

Época Normal – 2º Teste – Sem consulta –

Leia com atenção a informação constante desta página, enquanto espera a indicação do docente para começar a resolução do teste.

Este enunciado é composto por:

- Uma Folha de Rosto (esta)
- Uma Folha de Respostas
- Cinco Páginas de Perguntas
- Cinco Páginas de Rascunho

O teste é composto por dois grupos de perguntas:

- GRUPO I: Composto por 5 perguntas de resposta curta valendo 8 valores no total.
- GRUPO II: Composto por 10 perguntas de escolha múltipla valendo 12 valores no total.

GRUPO I:

- A simplicidade e clareza das respostas neste grupo contarão na avaliação. Poderá inclusivamente ter uma cotação de 0 valores numa pergunta se a sua solução for muito mais complicada do que o necessário.
- Todas as perguntas deste grupo têm aproximadamente a mesma cotação.

GRUPO II:

- Cada pergunta tem um número variado de respostas possíveis, onde **apenas uma está correcta**.
- Cada resposta correcta vale **1,2 valores**.
- **As respostas incorrectas descontam, de forma progressiva, de acordo com a seguinte regra:**
 - Se errar $n > 0$ respostas, terá um desconto de $\sum_{k=1}^n (0,2k - 0,1)$ valores.
 - Ou seja
 - a 1ª resposta errada desconta 0,1 valores,
 - a 2ª resposta errada desconta 0,3 valores (num total de 0,4 valores de desconto)
 - a 3ª resposta errada desconta 0,5 valores (num total de 0,9 valores de desconto)
 - a 4ª resposta errada desconta 0,7 valores (num total de 1,6 valores de desconto)
 - a 5ª resposta errada desconta 0,9 valores (num total de 2,5 valores de desconto)
 - ...
- A cotação mínima no GRUPO II é de 0 valores.

Preenchimento:

- Todas as respostas deverão ser dadas na folha de respostas fornecida.
- As zonas sombreadas da folha de respostas não deverão ser preenchidas. Para contribuir para a legibilidade da folha de respostas, minimizando rasuras, agradece-se a utilização do espaço de rascunho fornecido (quer ao longo do enunciado, quer no fim do mesmo) antes de escrever a resposta final na folha de respostas.
- No fim de 2 horas de teste **os docentes recolherão apenas a folha de respostas**.

Aconselha-se que veja todas as perguntas do teste antes de começar a sua resolução, para melhor planear a estratégia de resolução. Tem aproximadamente 8 minutos por pergunta pelo que não deve demorar demasiado tempo em cada uma.

Boa Sorte!

Número:

PARTE I

I.1

I.2

I.4 Decomposição:

Preserva as DFs (sim/não)?

Justificação:

I.5

I.6

PARTE II

[illegible]

GRUPO I

Voltemos ao cenário do 1º Teste onde considerámos a seguinte base de dados que armazena informação sobre vários ministérios (onde os atributos da chave primária de cada relação estão sublinhados) e os orçamentos de estado:

ministérios ({IdM, NomeM, IdP})
pessoas ({IdP, NomeP})

orçamentos ({IdO, Ano, DataAprovação})
verbas ({IdO, IdM, Valor})

A relação *pessoas* guarda a informação sobre as várias pessoas, tendo para cada uma informação sobre o seu identificador e nome. A relação *ministérios* guarda a informação sobre os vários ministérios, tendo para cada um deles informação sobre o seu identificador, nome, e identificador da pessoa que o tutela i.e. do seu ministro. A relação *orçamentos* guarda a informação sobre os vários orçamentos, tendo para cada um deles o seu identificador, o ano a que diz respeito, e a data de aprovação. Para simplificar, assume-se que só existe um orçamento por ano. Por fim, a relação *verbas* guarda a informação sobre o valor da verba orçamentada (em M€) para cada ministério em cada orçamento.

Implemente em SQL, sem recorrer a triggers, as seguintes restrições de integridade (Não se esqueça que a simplicidade da solução é um dos critérios de avaliação.):

I.1 Cada pessoa não pode tutelar mais do que um ministério.

I.2 Todos os ministérios têm verbas orçamentadas em todos os orçamentos.

XX

Considere a relação *imprensa* ({IdM, IdE, IdA, IdG}) que pretende manter informação sobre os assessores para a imprensa dos vários ministérios. Nesta relação, *IdM* representa um ministério, *IdE* um edifício, *IdA* um assessor de imprensa e *IdG* um gabinete. A relação *imprensa* tem o seguinte conjunto de dependências funcionais $F = \{IdA \rightarrow IdE; IdA \rightarrow IdG; IdM, IdE \rightarrow IdA; IdG \rightarrow IdE\}$.

I.3 Apresente uma decomposição da relação *imprensa* para a Forma Normal de Boyce-Codd, usando o algoritmo dado nas aulas, indicando, com justificação, se a decomposição preserva as dependências funcionais.

XX

Considere o seguinte XML DTD:

```
<!ELEMENT orçamentos((ministério*,orçamento+))>
  <!ELEMENT ministério (nome,ministro?)>
  <!ATTLIST ministério
    IdM      ID      #REQUIRED>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT ministro (#PCDATA)>
  <!ELEMENT orçamento (verba+)>
  <!ATTLIST orçamento
    Ano      CDATA    #IMPLIED>
  <!ELEMENT verba (valor)>
  <!ATTLIST verba
    IdM      IDREF #REQUIRED>
  <!ELEMENT valor (#PCDATA)>
```

I.4. Apresente um documento XML, o mais pequeno que conseguir (i.e. contendo o menor número de elementos e atributos), que esteja de acordo com o DTD acima.

I.5. Apresente uma expressão XPath que devolva os ministérios que tenham tido verbas orçamentadas em 2010 superiores a 20M. Nota: o resultado da expressão deve devolver os elementos “ministério” e não apenas o identificador “IdM”.

GRUPO II

II.1. Considere a tabela P(a, b, c) criada e instanciada com os seguintes comandos:

```
CREATE TABLE P(a INT, b INT, c INT, PRIMARY KEY (a,b));  
INSERT INTO P VALUES(4,3,8);
```

Posteriormente foi implementado o seguinte *trigger*:

```
CREATE TRIGGER t1  
BEFORE INSERT ON P FOR EACH ROW  
BEGIN  
    IF :NEW.a < :NEW.b  
    THEN UPDATE P SET P.a = P.a-1, P.b = P.b+1, P.c = :NEW.c+2;  
    ELSE INSERT INTO P VALUES(:NEW.a-1, :NEW.b+1, :NEW.c);  
    END IF;  
END;
```

Considere os três comandos SQL:

- I. INSERT INTO P VALUES(3,4,8);
- II. INSERT INTO P VALUES(10,1,6);
- III. UPDATE P SET P.a = P.a - 1, P.b = P.b + 1, P.c = 8;

Quais dos comandos anteriores fazem a linha (3,4,8) pertencer à tabela P?

- | | | | |
|--------------|----------------|--------------------|---------------------|
| (A) Nenhum | (C) Apenas II | (E) Apenas I e II | (G) Apenas II e III |
| (B) Apenas I | (D) Apenas III | (F) Apenas I e III | (H) Todos |

XX

II.2. Considere as seguintes tabelas:

```
CREATE TABLE A(w INT PRIMARY KEY);  
CREATE TABLE B(x INT PRIMARY KEY,  
                FOREIGN KEY (x) REFERENCES A ON DELETE SET NULL);  
CREATE TABLE C(y INT,  
                FOREIGN KEY (y) REFERENCES A ON UPDATE CASCADE);  
CREATE TABLE D(z1 INT,  
                z2 INT,  
                FOREIGN KEY (z1) REFERENCES B ON DELETE CASCADE,  
                FOREIGN KEY (z2) REFERENCES A);
```

Considere os seguintes scripts:

- I. DELETE FROM D; DELETE FROM C; DELETE FROM B; DELETE FROM A;
- II. DELETE FROM D; DELETE FROM C; DELETE FROM A; DELETE FROM B;
- III. DELETE FROM C; DELETE FROM B; DELETE FROM D; DELETE FROM A;

Quais dos scripts esvaziam todas as tabelas, quaisquer que sejam as suas instâncias, sem produzirem erro?

- | | | | |
|--------------|----------------|--------------------|---------------------|
| (A) Nenhum | (C) Apenas II | (E) Apenas I e II | (G) Apenas II e III |
| (B) Apenas I | (D) Apenas III | (F) Apenas I e III | (H) Todos |

Espaço de rascunho:

II.3. Considere a base de dados do grupo I.

Quais das seguintes asserções implementam a seguinte restrição de integridade: “Entre 2011 e 2014, não pode existir aumento da verba orçamentada para cada ministério.”

A1. CREATE ASSERTION A1 CHECK (NOT EXISTS (
SELECT *
FROM verbas v1, orçamentos o1
WHERE v1.IdO=o1.IdO AND o1.Ano BETWEEN 2011 AND 2014 AND
v1.Valor > SOME(SELECT v2.valor FROM verbas v2, orçamentos o2
WHERE v2.IdO=o2.IdO AND o1.Ano=o2.Ano+1 AND v1.IdM=v2.IdM)
))

A2. CREATE ASSERTION A2 CHECK (NOT EXISTS (
SELECT *
FROM verbas v1, orçamentos o1
WHERE v1.IdO=o1.IdO AND o1.Ano BETWEEN 2011 AND 2014 AND
NOT EXISTS (SELECT * FROM verbas v2, orçamentos o2
WHERE v2.IdO=o2.IdO AND o1.Ano=o2.Ano+1 AND v1.IdM=v2.IdM AND
v1.valor>v2.valor)
))

A3. CREATE ASSERTION A3 CHECK (NOT EXISTS (
SELECT *
FROM verbas v1
WHERE v1.Ano BETWEEN 2011 AND 2014 AND
v1.Valor > SOME(SELECT v2.valor FROM verbas v2
WHERE v1.Ano=v2.Ano+1)
))

- | | | | |
|---------------|---------------|--------------------|--------------------|
| (A) Nenhuma | (C) Apenas A2 | (E) Apenas A1 e A2 | (G) Apenas A2 e A3 |
| (B) Apenas A1 | (D) Apenas A3 | (F) Apenas A1 e A3 | (H) Todas |

XX

II.4. Considere a relação T(A) que inicialmente contém 3 tuplos { (1) , (2) , (3) } e a seguinte consulta (Nota: o operador UNION trata várias ocorrências do valor NULL como se fossem duplicados):

```
WITH RECURSIVE F(A) AS
  (SELECT A FROM T
   UNION
   SELECT A FROM F
   UNION
   SELECT SUM(a)
   FROM F F1
   WHERE 1 >= ALL(SELECT COUNT(*)FROM F F2 WHERE F2.A > F1.A)
   AND 8 > ALL(SELECT COUNT(*)FROM F)
   SELECT MAX(a) FROM F
```

Qual o resultado da consulta?

- | | | | |
|--------|--------|--------|---------|
| (A) 3 | (C) 21 | (E) 34 | (G) 96 |
| (B) 12 | (D) 24 | (F) 48 | (H) 192 |

Espaço de rascunho:

II.5. Assuma que o utilizador A é o dono da relação $T(A, B, C)$. A seguir à execução da seguinte sequência de comandos, pelos utilizadores indicados, quais os privilégios sobre T detidos pelo utilizador E?

Passo	Utilizador	Comando		
1	A	GRANT SELECT, DELETE ON T TO B WITH GRANT OPTION		
2	A	GRANT SELECT, DELETE ON T TO C WITH GRANT OPTION		
3	B	GRANT SELECT(A,B), DELETE ON T TO D WITH GRANT OPTION		
4	C	GRANT SELECT(A,C) ON T TO D WITH GRANT OPTION		
5	D	GRANT SELECT(A), DELETE ON T TO E		
6	A	REVOKE SELECT, DELETE ON T FROM B CASCADE		
I.	SELECT (A)	II. SELECT (B)	III. SELECT (C)	IV. DELETE
(A) Nenhum	(E) Apenas IV	(I) Apenas II e III	(M) Apenas I, II e IV	
(B) Apenas I	(F) Apenas I e II	(J) Apenas II e IV	(N) Apenas I, III e IV	
(C) Apenas II	(G) Apenas I e III	(K) Apenas III e IV	(O) Apenas II, III e IV	
(D) Apenas III	(H) Apenas I e IV	(L) Apenas I, II e III	(P) I, II, III e IV	

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.6. Quando é que se opta por fazer uma decomposição para a 3ª Forma Normal em vez de se fazer uma decomposição para a Forma Normal de Boyce-Codd?

- (A) Quando a decomposição para a Forma Normal de Boyce-Codd resulta numa elevada redução da eficiência na verificação da integridade dos dados.
- (B) Quando se pretende evitar redundância, mesmo que à custa da redução da eficiência na verificação da integridade dos dados.
- (C) Quando se pretende garantir uma decomposição sem perdas, mesmo que para tal não se garanta a preservação das dependências funcionais.
- (D) Quando não é possível fazer uma decomposição para a Forma Normal de Boyce-Codd.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.7. Considere o esquema de relação $R(A, B, C, D, E, F)$ com dependências funcionais:

$AB \rightarrow C$

$DE \rightarrow F$

$F \rightarrow B$

$C \rightarrow E$

Qual das seguintes frases é verdadeira em relação a R:

- (A) AB é chave candidata
- (B) ABDE, ACD e ADF são todas as chaves candidatas
- (C) ACD e ACDF são super-chaves
- (D) Nenhuma das anteriores.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.8. Considere o esquema de relação $R(A, B, C, D, E, F)$ com o seguinte conjunto de dependências funcionais na forma canónica $F=\{B \rightarrow AF; C \rightarrow D; D \rightarrow CE\}$.

Qual das seguintes decomposições de R poderia ser o resultado da aplicação do algoritmo de decomposição para a 3ª Forma Normal dado nas aulas?

- (A) $R_1(A,B,F); R_2(C,D,E)$
- (B) $R_1(A,B,F); R_2(C,D,E); R_3(C,D)$
- (C) $R_1(A,B,F); R_2(C,D,E); R_3(B,D)$
- (D) $R_1(A,B,F); R_2(C,D,E); R_3(B,C,D)$
- (E) Nenhum dos anteriores.

Espaço de rascunho:

As próximas duas perguntas são baseadas no seguinte documento XML “BIBLIOTECA.XML”:

```
<BIB>
  <LIVRO isbn="1234">
    <TITULO>TCP/IP Illustrated</TITULO>
    <AUTOR>Stevens</AUTOR>
    <EDITORIA>Addison-Wesley</EDITORIA>
  </LIVRO>
  <LIVRO>
    <TITULO>Advanced Unix Programming</TITULO>
    <AUTOR>Stevens</AUTOR>
  </LIVRO>
  <LIVRO isbn="5678">
    <TITULO>Data on the Web</TITULO>
    <AUTOR>Abiteboul</AUTOR>
    <AUTOR>Buneman</AUTOR>
    <AUTOR>Suciu</AUTOR>
    <ANO>2000</ANO>
  </LIVRO>
</BIB>
```

XX

II.9. O resultado da expressão XQuery:

```
let $d := document("BIBLIOTECA.XML")
for $b in $d/BIB/LIVRO
where $b/AUTOR = "Abiteboul"
return {$b/AUTOR}
```

é uma sequência de quantos nós do tipo AUTOR?

- (A) 0 (B) 1 (C) 3 (D) 9

XX

II.10. Num DTD que o documento “BIBLIOTECA.XML” satisfaça, qual das seguintes declarações de elemento poderia encontrar?

- (A) <!ELEMENT LIVRO (TITULO*, AUTOR*, EDITORA?, ANO*)>
(B) <!ELEMENT LIVRO (TITULO, AUTOR+, (EDITORIA | ANO))>
(C) <!ELEMENT LIVRO (TITULO, AUTOR*, EDITORA+, ANO?)>
(D) <!ELEMENT LIVRO (TITULO?, AUTOR?, EDITORA*, ANO?)>

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho: