e-Fólio B

U.C. 21103

Sistemas de Gestão de Bases de Dados

2017-2018

Resolução e Critérios de Correção

INSTRUÇÕES

1) O e-fólio é constituído por 5 perguntas. A cotação global é de 5 valores.

2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 10% a 100%.

3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.

4) O nome do ficheiro deve seguir a normal "eFolioB" + <nº estudante> + <nome estudante com o máximo de 3 palavras>. Penalização de 10% a 100%.

5) Na primeira página do e-fólio deve constar o nome completo do estudante bem como o seu número. Penalização de 10% a 100%.

6) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.

7) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.

8**)** A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

9) Critérios de correção gerais: todas as respostas devem ser justificadas, incluir imagens e exemplos com vista a clarificar os argumentos expostos.

Vetor Cotações
1 2 3, 4 5 pergunta
10 10 10, 10 10 décimas

1) (1 valor) Prática em MySQL de planos de execução de consultas.

1.a) Instale no seu computador o SGBD MySQL.

1.b) Utilize a base de dados existente *World* que contem as tabelas *City* (4079 registos), *Country* (com 239 registos) e *Countrylanguage* (com 984 registos):

- City (id -> name, countrycode, district, population, …)
- Country (code -> name, continent, region, surfaceArea, indepYear, …)
- Countrylanguage (countrycode, language -> isOfficial, percetange, …)

1.c) Escreva uma consulta que devolve todas as cidades, e respetivo país, em que se fala a língua portuguesa.

Resposta:

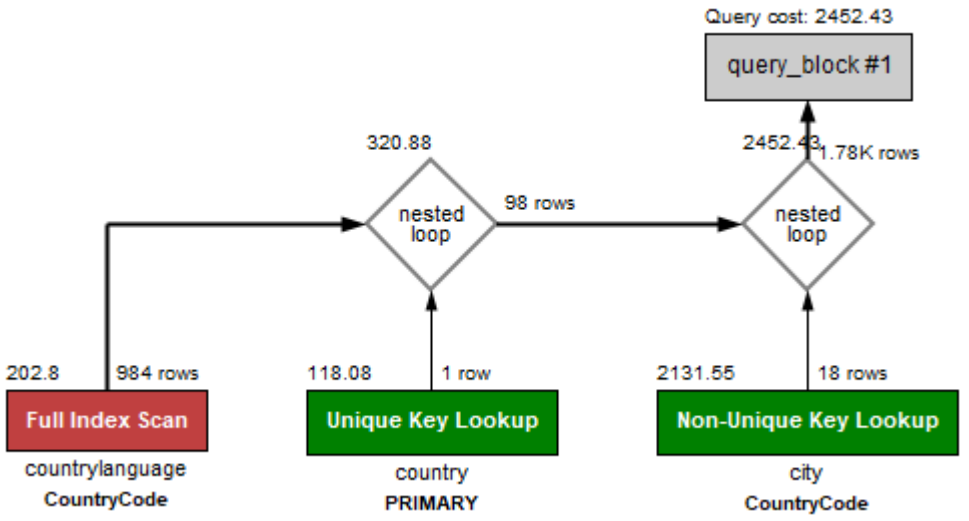
```
-- EXPLAIN
SELECT city.name, country.name
FROM city
INNER JOIN country ON city.CountryCode = country.Code
INNER JOIN countrylanguage ON countrylanguage.CountryCode = country.Code
WHERE countrylanguage.Language = "Portuguese"
```

1.d) Utilize o comando EXPLAIN, analise o resultado e desenhe a árvore do plano de execução da consulta.

Resposta:

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	countrylanguage	NULL	index	PRIMARY,CountryCode	CountryCode
1	SIMPLE	country	NULL	eq_ref	PRIMARY	PRIMARY
1	SIMPLE	city	NULL	ref	CountryCode	CountryCode

key_len	ref	rows	filtered	Extra
3	NULL	984	10.00	Using where; Using index
3	world.countrylanguage.CountryCode	1	100.00	NULL
3	world.countrylanguage.CountryCode	18	100.00	NULL



- o número de linhas a serem examinadas é 984 linhas da tabela *countrylanguage*,

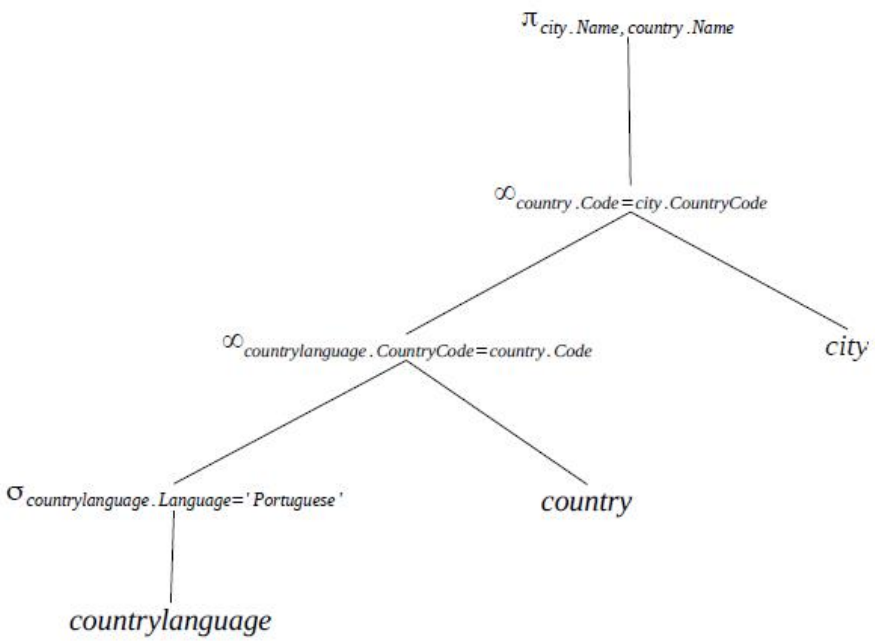
- estimativa de 1 linha (‘rows examined per scan’) da tabela *country* por cada uma das linhas da tabela anterior,

- no 1º ‘nested-loop’ a percentagem de linhas filtradas com o ‘where’ será de aproximadamente 10%, resultando, em cerca de 98 linhas na junção de *countrylanguage* com *country*,

- estimativa de 18 linhas (‘rows examined per scan’) da tabela *city* por cada uma das linhas resultantes da junção das tabelas countrylanguage e country, Ncity/Ncountry = 4079/239 = 18,

- no 2º ‘nested-loop’ o valor de 1764 resulta do produto de 18 x 98 linhas.

O plano de execução será com estrutura ‘left-deep-join-tree’ :



Critérios de correção:
- 1.c) 2 décimas, consulta SQL
- 1.d) 3 décimas, output comando Explain
- 1.d) 3 décimas, texto do estudante sobre a análise do resultado
- 1.d) 2 décimas, plano de execução
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

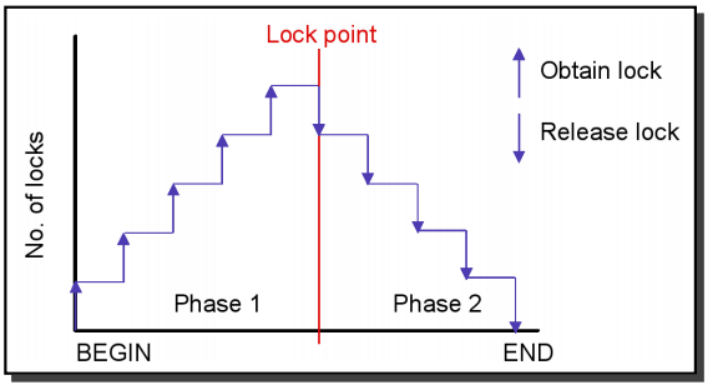
2) (1 valor) Capítulo 15, Concurrency Control

2.a) Defina o protocolo 2-PL. Quais as vantagens e desvantagens?

Resposta:
O protocolo 2PL (2-phase locking) é um algoritmo de bloqueio utilizado para o controlo de concorrência entre transações. Os algoritmos de bloqueio são os mais utilizados nos SGBD e, entre eles, o 2PL é o mais aplicado.

O protocolo 2PL utiliza dois tipos de bloqueios (locks):
• Bloqueio partilhado (S-lock) (utilizado nas operações de leitura (Read)): o item de dados pode ser partilhado por várias transações.
• Bloqueio exclusivo (X-lock) (utilizado nas operações de escrita (Write)): o item de dados não pode ser partilhado por várias transações.

O protocolo 2PL exige que todas as transações solicitem todos os bloqueios que necessitem antes de libertar qualquer um dos bloqueios que detenha. Desta forma a gestão de bloqueios é realizada em duas fases:
• Fase de crescimento ou expansão: a transação apenas pode adquirir bloqueios.
• Fase de encolhimento ou contenção: a transação apenas pode libertar bloqueios.
O ponto de mudança de fases é designado de ponto de bloqueio (lock point).



• Vantagens:
- produz escalonamentos serializáveis
- de fácil implementação

• Desvantagens:
- os pedidos de ‘locks’ são problemáticos para o ‘lock manager’
- limita a concorrência, colocando em transações em fila de espera
- não evita a ocorrência de ‘deadlock’
- não evita recuperações em cascata (‘cascading rollback’)

2.b) Considere o protocolo 2-PL e explique detalhadamente a execução das seguintes transações: W1(x), R2(y), R1(x), Commit_1, R2(x), W2(y), Commit_2

Resposta:

T1	**T2**	**Gestão de 'Locks'**
X-Lock(X)		Exclusive lock(T1, X) permite o acesso exclusivo da transação T1 ao item X
W(X)		
	S-Lock(Y)	Shared lock (T2, Y) permite o acesso partilhado da transação T2 ao item Y
	R(Y)	
R(X)		
UNLOCK(X)		Liberta lock(x)
Commit		
	S-Lock(X)	Shared lock (T2, X) permite o acesso partilhado da transação T2 ao item X
	R(X)	
	X-Lock(Y)	Exlusive lock(T2, Y) permite o acesso exclusivo da transação T2 ao item Y, fase de crescimento
	W(Y)	
	UNLOCK(X)	Liberta lock(X), fase de decrescimento
	UNLOCK(Y)	Liberta lock(Y)
	Commit	

Critérios de correção:
- 2.a) 3 décimas, definir 2-PL
- 2.a) 2 décimas, vantagens e desvantagens
- 2.b) 3 décimas, tabela com transações e locks
- 2.b) 2 décimas, considerar as fases crescimento e decrescimento
- erros, omissões ou redundâncias: -20% a -100%

3) (1 valor) Capítulo 16, Recovery System

3.a) Defina o algoritmo ARIES. Quais as principais fases?

ARIES "Algorithms for Recovery and Isolation Exploiting Semantics" é um algoritmo para recuperação de base de dados, desenvolvido para trabalhar em três fases.

As fases do protocolo ARIES são: análise, refazer (redo) e desfazer (undo).

A fase de análise destina-se a recolher informação sobre páginas sujas e transações perdedoras (isto é, não confirmadas à altura da falha). Começa no ponto de controlo mais recente e analisa as entradas de *log* que ocorreram depois até à falha.

A fase de refazer (redo) percorre o *log* até à entrada mais recente, refazendo modificações desde a primeira entrada do *log*, cuja modificação poderá ter sido perdida devido à falha. Esta entrada é a correspondente ao redoLSN encontrado na fase anterior. O objetivo desta fase é "repetir a história" para garantir que as operações no *log* são aplicadas antes da fase seguinte.

A fase de desfazer (undo) percorre o *log* da entrada mais recente para a mais antiga, removendo da base de dados todos os efeitos das transações perdedoras.

O algoritmo ARIES utiliza:
1. Um log com "Log Sequence Number" (LSN): Este log faz parte do SGBD e é aqui usado para guardar quais operações foram aplicadas na base de dados.
2. Uma "Transaction Table" (TT): lista cada transação ativa, com o seu respetivo ID, o último LSN no registo de log dessa transação e o status da mesma.
3. Uma "Dirty Page Table" (DPT): lista cada página suja no 'buffer' e o número LSN da primeira atualização desta página.

3.b) Aplique o algoritmo ao seguinte log:
10 begin chk
20 end chk
30 T1 start
40 T1, P5, update
50 T2 start
60 T2, P3, update
70 T1 abort
80 clr undo LSN 40
90 T1 end
100 T3 start
110 T3, P1, update
120 T2, P5, update
FAIL

Resposta:

Fase de análise: começa no LSN 10 (checkpoint)

Dirty Page Table

Page	First-LSN
P1	110
P3	60
P5	40

Transaction Table (T1 é removida da tabela)

Transaction	Last-LSN	status
T1	90	abort, end
T2	120	live
T3	110	live

Fase redo: começa no LSN 40 (menor first LSN da DPT).
A fase de refazer começa no LSN mais antigo na DPT, ou seja 40, e percorre o *log* até à entrada mais recente. Para cada entrada (CLR ou UPDATE):

LSN 40	Redo Update P5
LSN 50	No action
LSN 60	Redo Update P3
LSN 70	No action
LSN 80	Redo CLR Undo LSN 40, P5
LSN 90	No action
LSN 100	No action
LSN 110	Redo Update P1
LSN 120	Redo Update P5

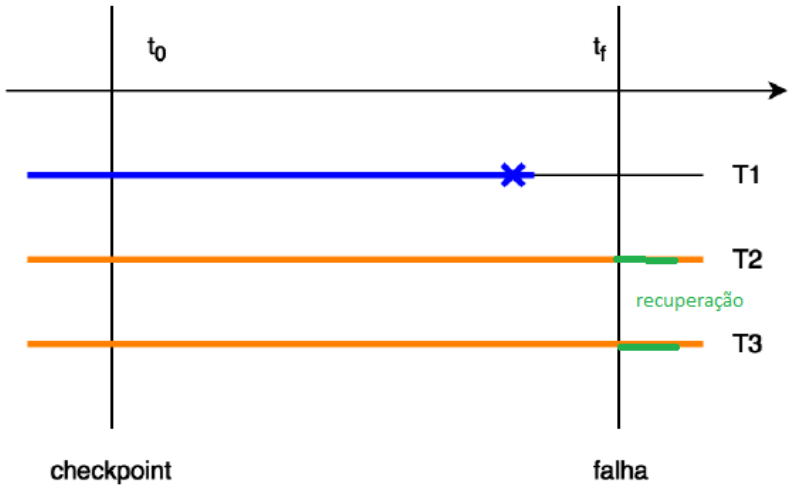
Fase undo: começa no LSN 120 (maior LSN da TT).
A fase de desfazer tem de remover os efeitos das transações 2 e 3. Coloca o undoNxtLSN de cada transação perdedora a Last-LSN, escolhe o maior dos undoNxtLSN, 120, e começa nessa entrada percorrendo o log até à entrada mais antiga. Esta fase acrescenta as seguintes entradas no log:

LSN	Ti	Operação	Página	Undo-Nxt-LSN
130	2	CLR UNDO 120	5	60
140	2	CLR UNDO 60	3	50
150	2	CLR UNDO 50		
160	3	CLR UNDO 110	1	100
170	3	CLR UNDO 100		

Esta frase acrescenta as seguintes entradas à tabela de log:

LSN	**LOG**	**Prev LSN**
LSN 10	begin chk	-
LSN 20	end chk	-
LSN 30	T1 start	-
LSN 40	T1, P5, update	30
LSN 50	T2 start	-
LSN 60	T2, P3, update	50
LSN 70	T1 abort	40
LSN 80	clr undo LSN 40	-
LSN 90	T1 end	80
LSN 100	T3 start	-
LSN 110	T3, P1, update	100
LSN 120	T2, P5, update	60
	Falha	-
LSN 130	CLR T2 LSN 120, P5	120
LSN 140	CLR T2 LSN 60, P3	60
LSN 150	T2 end	50
LSN 160	CLR T3 LSN 110, P1	110
LSN 170	T3 end	100

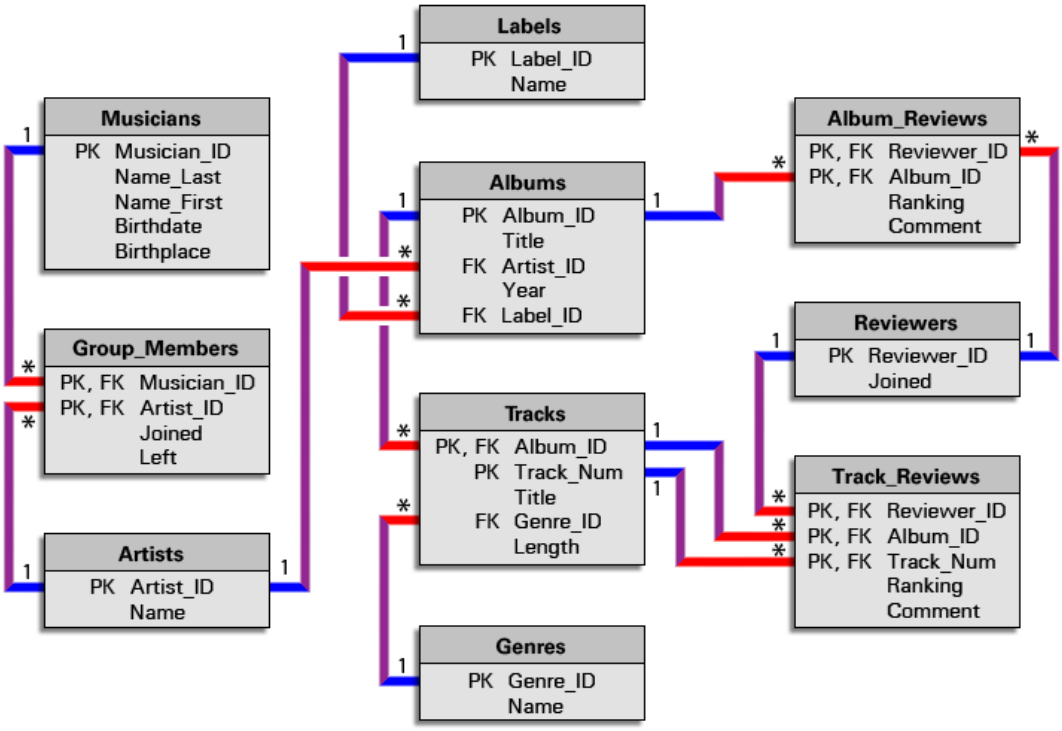
Em suma, para cada transação:
T1 fez abort (redo)
T2 fez rollback (redo, undo)
T3 fez rollback (redo, undo)



Critérios de correção:
- 3.a) 2 décimas ARIES e fases
- 3.b) 3 décimas, fase análise
- 3.b) 3 décimas, fase redo
- 3.b) 2 décimas, fase undo
- erros, omissões ou redundâncias: -20% a -100%

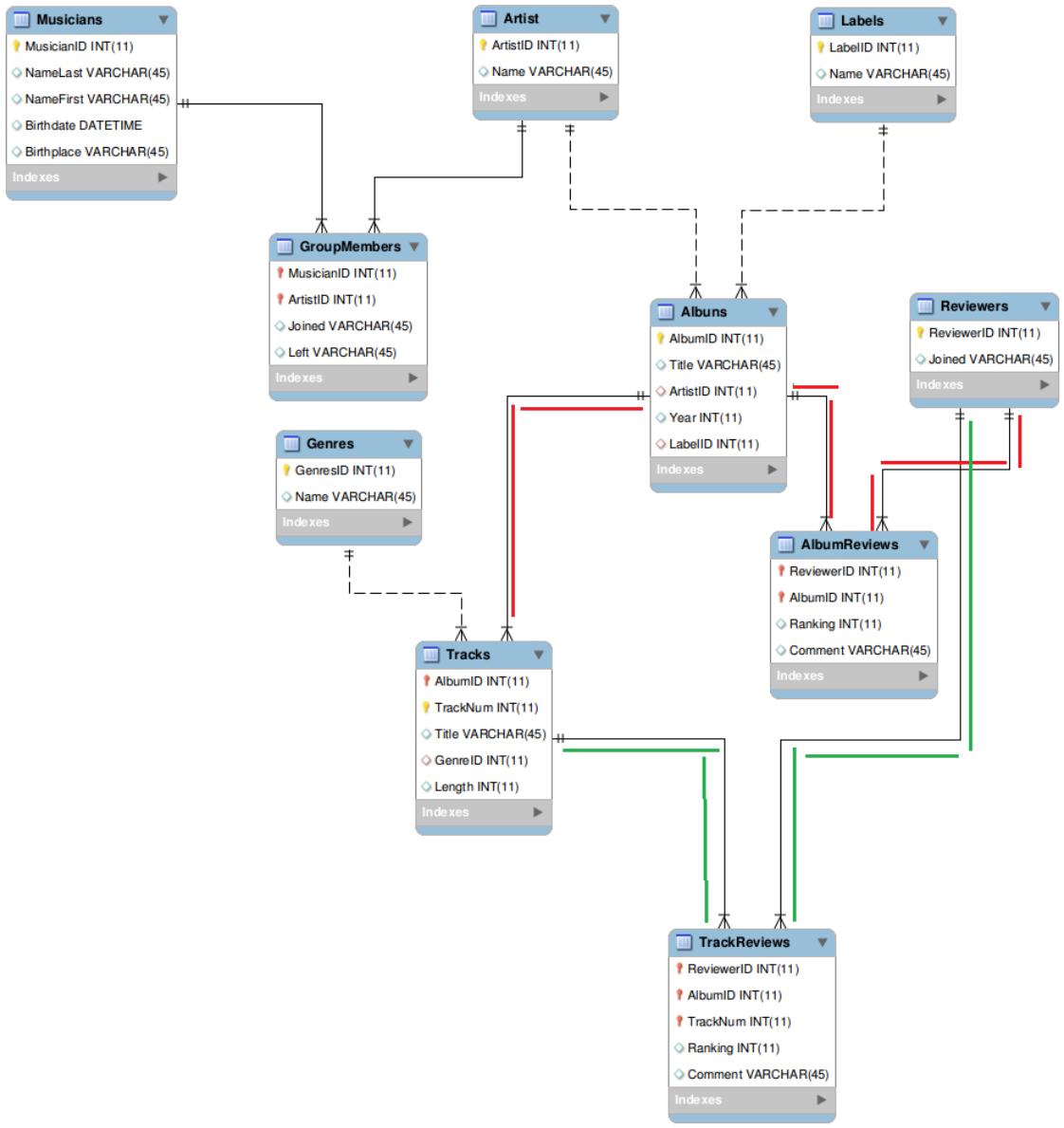
4) (1 valor) No processo de extração de dados de uma base de dados transacional existem 2 tipos de armadilhas no SQL (SQL traps):
- junções com múltiplos caminhos (“multiple access path problem”)
- junções com agregações de dados de 2 tabelas (“connection trap”)

Considere a seguinte base de dados de músicas:



4.a) Para a base de dados da figura exemplifique uma consulta que evidencie a armadilha de junções com múltiplos caminhos, com dados e resultados.

Resposta:



Obtemos resultados diferentes para as seguintes consultas:

```
SELECT T.AlbumID, T.TrackNum, R.ReviewerID
FROM Tracks T
INNER JOIN Albuns A ON T.AlbumID = A.AlbumID
INNER JOIN AlbumReviews AR ON A.AlbumID = AR.AlbumID
INNER JOIN Reviewers R ON AR.ReviewerID = R.ReviewerID
ORDER BY T.AlbumID, T.TrackNum
```

```
SELECT T.AlbumID, T.TrackNum, R.ReviewerID
FROM Tracks T
INNER JOIN TrackReviews TR ON (T.AlbumID = TR.AlbumID
                               AND T.TrackNum = TR.TrackNum)
INNER JOIN Reviewers R ON TR.ReviewerID = R.ReviewerID
ORDER BY T.AlbumID, T.TrackNum
```

AlbumID	TrackNum	ReviewerID
1	1	1
1	1	5
1	1	3
1	2	3
1	2	1
1	2	5
2	1	4
2	2	4
3	1	2
3	1	5
3	2	2
3	2	5
4	1	3
4	2	3
4	3	3
5	1	1
5	1	5
5	2	1
5	2	5
5	3	1
5	3	5
6	1	3
6	1	2
7	1	3
7	1	4
8	1	1
8	1	2
8	2	1
8	2	2

Resultado da primeira consulta

AlbumID	TrackNum	ReviewerID
1	1	1
1	1	3
1	2	5
1	2	3
2	1	2
2	2	1
2	2	3
3	1	4
3	2	4
3	2	2
4	1	4
4	2	3
4	3	5
5	1	2
5	2	4
5	3	3
6	1	5
7	1	5
8	1	1
8	2	4

Resultado da segunda consulta

4.b) Para a base de dados da figura exemplifique uma consulta que evidencie a junção com agregações de dados de 2 tabelas, com dados e resultados.

Resposta:

```
SELECT R.ReviewerID,SUM(AR.Ranking), SUM(TR.Ranking)
FROM Reviewers R
INNER JOIN AlbumReviews AR ON R.ReviewerID = AR.ReviewerID
INNER JOIN TrackReviews TR ON R.ReviewerID = AR.ReviewerID
GROUP BY R.ReviewerID
```

Resultam os seguintes valores:

ReviewerID	somaAlbumReviews	somaTrackReviews
1	320	396
2	480	396
3	480	528
4	220	264
5	340	396

Soma dos Rankings atribuídos por cada Reviewer nos Albuns e nas Tracks (os valores não estão corretos)

Quando, na realidade deveria obter-se:

ReviewerID	somaAlbumReviews	somaTrackReviews
1	16	18
2	24	17
3	24	37
4	11	32
5	17	28

Soma dos Rankings atribuídos por cada Reviewer nos Albuns e nas Tracks (os valores estão corretos)

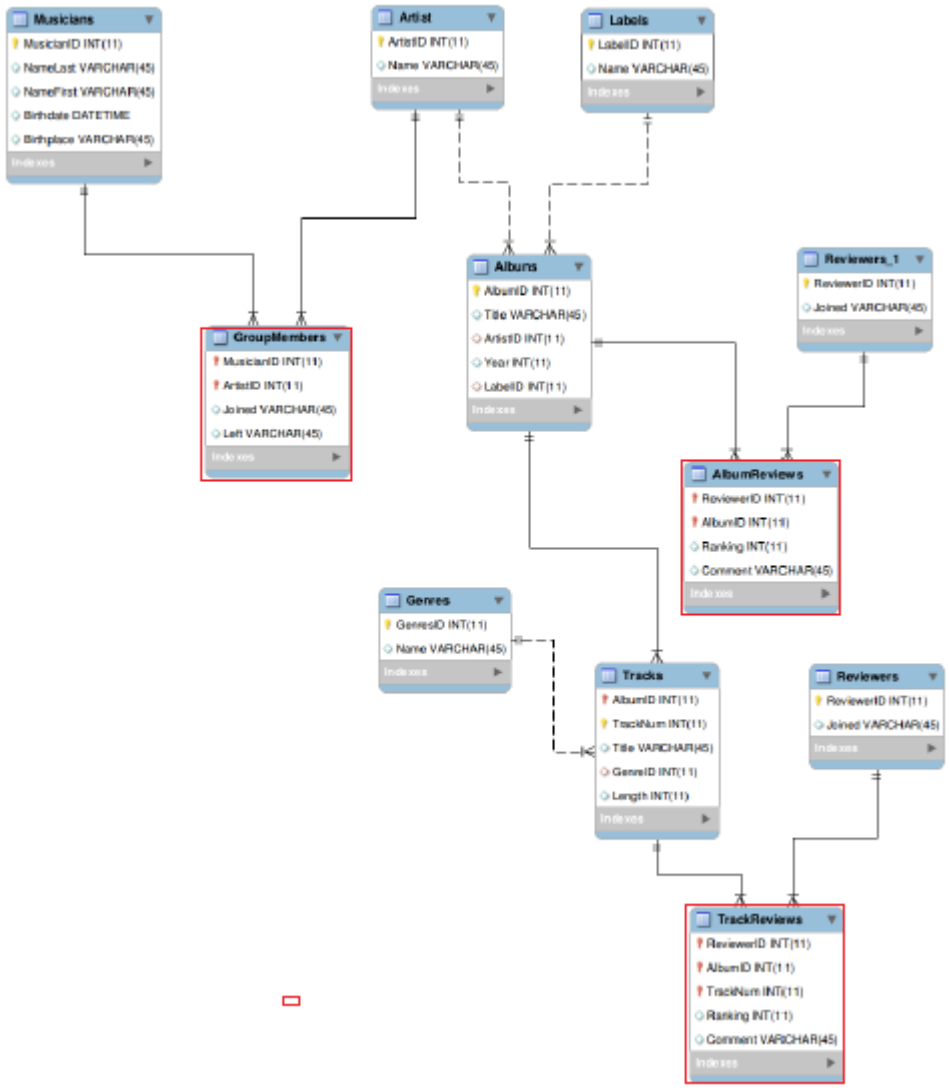
Critérios de correção:
- 4.a) 2 décimas, consultas com múltiplos caminhos
- 4.a) 3 décimas, dados e resultados
- 4.b) 3 décimas, consulta com ‘connection traps’
- 4.b) 2 décimas, dados e resultados
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

5) (1 valor) Para a mesma base de dados transacional da alínea anterior:

5.a) Reutilize a base de dados transacional na 3ª forma normal. Faça o carregamento de dados. Na representação gráfica das ligações de 1:N, a tabela com uma única linha é desenhada em cima e a tabela com várias linhas é desenhada por baixo.

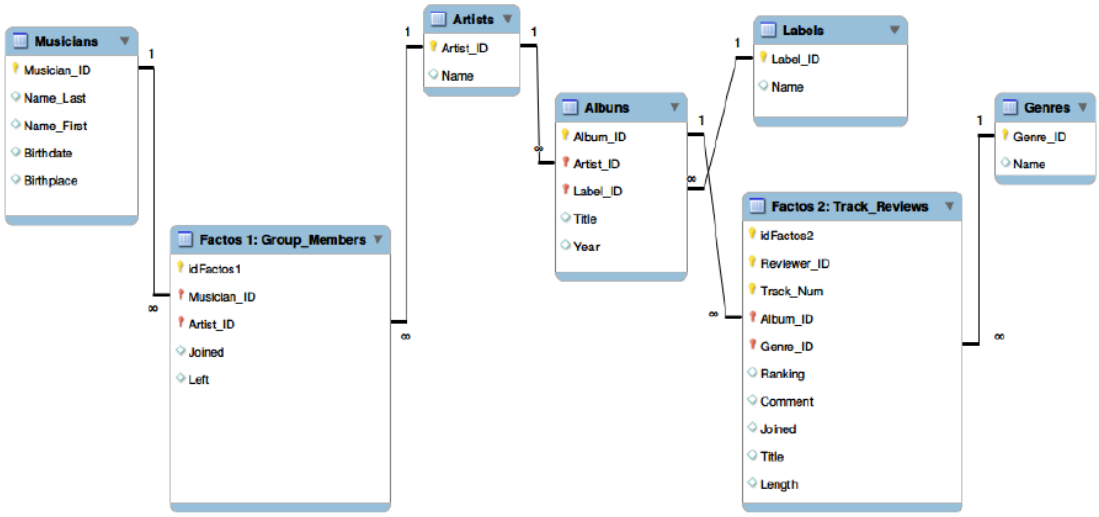
5.b) De seguida remova os caminhos múltiplos que eventualmente existam no esquema de base de dados, criando uma poli-árvore. Na representação gráfica das ligações de 1:N, a tabela com uma única linha é desenhada em cima e a tabela com várias linhas é desenhada por baixo.

Resposta: Repetindo a tabela 'Reviewers' obtemos uma base de dados 3FN com estrutura de 'poly-tree'. Note que existem 3 tabelas folhas (sem filhos) assinaldas a vermelho. Cada tabela folha pode representar uma tabela de factos.



5.c) Pretendemos desenhar um "Data Warehouse" relacional em estrela ou em constelação, i.e. com duas ou mais estrelas com a maior granularidade possível. Defina a(s) tabela(s) de factos e mostre a tabela depois da desnormalização dos dados. Defina as dimensões com os níveis de agregação para o "Data Warehouse" relacional. Apresente a(s) tabela(s) de factos associada às dimensões. Ao juntar as tabelas transacionais tenha em consideração as eventuais armadilhas referidas na pergunta anterior.

Resposta: DW com 2 tabelas de factos



5.d) Crie duas perguntas e traduza para SQL com Pivot Tables utilizando pelo menos duas dimensões (OLAP).

Resposta: Consulta para encontrar a Frequência de Artistas e Músicos

```
TRANSFORM Count ([ Factos1:Group_Members]. idFactos1)
SELECT [ Factos 1: Group_Members].[Musician_ID ],
        Count ([ Factos1:Group_Members ]. idFactos1 )
FROM [ Factos 1: Group_Members]
GROUP BY [ Factos 1: Group_Members ].[ Musician_ID ]
PIVOT [ Factos 1: Group_Members ].[ Artist_ID ];
```

Critérios de correção:
- 5.a) 1 décimas, base de dados 3FN
- 5.b) 3 décimas, base dados 'poly-tree'
- 5.c) 3 décimas, DW com 2 a 3 tabelas de factos (1 décima com 1 tabela)
- 5.d) 3 décimas, consultas OLAP
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%