sua instrução.

Q 4. Elabore uma view chamada Estilos_musicais_por_edicao que mostra a quantidade de artistas por cada estilo musical em cada edição do festival:

Edição	Estilo	Qtd_artistas
1	Rock	15
1	Hip-hop	2
1	Pop	3
2	Rock	18
2	Heavy metal	2
…	…	…

Q 5. Elabore uma view chamada Todos_os_participantes que lista de todos os artistas participantes registados, para cada um mostrando o nome, há quantos anos foi a sua última atuação no festival e qual foi o seu cachet nessa ocasião.

Q 6. Lista dos artistas participantes numa edição com entrevista realizada por um/a determinado/a jornalista.

Se preferir coloque a sua instrução dentro de uma stored procedure chamada Entrevistados_por, através de cujos parâmetros passamos o número da edição e o nome do/a jornalista.

Q 7. Produza a lista dos artistas ainda sem entrevista na edição mais recente por um determinado jornalista.

Se preferir coloque a sua instrução dentro de uma stored procedure chamada Ainda_nao_entrevistados_por, através de cujo parâmetro passamos o nome do/a jornalista.

4 PROTÓTIPO DO SISTEMA *WEB-BASED* (HTML/PHP +SQL)

Crie um protótipo que permita:

W 1. Fazer o registo de uma edição do festival.

W 2. Fazer pesquisas sobre todos os artistas participantes, nas várias edições:

a. Pesquisa básica - selecionar o artista com base no seu código de participante.

b. Pesquisa avançada - selecionar o artista com base nos seguintes critérios:

i. Palco em que já atuou;

ii. Número mínimo de entrevistas dadas.

W 3. Listar todas as próximas edições:

c. A partir de cada uma das futuras edições, listar os artistas e o palco em que vão atuar.

d. Remover um artista (o artista cancelou o show).

e. Mudar um artista de um palco para outro.

Bom trabalho!