



Especificação de Requisitos de Dados.

Um colecionador de *música clássica* resolveu utilizar a tecnologia de banco de dados para implementar uma versão personalizada do Spotify, o **SpotPer**. Para tanto, resolveu contratá-los para realizar o projeto e a implementação do banco de dados do SpotPer, o **BDSpotPer**. O SGBD a ser utilizado pelo SpotPer pode ser SQL Server, Oracle ou Postgres. Após a análise de requisitos, obtida através de entrevistas com o usuário, você identificou as seguintes características:

- (i) Cada **álbum**, uma coleção de músicas agrupadas em um meio físico de armazenamento, possui:
 - a. um código identificador uma descrição, gravadora, preço de compra, data de compra, data de gravação e o tipo de compra.
 - b. A data de gravação deve ser obrigatoriamente posterior a 01.01.2000.
 - c. O meio físico do álbum, que pode ser CD, vinil ou *download*.
 - i. Quando o meio físico for CD ou vinil, o álbum pode ser composto por um ou mais CDs ou vinis.
 - d. O preço de compra.
- (ii) Cada CD, vinil ou *download* possui ainda um conjunto de faixas (músicas).
- (iii) Cada **faixa** de um álbum possui obrigatoriamente como propriedades
 - a. o número da faixa (posição da faixa no álbum), uma descrição, tipo de composição, intérprete(s), compositor(es), tempo de execução e tipo de gravação.
 - b. Quando o meio físico de armazenamento é CD, o tipo de gravação tem que ser ADD ou DDD. Quando o meio físico de armazenamento é vinil ou download, o tipo de gravação não terá valor algum.
 - c. Uma faixa pode estar associada a vários compositores e intérpretes.
- (iv) Para cada tipo de **composição**, devem estar associados um código identificador e a descrição. O tipo deve caracterizar se a obra gravada é uma sinfonia, ópera, sonata, concerto e assim por diante. É obrigatório identificar o tipo de composição para cada faixa existente. Uma faixa só pode apresentar um tipo de composição.
- (v) Cada **intérprete** possui um código identificador, nome, tipo. Tipo de intérprete pode ser orquestra, trio, quarteto, ensemble, soprano, tenor, etc...
- (vi) Um **compositor** deve possuir, como propriedades, nome, local de nascimento (cidade e país), data de nascimento e data de morte (se for o caso). Cada compositor possui um identificador. Podem existir compositores no banco de dados, sem estarem associados a faixas. Cada compositor deve estar obrigatoriamente associado a um período musical.
- (vii) Cada **período musical** possuirá um código, uma descrição (idade média, renascença, barroco, clássico, romântico e moderno) e intervalo de tempo em que esteve ativo.
- (viii) Para cada **gravadora**, estão associados um código, nome, endereço, telefones e endereço da *home page*.
- (ix) O usuário do *SpotPer* pode definir **Playlists**. Uma *playlist* pode ser composta por uma ou mais faixas, que, por sua vez, podem pertencer a álbuns distintos. Uma *playlist* terá como propriedades:
 - a. Código identificador, nome, data de criação, tempo total de execução.



- b. Para cada faixa de uma *playlist*, tem-se a data da última vez que foi tocada e o número de vezes que foi tocada.

Parte I

- 1) Utilize o DER para modelar os dados do SpoPer, considerando as especificações apresentadas acima.
- 2) Construa o diagrama relacional correspondente ao DER da questão 1.

Parte II

- 1) Crie o banco de dados *BDSpotPer*, considerando o seguinte: o banco de dados deve possuir três *filegroups* (*tablespaces*) e o arquivo de *log*. O *filegroup* primário deve conter apenas o arquivo primário do banco de dados. Um segundo *filegroup* deve conter dois arquivos e um terceiro deve conter apenas um arquivo.
- 2) As tabelas referentes aos conjuntos de *playlists*, faixas e de relacionamento entre as duas devem ser alocadas no *filegroup* (*tablespace*), definido com apenas um arquivo. As outras tabelas devem ser alocadas no *filegroup* com dois arquivos.
- 3) Defina as seguintes restrições
 - a) Um álbum, com faixas de músicas do período barroco, só pode ser inserido no banco de dados, caso o tipo de gravação seja DDD.
 - b) Um álbum não pode ter mais que 64 faixas (músicas).
 - c) No caso de remoção de um álbum do banco de dados, todas as suas faixas devem ser removidas. Lembre-se que faixas podem apresentar, por sua vez, outros relacionamentos.
 - d) O preço de compra de um álbum não dever ser superior a três vezes a média do preço de compra de álbuns, com todas as faixas com tipo de gravação DDD.
- 4) Defina um índice primário para a tabela de Faixas sobre o atributo código do álbum. Defina um índice secundário para a mesma tabela sobre o atributo tipo de composição. Os dois com taxas de preenchimento máxima.
- 5) Criar uma visão materializada que tem como atributos o nome da *playlist* e a quantidade de álbuns que a compõem.
- 6) Defina uma função que tem como parâmetro de entrada o nome (ou parte do) nome do compositor e o parâmetro de saída todos os álbuns com obras compostas pelo compositor.
- 7) Implemente um aplicativo Java, C ou Python, que implementa as seguintes funcionalidades:
 - (i) Criação de *playlists* no banco de dados. Esta função deve mostrar todos os álbuns existentes. O usuário pode, assim, escolher o(s) álbum(ns) e quais faixas destes que devem compor a *playlist*.



- (ii) Manutenção de *playlists*. Esta funcionalidade deve mostrar todas as *playlists* existentes. Ao escolha uma *playlist*, a função deve permitir a remoção de músicas existentes e a inserção de novas músicas na *playlist* escolhida
- (iii) Apresente o resultado das seguintes consultas sobre o banco de dados:
 - a. Listar os álbuns com preço de compra maior que a média de preços de compra de todos os álbuns.
 - b. Listar nome da gravadora com maior número de *playlists* que possuem pelo uma faixa composta pelo compositor Dvorack.
 - c. Listar nome do compositor com maior número de faixas nas *playlists* existentes.
 - d. Listar *playlists*, cujas faixas (todas) têm tipo de composição “Concerto” e período “Barroco”.