

EXAME 2016 - RECURSO

1. Qual a expressão em álgebra relacional equivalente à seguinte consulta SQL:

SELECT a FROM T1 JOIN T2 WHERE T1.a>T2.b AND c=-1;

- a. $\sigma_{(c=-1 \wedge T1.a > T2.b)}(\pi_a(T1 \bowtie T2))$
- b. $\pi_a((\sigma_{(c=-1)}(T1)) \bowtie_{(T1.a > T2.b)} T2)$
- c. $\sigma_{(c=-1)}(\pi_a(T1 \bowtie_{(T1.a > T2.b)} T2))$
- d. $\pi_a(\sigma_{(c=-1)}(T1 \bowtie_{(T1.a > T2.b)} T2))$

2. Em LDD-SQL, FOREIGN KEY e REFERENCES são

- a. a forma de definir uma chave externa e um campo referência, respetivamente
- b. duas formas de definir chaves externas
- c. duas formas de definir chaves candidatas
- d. a forma de definir uma chave externa e uma chave candidata, respetivamente

3. Um escalonamento recuperável sobre um conjunto de transações

- a. não gera bloqueios
- b. garante que a base de dados fica consistente
- c. só comete uma transação T_i se as transações T_j cujas alterações foram usadas por T_i já tiverem sido cometidas
- d. resulta numa base de dados igual à que seria obtida se as transações fossem executadas em série

4. Uma base de dados NoSQL do tipo “key value store” consiste:

- a. numa estrutura organizada por colunas
- b. em 2 tabelas, uma com as propriedades dos nós e outra com as suas relações
- c. num base sem qualquer esquema explícito
- d. simplesmente numa tabela com 2 colunas, a chave e o valor respetivo

5. O esquema dos dados numa base de dados relacional e num data warehouse é de natureza diferente, nomeadamente relacional e multidimensional:

- a. mas, tipicamente, são ambas implementadas com software de gestão de bases de dados relacionais
- b. tendo, por isso, de ser implementadas com software de gestão de bases de dados de tipo diferentes
- c. sendo, por isso, tipicamente implementadas com software de gestão de bases de dados de tipos diferentes, embora seja possível usar software do mesmo tipo
- d. mas podem ser ambas implementadas com software de gestão de bases de dados relacionais e multidimensionais

6.

6.1. Considere o esquema relacional $R(A,B,C,D,E,F,G,H)$ e as seguintes dependências funcionais: $ABC \rightarrow DGH$; $F \rightarrow BA$; $B \rightarrow E$; $D \rightarrow H$; $E \rightarrow BC$; $D \rightarrow F$.

Qual das seguintes dependências funcionais não pode ser obtida?

- a. $AE \rightarrow G$
- b. $BCD \rightarrow F$
- c. $AC \rightarrow FH$
- d. $ACD \rightarrow FG$

6.2. Verifique se relação R se encontra na forma normal de Boyce-Codd. Justifique.

7. Uma hierarquia de classes representa:

- a. uma associação reflexiva com base na relação de especialização
- b. uma associação com base na relação de agregação
- c. um conjunto de classes organizadas com base na relação de especialização
- d. um conjunto de classes organizadas com base na relação de composição

8. A relação Empresa(códigoEmpresa, NúmeroIdentificaçãoFiscal, NomeEMorada, RamoAtividade) não está na 1FN porque:

- a. nem todos os atributos dependem da chave
- b. nenhuma das outras alternativas porque a relação está na 1FN
- c. tem duas chaves candidatas
- d. nem todos os atributos são atômicos

Informação

Os estudantes da FEUP decidiram organizar a sua rede social usando uma base de dados com o seguinte modelo relacional:

Curso (ID, nome)

Estudante (ID, nome, curso->Curso, anoCurricular)

Amizade (ID1->Estudante, ID2->Estudante)

O estudante com *ID1* é amigo do estudante com *ID2*. Como as amizades são mútuas, se (176, 9) está na tabela Amizade, (9, 176) também está.

A execução das interrogações deve ser feita numa base de dados SQLite criada com as instruções SQL existentes no ficheiro redesocial.sql.

Deve escrever a interrogação usando SQL. Como as interrogações serão executadas usando SQLite, devem ser compatíveis com a sintaxe SQL suportada pelo SQLite.

Para cada interrogação é apresentada a resposta esperada para a base de dados associada ao ficheiro redesocial.sql. Esta informação deve ser utilizada para validar a resposta antes de a submeter. Por omissão, a saída do SQLite só apresenta colunas com 10 caracteres. O número de caracteres de cada coluna pode ser ajustado utilizando o comando:

.width <num caracteres coluna 1> <num caracteres coluna 2> ...

9.

Liste o nome de todos os estudantes que são amigos de alguém cujo nome comece por Gabriel.

nome
Ana Lopes
Joana Teixeira
Mafalda Pimentel
Luis Goncalves
Barbara Coutinho

10.

Liste o nome de cada estudante inscrito no MIEIC cujo nome contenha a letra 'a'.

Estudante
Ana Lopes
Joao Nunes
Maria Felisberta
Cristiano Rodrigo
Carla Silva
Carlos Rodrigues
Mafalda Pimentel

11.

Liste o nome dos estudantes com amigos em todos os anos curriculares.

Nome
Gabriel Maria

12.

Considere que, para todos os casos em que A é amigo de B e B é amigo de C, A é amigo de C. Crie uma tabela chamada AmizadeTransitiva que contenha estas novas relações de amizade. Não adicione amizades duplicadas, amizades que já existem na tabela Amizade ou auto-amizades. No final, a tabela AmizadeTransitiva deverá conter os seguintes tuplos:

ID1	ID2
201101025	201101304
201101247	201101510
201101247	201101689
201101247	201101934
201101304	201101025
201101304	201101316
201101304	201101501
201101304	201101510
201101304	201101709
201101316	201101304
201101316	201101501
201101316	201101510
201101316	201101709
201101381	201101501
201101381	201101689
201101381	201101709
201101381	201101911
201101501	201101304
201101501	201101316
201101501	201101381
201101501	201101709
201101510	201101247
201101510	201101304
201101510	201101316
201101510	201101709
201101510	201101304
201101510	201101316
201101510	201101709
201101510	201101304
201101510	201101316
201101510	201101709
201101510	201101316
201101510	201101709
201101510	201101782
201101661	201101689
201101661	201101782
201101661	201101934
201101689	201101247
201101689	201101381
201101689	201101661
201101689	201101934
201101709	201101304
201101709	201101316
201101709	201101381
201101709	201101501
201101709	201101510
201101709	201101782
201101709	201101911
201101782	201101510
201101782	201101661
201101782	201101709
201101782	201101934
201101911	201101381
201101911	201101709
201101911	201101934
201101934	201101247
201101934	201101661
201101934	201101689
201101934	201101782
201101934	201101911

13.

Liste todos os pares de amigos de cursos diferentes. A listagem não deve conter pares repetidos, nem pares simétricos. Se contiver o par (Miguel, Nuno) não deve ter o par (Nuno, Miguel).

Nome	Nome
Ana Lopes	Gabriel Maria
Maria Felisberta	Diogo Teixeira
Carla Silva	Sergio Carvalho
Sergio Carvalho	Pedro Nunes
Joana Teixeira	Pedro Nunes
Mafalda Pimentel	Cristina Ribeiro
Mafalda Pimentel	Gabriel Maria
Carla Silva	Luis Goncalves
Mafalda Pimentel	Barbara Coutinho
Sergio Carvalho	Carlos Rodrigues

14.

Crie os gatilhos necessários para garantir que a amizade é mútua sempre que haja inserções ou eliminações na tabela Amizade. Crie também um gatilho que impeça atualizações à tabela Amizade.

Soluções:

1. Opção d: $\pi_{\text{a}}(\sigma_{\text{c}=-1}(T1 \bowtie (T1.a > T2.b) T2))$
2. Opção b: duas formas de definir chaves externas
3. Opção c: só comete uma transação T_i se as transações T_j cujas alterações foram usadas por T_i já tiverem sido cometidas
4. Opção d: simplesmente numa tabela com 2 colunas, a chave e o valor respetivo
5. Opção a: mas, tipicamente, são ambas implementadas com software de gestão de bases de dados relacionais
6. 6.1. Opção c: AC->FH
7. Opção c: um conjunto de classes organizadas com base na relação de especialização
8. Opção d: nem todos os atributos são atómicos
- 9.

```
SELECT e1.nome FROM Estudante AS e1, Estudante AS e2, Amizade
WHERE ID1 = e1.ID AND ID2 = e2.ID
AND e2.nome LIKE "Gabriel%";
```

10.

```
SELECT Estudante.nome AS Estudante FROM Estudante
WHERE nome LIKE "%a%"
AND curso IN (SELECT ID FROM Curso WHERE nome LIKE "MIEIC");
```


11.

```
SELECT nome FROM (
    SELECT nome, COUNT(*) AS NumAnos FROM (
        -- Anos curriculares em que cada ID tem amigos
        SELECT DISTINCT e1.nome, e1.ID, e2.anoCurricular
        FROM Estudante AS e1, Estudante AS e2, Amizade
        WHERE ID1 = e1.ID AND ID2 = e2.ID
    ) GROUP BY ID
) WHERE NumAnos = 5;
```

12.

```
DROP TABLE IF EXISTS AmizadeTransitiva;

CREATE TABLE AmizadeTransitiva(
    ID1 INT REFERENCES Estudante(ID),
    ID2 INT REFERENCES Estudante(ID),
    PRIMARY KEY (ID1, ID2)
);

INSERT INTO AmizadeTransitiva(ID1, ID2)
SELECT DISTINCT a1.ID1, a2.ID2 FROM Amizade as a1, Amizade as a2
WHERE a1.ID2 = a2.ID1
AND a1.ID1 != a2.ID2
AND (a1.ID1, a2.ID2) NOT IN (SELECT * FROM Amizade)
ORDER BY a1.ID1, a2.ID2;

SELECT * FROM AmizadeTransitiva;
```

13.

```
SELECT nome1, nome2 FROM (
    -- Pares de amigos de diferentes cursos
    SELECT ID1, ID2, e1.nome AS nome1, e2.nome AS nome2
    FROM Estudante AS e1, Estudante AS e2, Amizade AS a
    WHERE e1.ID=a.ID1 AND e2.ID=a.ID2 AND e1.curso != e2.curso
) WHERE ID1 < ID2 OR (ID2, ID1) NOT IN (
    -- Pares de amigos de diferentes cursos
    SELECT ID1, ID2
    FROM Estudante AS e1, Estudante AS e2, Amizade AS a
    WHERE e1.ID=a.ID1 AND e2.ID=a.ID2 AND e1.curso != e2.curso
);
```

14.

```
DROP TRIGGER IF EXISTS AmizadeInsert;
DROP TRIGGER IF EXISTS AmizadeDelete;
DROP TRIGGER IF EXISTS AmizadeUpdate;

CREATE TRIGGER AmizadeInsert AFTER INSERT ON Amizade
FOR EACH ROW BEGIN
    INSERT INTO Amizade
    SELECT ID2 AS ID1, ID1 AS ID2 FROM New;
END;

CREATE TRIGGER AmizadeDelete AFTER DELETE ON Amizade
FOR EACH ROW BEGIN
    DELETE FROM Amizade
    WHERE ID2=Old.ID1 AND ID1=Old.ID2;
END;

CREATE TRIGGER AmizadeUpdate BEFORE UPDATE ON Amizade
BEGIN
    SELECT raise(rollback, 'Amizade table can not be updated');
END;
```