

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!

Nome:

Número:

PARTE I

I.1

I.2

I.3

I.4 Decomposição:

Preserva as DFs (sim/não)?

Justificação:

1.5

PARTE II

[illegible]

GRUPO I

Voltemos ao cenário do 1º Teste onde considerámos a seguinte base de dados, que armazena informação sobre uma edição da Volta a Portugal em Bicicleta (onde os atributos da chave primária de cada relação estão sublinhados):

<code>equipas ({<u>IdE</u>, NomeE})</code>	<code>etapas ({<u>Número</u>, Partida, Chegada, Distancia})</code>
<code>ciclistas ({<u>IdC</u>, NomeC, IdE})</code>	<code>resultados ({<u>Número</u>, <u>IdC</u>, Tempo})</code>

A relação `equipas` guarda a informação sobre as várias equipas, tendo para cada uma informação sobre o seu identificador e nome. A relação `ciclistas` guarda a informação sobre os vários ciclistas, tendo para cada um informação sobre o seu identificador, nome e equipa. A relação `etapas` guarda a informação sobre as várias etapas da Edição da volta a Portugal, tendo para cada uma delas informação sobre o seu número de ordem, localidades de partida e de chegada, bem como a distância, em quilómetros, da etapa. A relação `resultados` guarda a informação sobre os resultados individuais de cada ciclista em cada etapa, tendo para cada um deles o número da etapa, o identificador do ciclista, e o tempo por ele obtido (hh:mm:ss). Se um ciclista não terminou uma etapa, não existirá o tuplo correspondente na relação `resultados`.

Verificou-se que a Base de Dados tem alguns dados inconsistentes. Nomeadamente, a tabela `resultados` contém tuplos correspondentes aos tempos obtidos por ciclistas nalgumas etapas, sem que tenham completado todas as etapas anteriores. Vamos resolver este problema.

I.1 Apresente o comando SQL para criar uma vista `desistente (IdC, Número)` que contenha tuplos para cada ciclista e etapa (`IdC, Número`), tal que esse ciclista tenha terminado essa etapa, mas não tenha terminado a etapa anterior. A vista não deve conter tuplos referentes à primeira etapa.

Nas próximas duas alíneas, poderá usar a vista `desistente (IdC, Número)`, mesmo que não tenha respondido à alínea anterior.

I.2 Apresente o comando SQL para eliminar da tabela `resultados` todos os tuplos correspondentes a resultados obtidos por um ciclista em todas as etapas posteriores à sua desistência (primeira etapa que não tenha completado).

I.3 Apresente o comando SQL, sem recorrer a triggers, que implemente as restrições de integridade necessárias para garantir que a Base de Dados não voltará a conter resultados de ciclistas após a sua desistência. (Não se esqueça que a simplicidade da solução é um dos critérios de avaliação.).

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Considere a relação `médicos ({IdE, Número, IdM, Carro})` que pretende manter informação sobre os médicos que prestam assistência durante a volta. Nesta relação, `IdE` representa uma equipa, `Número` uma etapa, `IdM` um médico e `Carro` um dos carros da organização da prova. A relação `médicos` tem o seguinte conjunto de dependências funcionais $F = \{IdM \rightarrow Número; IdM \rightarrow Carro; IdE, Número \rightarrow IdM; Carro \rightarrow Número\}$.

I.4 Apresente uma decomposição da relação `médicos` para a Forma Normal de Boyce-Codd, usando o algoritmo dado nas aulas, indicando, com justificação, se a decomposição preserva as dependências funcionais.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Considere o seguinte XML DTD:

<code><!ELEMENT volta (equipa+,etapa+)></code>	<code><!ELEMENT partida (#PCDATA)></code>
<code><!ELEMENT equipa (nome,ciclista+)></code>	<code><!ELEMENT chegada (#PCDATA)></code>
<code><!ATTLIST equipa IdE ID #REQUIRED></code>	<code><!ELEMENT distancia (#PCDATA)></code>
<code><!ELEMENT nome (#PCDATA)></code>	<code><!ELEMENT resultado (tempo)></code>
<code><!ELEMENT ciclista (nome)></code>	<code><!ELEMENT tempo (#PCDATA)></code>
<code><!ATTLIST ciclista IdC ID #REQUIRED></code>	<code><!ATTLIST resultado IdC IDREF #REQUIRED></code>
<code><!ELEMENT etapa (partida,chegada,distancia,resultado*)></code>	

I.5. Considerando documentos XML de acordo com o DTD acima, apresente uma expressão XPath que devolva os nomes dos ciclistas que tenham terminado uma etapa com chegada a Lisboa.

GRUPO II

II.1. Considere as tabelas criadas e instanciadas com os seguintes comandos:

```
CREATE TABLE P(a INT, b INT, PRIMARY KEY (a));
CREATE TABLE Q(x INT, y INT, PRIMARY KEY (x),
                FOREIGN KEY (y) REFERENCES P ON DELETE SET NULL);
INSERT INTO P VALUES(4,3);
INSERT INTO Q VALUES(3,4);
```

Posteriormente foi implementado o seguinte *trigger*:

```
CREATE TRIGGER t1
BEFORE INSERT ON P FOR EACH ROW
BEGIN
    IF :NEW.a < :NEW.b
    THEN UPDATE Q SET Q.y=NULL;
    ELSE DELETE FROM P WHERE P.a=:NEW.b+1;
    END IF;
END;
```

Considere os três comandos SQL:

- I. INSERT INTO P VALUES(3,4);
- II. INSERT INTO P VALUES(4,2);
- III. DELETE FROM P WHERE a=4;

Quais dos comandos anteriores fazem a linha (3,NULL) pertencer à tabela Q?

- | | | | |
|--------------|----------------|--------------------|---------------------|
| (F) Nenhum | (H) Apenas II | (J) Apenas I e II | (L) Apenas II e III |
| (G) Apenas I | (I) Apenas III | (K) Apenas I e III | (M) Todos |

XX

II.2. Considere as seguintes tabelas:

```
CREATE TABLE A(x INT PRIMARY KEY);
CREATE TABLE B(z INT PRIMARY KEY);
CREATE TABLE C(y INT,
                FOREIGN KEY (y) REFERENCES A ON DELETE SET NULL,
                FOREIGN KEY (y) REFERENCES B ON DELETE CASCADE);
CREATE TABLE D(w1 INT PRIMARY KEY,
                w2 INT NOT NULL,
                FOREIGN KEY (w1) REFERENCES A,
                FOREIGN KEY (w2) REFERENCES B ON DELETE SET NULL);
```

Considere os seguintes scripts:

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

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. 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), DELETE ON T TO D WITH GRANT OPTION
4	C	GRANT SELECT(C), DELETE ON T TO D WITH GRANT OPTION
5	D	GRANT SELECT(A,C), 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.4. Considere a seguinte consulta recursiva:

```
WITH RECURSIVE F(A,B,C,D) AS
  (SELECT 0 AS A, 0 AS B, 1 AS C, 1 AS D
   UNION
   SELECT A+C, A+B, A, D+1 FROM F WHERE D<8)
SELECT MAX(A) FROM F
```

Qual o resultado da consulta?

- | | | | |
|-------|-------|-------|--------|
| (A) 0 | (C) 2 | (E) 5 | (G) 13 |
| (B) 1 | (D) 3 | (F) 8 | (H) 21 |

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.5. Considere o esquema de relação $R(A, B, C, D, E, F)$ com dependências funcionais:

$AB \rightarrow C$ $D \rightarrow EB$ $F \rightarrow D$ $C \rightarrow A$

Qual das seguintes frases é verdadeira em relação a R:

- (F) ABCF é chave candidata
 (G) CF e ABF são todas as chaves candidatas
 (H) FBD e ABF são super-chaves
 (I) Nenhuma das anteriores.

Espaço de rascunho:

II.6. Quais das seguintes ocorrências do atributo A são dispensáveis no seguinte conjunto de dependências funcionais?

$$\begin{array}{c} B \rightarrow AD \\ 1^a \end{array}$$

$$\begin{array}{c} BC \rightarrow A \\ 2^a \end{array}$$

$$\begin{array}{c} A \rightarrow AB \\ 3^a \quad 4^a \end{array}$$

- | | | | |
|-----------------------------|---|--|--|
| (A) Nenhuma | (E) Apenas a 4 ^a | (I) Apenas as 2 ^a e 3 ^a | (M) Apenas as 1 ^a , 2 ^a e 4 ^a |
| (B) Apenas a 1 ^a | (F) Apenas as 1 ^a e 2 ^a | (J) Apenas as 2 ^a e 4 ^a | (N) Apenas as 1 ^a , 3 ^a e 4 ^a |
| (C) Apenas a 2 ^a | (G) Apenas as 1 ^a e 3 ^a | (K) Apenas as 3 ^a e 4 ^a | (O) Apenas as 2 ^a , 3 ^a e 4 ^a |
| (D) Apenas a 3 ^a | (H) Apenas as 1 ^a e 4 ^a | (L) Apenas as 1 ^a , 2 ^a e 3 ^a | (P) Todas |

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.7. Considere o esquema de relação R(A, B, C, D, E) com o seguinte conjunto de dependências funcionais na forma canónica $F = \{D \rightarrow B; C \rightarrow A; A \rightarrow CE\}$.

Pretende-se decompor a relação R para a terceira forma normal. Quais dos esquemas de relação R3(A,C), R4(C,D) e R5(A,C,D) serão incluídos na decomposição efectuada através do algoritmo de normalização estudado nas aulas, sabendo que R1(D,B) e R2(C,A,E) foram anteriormente gerados?

- | | | | |
|---------------|---------------|--------------------|--------------------|
| (Q) Nenhum | (S) Apenas R4 | (U) Apenas R3 e R4 | (W) Apenas R4 e R5 |
| (R) Apenas R3 | (T) Apenas R5 | (V) Apenas R3 e R5 | (X) Todos |

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

II.8. Considere os seguintes programas Datalog, onde o resultado é dado, em ambos os casos, pela relação p.

Q1: $s(X) :- r(X, Y).$
 $p(X) :- q(X, Y), \text{ not } s(X).$
 Q2: $p(X) :- q(X, Y), \text{ not } r(X, Y).$

Qual das seguintes frases é verdadeira?

- (W) Q1 e Q2 produzem sempre a mesma resposta.
 (X) A frase (W) não é verdadeira e a resposta a Q1 está sempre contida na resposta a Q2.
 (Y) A frase (W) não é verdadeira e a resposta a Q2 está sempre contida na resposta a Q1.
 (Z) Nenhuma das anteriores.

Espaço de rascunho:

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

```
<X>
  <Y>
    <A i="I1" j="I1"/>
    <A i="I3" j="I3"/>
  </Y>
  <Y>
    <B i="I1" j="I2">
      <B j="I4"/>
    </B>
  </Y>
</X>
```

XX

II.9. Assumindo que “XY” está armazenado no ficheiro XY.xml e executamos a consulta Xquery:

```
for $a in doc("XY.xml")//A,
  $b in doc("XY.xml")//B[../@i = $a/@i]
return <XPTO>{ $a/@i } - { $b/@j } </XPTO>
```

Qual dos seguintes fragmentos de texto ocorre no output?

- (M) I1 - I2 (N) I1 - I4 (O) I3 - I2 (P) I3 - I4

XX

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

- (A) <!ATTLIST A i CDATA #IMPLIED j ID #REQUIRED>
<!ATTLIST B i CDATA #REQUIRED j CDATA #REQUIRED>
- (B) <!ATTLIST A i ID #IMPLIED j CDATA #REQUIRED>
<!ATTLIST B i ID #IMPLIED j CDATA #REQUIRED>
- (C) <!ATTLIST A i CDATA #IMPLIED j ID #REQUIRED>
<!ATTLIST B i CDATA #IMPLIED j IDREF #REQUIRED>
- (D) <!ATTLIST A i ID #IMPLIED j CDATA #REQUIRED>
<!ATTLIST B i IDREF #IMPLIED j ID #REQUIRED>

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho:

Espaço de rascunho: