



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
- Quatro Páginas de Perguntas
- Seis 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**.
- **Cada resposta incorrecta desconta 0,5 valores**. Ou seja, é melhor não responder do que responder errado!
 - Por exemplo, se apenas responder a 3 perguntas e estiverem todas certas, tem 3,6 valores no grupo II. Se, para além das 3 respostas certas, responder erradamente a 5 perguntas, então terá 1,1 valores no grupo II, insuficiente para atingir uma nota positiva.
- 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 e 30 segundos por pergunta pelo que não deve demorar demasiado tempo em cada uma.

Boa Sorte!

Número:

I.1

I.2

I.3

I.4

I.5

[illegible]

GRUPO I

Voltemos ao cenário do 1º teste, onde a Cooperativa dos Fogareiros decidiu criar uma base de dados para guardar informação sobre os táxis dos seus associados e corridas efectuadas. Os táxis podem ser detidos por diversos sócios que são motoristas. Nesta segunda versão, a base de dados da Cooperativa contempla a existência de:

- táxis que, para além do ano de entrada ao serviço, são identificados por um código único;
- modelos de veículos que, para além da marca e do seu consumo, são identificados por um código único;
- motoristas que, para além de terem um nome, são identificadas por um código único;
- locais, que para além de terem um nome, são identificados por um código único.

Para além desta informação, a base de dados guardará informação sobre:

- o modelo de cada táxi;
- os proprietários de cada táxi e respectiva data de entrada (tipo DATE) para a sociedade detentora do táxi;
- as corridas de cada táxi, indicando o motorista, o táxi, a data e hora do início da corrida (TIMESTAMP), a data e hora do final da corrida (TIMESTAMP), o preço e locais de partida e destino.

A base de dados da Cooperativa, com o objectivo de guardar a informação descrita, tem as seguintes relações cujos atributos **não** podem conter nulos (os atributos que compõem a chave primária encontram-se sublinhados):

Táxis({ <u>IdT</u> , AnoT, IdMd})	Modelos({ <u>IdMd</u> , MarcaMd, ConsumoMd})
Proprietários({ <u>IdT</u> , <u>IdM</u> , DataP})	Motoristas({ <u>IdM</u> , NomeM})
Corridas({ <u>IdM</u> , <u>IdT</u> , <u>HrInício</u> , HrFim, PreçoC, PartidaL, DestinoL})	Locais({ <u>IdL</u> , NomeL})

A relação **Modelos** contém informação sobre os modelos de veículos; a relação **Motoristas** contém informação sobre os motoristas; a relação **Locais** contém informação sobre os locais; a relação **Táxis** contém informação sobre os táxis bem como, para cada um, o tipo de modelo; a relação **Proprietários** indica quais os motoristas detentores de cada táxi, e respectivas datas de entrada para a sociedade; a relação **Corridas** guarda informação sobre as corridas, i.e. o motorista, o táxi, a data e a hora de início e de chegada (a inovação nesta versão da base de dados), o preço, o local de partida e local de destino (chaves externas para Locais).

I.1. Usando SQL garanta que numa corrida a data e hora final tem de ser superior à data e hora de início da corrida. Nota: Dados do tipo **TIMESTAMP** comparam-se como dados do tipo **DATE**.

I.2. Recorrendo a um **TRIGGER**, garanta que para um mesmo táxi não haja duas corridas a decorrer simultaneamente (i.e. cujos intervalos [**HrInício**, **HrFim**] se intersectem). Assuma que só há inserções de tuplos na tabela **corridas**, não podendo ser alterados posteriormente.

I.3. Considere a relação **Fogareiros**({IdT, AnoT, IdM, NomeM, DataP, IdMd, MarcaMd, ConsumoMd}) obtida por junção de **Táxis**, **Proprietários**, **Modelos** e **Motoristas**. Indique uma chave candidata para **Fogareiros**. Justifique recorrendo às dependências funcionais obtidas a partir das chaves primárias das relações **Táxis**, **Proprietários**, **Modelos** e **Motoristas**.

Considere agora o seguinte XML DTD onde se ignora a data de entrada para a sociedade dos proprietários:

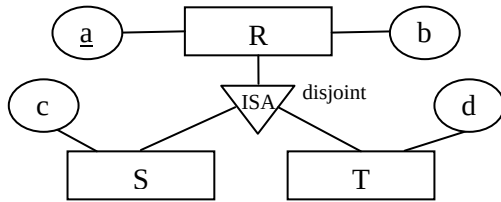
```
<!DOCTYPE Fogareiros [  
<!ELEMENT Fogareiros ((Motorista)+,(Táxi)+)>  
<!ELEMENT Táxi (AnoT,(Proprietário)+)>           <!ELEMENT Motorista (NomeM)>  
<!ATTLIST Táxi IdT ID #REQUIRED>                 <!ATTLIST Motorista IdM ID #REQUIRED>  
<!ELEMENT AnoT (#PCDATA)>                         <!ELEMENT NomeM (#PCDATA)>  
<!ELEMENT Proprietário EMPTY>  
<!ATTLIST Proprietário ref IDREF #REQUIRED> ]>
```

I.4. Apresente um documento XML que capture o seguinte cenário: existem dois motoristas Zé Carlos e Xico Silva, e os táxis “AA-00-00” e “XX-77-77” ambos de 2007. Sabe-se ainda que o Zé Carlos é proprietário de ambos os táxis e que o Xico Silva também é proprietário do primeiro táxi.

I.5. Apresente uma expressão XPath que devolva os elementos Táxi que têm como proprietário pelo menos o Zé Carlos. Serão valorizadas as respostas que efectuem a procura por nome do motorista em vez de simplesmente por identificador.

GRUPO II

II.1. Sejam as tabelas $R(\underline{a}, b)$, $S(\underline{a}, c)$ e $T(\underline{a}, d)$ resultantes do processo de derivação de esquemas de tabelas para o DER apresentado abaixo. O atributo **a** é do tipo inteiro sendo chave primária de todas as tabelas. Considere ainda o esqueleto de trigger à direita.



```
CREATE TRIGGER s_alterado
BEFORE UPDATE ON S
FOR EACH ROW
WHEN (NEW.a < > OLD.a)
DECLARE x NUMBER;
BEGIN
    /* Código aqui */
END;
```

Qual dos seguintes trechos de código, a incluir no ponto indicado no código anterior, implementa correctamente a condição “disjoint” do DER aquando de alterações de tuplos da tabela S?

(F) <pre>SELECT COUNT(*) INTO x FROM T WHERE T.a = :OLD.a; IF x >= 0 THEN Raise_application_error(-20001,'Erro'); END IF;</pre>	(G) <pre>SELECT COUNT(*) INTO x FROM T WHERE T.a = :OLD.a; IF x > 0 THEN Raise_application_error(-20001,'Erro'); END IF;</pre>
(H) <pre>SELECT COUNT(*) INTO x FROM T WHERE T.a = :NEW.a; IF x >= 0 THEN Raise_application_error(-20001,'Erro'); END IF;</pre>	(I) <pre>SELECT COUNT(*) INTO x FROM T WHERE T.a = :NEW.a; IF x > 0 THEN Raise_application_error(-20001,'Erro'); END IF;</pre>

XX

II.2. Considere as seguintes tabelas:

```
CREATE TABLE R(a INT, b INT, c INT, PRIMARY KEY (a,b,c), UNIQUE (a,b));
CREATE TABLE S(d INT PRIMARY KEY, c INT UNIQUE);
CREATE TABLE T(d INT PRIMARY KEY, a INT, b INT);

ALTER TABLE R ADD FOREIGN KEY (c) REFERENCES S(c);
ALTER TABLE S ADD FOREIGN KEY (d) REFERENCES T;
ALTER TABLE T ADD FOREIGN KEY (a,b) REFERENCES R(a,b);
```

Considere os seguintes scripts:

- I. INSERT INTO T(d) VALUES(4); INSERT INTO S(d,c) VALUES(4,3);
INSERT INTO R VALUES (1,2,3); UPDATE T SET a=1,b=2 WHERE d=4;
- II. INSERT INTO R(a,b,c) VALUES(1,2,NULL); INSERT INTO T(d,a,b) VALUES(4,1,2);
INSERT INTO S(d,c) VALUES (4,3); UPDATE R SET c=3 WHERE a=1 AND b=2;

Quais dos scripts correm sem produzirem erro sabendo que as tabelas se encontram inicialmente vazias?

- (J) Nenhum deles. (K) Apenas I. (L) Apenas II. (M) I e II.

II.3. Considere o esquema de relação R(A, B, C, D) com as seguintes dependências funcionais:

$$A \rightarrow B \quad B \rightarrow A \quad C \rightarrow D \quad D \rightarrow C$$

Quantas chaves candidatas tem o esquema dado?

- (N) 1 (O) 2 (P) 3 (Q) 4

XX

II.4. Considere o esquema de relação R(A, B, C, D, E) com o seguinte conjunto de dependências funcionais na forma canónica:

$$A \rightarrow B \quad C \rightarrow D \quad D \rightarrow CE$$

E os seguintes esquemas de relação:

R3(C,D) R4(A,C) R5(A,C,D)

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

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

XX

II.5. Quais das seguintes frases são verdadeiras?

- I. Toda a relação possui pelo menos uma super-chave.
II. Toda a relação possui pelo menos uma chave candidata.
III. Toda a relação possui pelo menos uma super-chave e uma chave candidata **distintas**.

- (A) Nenhuma (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) Todas

XX

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

$$\begin{array}{cc} A \rightarrow A & B \\ 1^a & 2^a \end{array} \quad BC \rightarrow A \quad 3^a \quad B \rightarrow A \quad 4^a$$

- (I) Nenhuma (K) A 1^a e a 2^a (M) A 1^a e a 3^a (O) A 1^a e a 4^a
(J) A 2^a e a 3^a (L) A 2^a e a 4^a (N) A 3^a e a 4^a (P) Todas

XX

II.7. Considere as seguintes instâncias de relação R1(a,c,d) e R2(a,b) obtidas por decomposição de R(a,b,c,d):

R1			R2	
a	c	d	a	b
1	1	1	1	2
1	1	2	1	3
2	1	1	2	2
2	1	2		

Qual das seguintes dependências não é satisfeita pela instância da relação original R(a,b,c,d)?

- (B) $A \rightarrow C$ (C) $A \twoheadrightarrow C$ (D) $A \rightarrow C D$ (E) $AB \twoheadrightarrow D$

As próximas três perguntas são baseadas no seguinte document XML “Faculdade”:

```
<Faculdade>
  <Cursos>
    <ROW>
      <Nome id="C1">Informática</Nome>
      <Grau>1º ciclo</Grau>
    </ROW>
    <ROW>
      <Nome id="C2">Informática</Nome>
      <Grau>2º ciclo</Grau>
    </ROW>
  </Cursos>
  <Alunos>
    <ROW>
      <Nome id="A1">João Manuel</Nome>
      <Curso ref="C1"/>
    </ROW>
    <ROW>
      <Nome id="A2">Joana Manuela</Nome>
      <Curso ref="C2"/>
    </ROW>
    <ROW>
      <Nome id="A3">Manuel João</Nome>
      <Curso ref="C1"/>
    </ROW>
  </Alunos>
</Faculdade>
```

II.8. Qual das seguintes expressões Xpath produz um conjunto com **mais de 2 nós** quando aplicada ao documento “Faculdade”?

- (M) //ROW[Nome/text() = "Informática"]/Grau
- (N) /Faculdade/Alunos/ROW[Curso]
- (O) /Faculdade/*
- (P) /*/*

II.9. Numa DTD que o documento “Faculdade” satisfaça, qual das seguintes declarações de elemento **não** encontraria certamente?

- (V) <!ELEMENT ROW ANY>
- (W) <!ELEMENT ROW ((Nome | Curso | Grau)*)>
- (X) <!ELEMENT ROW ((Nome, (Grau | Curso))) >
- (Y) <!ELEMENT ROW ((Nome, Grau)?, (Curso)?)>

II.10. Assumindo que “Faculdade” está armazenado no ficheiro Faculdade.xml e executamos a consulta Xquery:

```
for $c in doc("Faculdade.xml")/Faculdade/Cursos/ROW,
  $a in doc("Faculdade.xml")//ROW[Curso/@ref = $c/Nome/@id]
return <XPTO>{ $c/Nome/text() } - { $a/Nome/text() } </XPTO>
```

Quantas vezes a cadeia de caracteres “Informática” aparece no resultado?

- (C) 0
- (D) 2
- (E) 3
- (F) 6

FIM