Bases de Dados, 2018/19
Departamento de Ciência de Computadores, Faculdade de Ciências da Universidade do Porto
Exame de época normal — 22/06/2019 — Duração: 2:30 ■
Número: _____

Nome: _____

GRUPO A (25 %) — Responda na folha do enunciado.

1. Complete o seguinte código para criação de uma tabela `FUNCIONARIO`, com os seguintes atributos: `FuncId` de tipo inteiro para o identificador do funcionário, a chave primária da tabela; `Nome`, uma string de tamanho 32 para o nome; `DepId`, um inteiro indicando o departamento a que o funcionário está associado, chave externa para o atributo com o mesmo nome na tabela `DEPARTAMENTO`; `Supervisor`, o supervisor do funcionário, uma chave externa para `FUNCIONARIO.FuncId`. O atributo `Supervisor` é o único que poderá não estar definido (i.e., pode tomar valor `NULL`),

```
CREATE TABLE FUNCIONARIO
(
  FuncId     _____________________________________________,
  Nome       _____________________________________________,
  DepId      _____________________________________________,
  Supervisor _____________________________________________,
  ____________________________________________________________
  ____________________________________________________________
  ____________________________________________________________
);
```

2. **(Escolha múltipla)** Tenha em conta as tabelas `FUNCIONARIO` e `DEPARTAMENTO` da questão 1 e a instrução `DELETE FROM DEPARTAMENTO WHERE DepId = 1`. Seja d_1 o registo do departamento 1 e F_1 o conjunto de registos em `FUNCIONARIO` que se referem ao departamento 1.

☐ Além de d_1, os registos em F_1 são também removidos se a chave externa `DepId` em `FUNCIONARIO` fôr adicionalmente configurada com a opção `ON DELETE CASCADE`.
☐ Além de d_1, os registos em F_1 são também removidos se a chave primária `DepId` em `DEPARTAMENTO` fôr adicionalmente configurada com a opção `ON DELETE CASCADE`.
☐ d_1 só pode ser removido se alterarmos o esquema tal que `FUNCIONARIO.DepId` deixe de ser uma chave externa.
☐ Além de d_1, os registos em F_1 são também removidos em qualquer caso.
☐ Se o esquema não sofrer alterações, d_1 só pode ser removido se $F_1 = \emptyset$ (não houverem funcionários associados ao departamento 1).

3. **(Escolha múltipla)** Considere o seguinte esqueleto para um "trigger":

```
CREATE TRIGGER beforeFuncUpdate
BEFORE UPDATE ON FUNCIONARIO FOR EACH ROW
BEGIN ... END
```

Assuma que o "trigger" está definido e que executamos uma instrução `UPDATE` sobre a tabela `FUNCIONARIO`.

☐ Podemos ter instruções no corpo do "trigger" para abortar a instrução `UPDATE`.
☐ O "trigger" é executado apenas para os registos da tabela `FUNCIONARIO` que são manipulados pela instrução `UPDATE`.
☐ O "trigger" só executa se usarmos uma instrução `CALL` para o efeito antes da instrução `UPDATE`.
☐ O "trigger" executa de forma automática antes da instrução `UPDATE` ter efeito.
☐ O "trigger" é executado para todos os registos da tabela `FUNCIONARIO`.
☐ A instrução `UPDATE` tem sempre efeito de forma independente da execução do "trigger".

Número: _____

Nome: _____

4. (Escolha múltipla) Considere a seguinte consulta SQL, onde, além da caracterização da tabela `DEPARTAMENTO` e `FUNCIONARIO` na questão 1, deve assumir que `Gestor` é um atributo da tabela `DEPARTAMENTO` identificando o funcionário gestor de um departamento (chave externa para `FUNCIONARIO.FuncId`).

```
SELECT Gestor, COUNT(*) AS N
FROM DEPARTAMENTO JOIN FUNCIONARIO USING(DepId)
WHERE Supervisor = Gestor
GROUP BY Gestor
HAVING N >= 10;
```

Assinale as consultas expressas em álgebra relacional abaixo que são equivalentes à consulta SQL, onde F e D são abreviaturas paea `FUNCIONARIO` e `DEPARTAMENTO` respectivamente.

☐ $R_1 \longleftarrow \sigma_{\text{Supervisor = Gestor}}(F)$
$R_2 \longleftarrow D \bowtie_{\text{DepId}} R_1$
$R_3 \longleftarrow {}_{\text{Gestor}}\mathcal{F}_{\text{COUNT}(*)}(R_2)$
$R_4 \longleftarrow \rho_{\text{Gestor,N}}(R_3)$
$R_5 \longleftarrow \sigma_{\text{N}\geq 10}(R_4)$

☐ $R_1 \longleftarrow D \bowtie_{\text{DepId}} F$
$R_2 \longleftarrow \sigma_{\text{Supervisor = Gestor}}(R_1)$
$R_3 \longleftarrow {}_{\text{Gestor}}\mathcal{F}_{\text{COUNT}(*)}(R_2)$
$R_4 \longleftarrow \rho_{\text{Gestor,N}}(R_3)$
$R_5 \longleftarrow \sigma_{\text{N}\geq 10}(R_4)$

☐ $R_1 \longleftarrow D \bowtie_{\text{DepId}} F$
$R_2 \longleftarrow {}_{\text{DepId}}\mathcal{F}_{\text{COUNT}(*)}(R_1)$
$R_3 \longleftarrow \rho_{\text{DepId,N}}(R_2)$
$R_4 \longleftarrow \sigma_{\text{N}\geq 10}(R_3)$
$R_5 \longleftarrow \sigma_{\text{Supervisor = Gestor}}(R_4)$

☐ $R_1 \longleftarrow D \times F$
$R_2 \longleftarrow \sigma_{\text{Supervisor = Gestor} \wedge F.\text{DepId}=D.\text{DepId}}(R_1)$
$R_3 \longleftarrow {}_{\text{Gestor}}\mathcal{F}_{\text{COUNT}(*)}(R_2)$
$R_4 \longleftarrow \rho_{\text{Gestor,N}}(R_3)$
$R_5 \longleftarrow \sigma_{\text{N}\geq 10}(R_4)$

5. (Escolha múltipla) Considere o seguinte código de uma aplicação Python que acede a uma base de dados com a tabela `FUNCIONARIO`:

```
data = db.execute(
  'SELECT * FROM FUNCIONARIO WHERE Name LIKE \'%' + expr + '%\''
  ).fetchall()
```

Suponha que `expr` é um valor obtido de uma fonte externa à aplicação (por ex. através da rede), e que esse valor não é alvo de qualquer validação.

☐ A possibilidade de injeção SQL não é neste caso problemática, já que está em causa apenas a execução de uma instrução `SELECT` para leitura de dados.
☐ A possibilidade de injeção SQL depende exclusivamente do código SQL construído por uma aplicação, e não do SGBD.
☐ O código é passível de injeção SQL, podendo portanto levar à execução de código SQL indevido e potencialmente malicioso.
☐ Não há possibilidade de injeção SQL neste caso.
☐ Um SGBD moderno tipicamente detecta situações de injeção SQL.

GRUPO B (25 %) — Use folhas de exame.

Considere os seguintes requisitos para uma base de dados de uma empresa de táxis:

— Um taxista é caracterizado por um número de carta de condução única, data de validade da carta de condução, nome, data de nascimento, e um contacto telefónico.
— Uma praça de táxis tem associada um identificador único, e uma localização expressa em termos do nome de uma rua e do nome de uma localidade.
— Um táxi é caracterizado por uma matrícula única, a marca do carro, o modelo do carro, e ano.
— Uma corrida de táxi é identificada por um número, data, hora, distância, e preço cobrado. O número de uma corrida é único por táxi, mas pode-se repetir-se para táxis diferentes.
— Um táxi está associado a uma única praça de táxis. Vários táxis podem estar associados à mesma praça, existindo pelo menos um táxi por praça.
— Um táxi pode ser conduzido por taxistas diferentes em turnos distintos. Analogamente, um taxista pode conduzir diferentes táxis em diferentes turnos. Um turno de táxi é caracterizado por um dia da semana e alturas do dia distintas: manhã, tarde e noite. Temporariamente, um taxista pode não ter serviço atribuído e um táxi pode não estar ao serviço.

1. Apresente um modelo ER na forma de um diagrama para a base de dados.

2. Apresente uma tradução do modelo ER para um esquema relacional.

GRUPO C (15 %) — Use folhas de exame.

Considere uma BD para uma empresa de organização de concertos de música em salas de espetáculos. Um concerto de música:

— é caracterizado por um identificador único, nome do artista ou grupo, data e hora;
— tem lugar numa única sala de espetáculos, em que cada sala é caracterizada por um identificador único, um nome, e um número de lugares;
— pode ter associado vários funcionários da empresa em que cada funcionário tem associado um identificador único, um nome, e um papel na organização do concerto que pode ser distinto para concertos diferentes (veja por ex. as entradas envolvendo `Sérgio Abreu` abaixo).

Assuma o seguinte esquema 1NF para a BD e exemplos de entradas correspondentes na BD:

`CONCERTO(CId,CArt,CData,CHora,SId,SNome,SLugares,FId,FNome,FPapel)`

CId	CArt	CData	CHora	SId	SNome	SLugares	FId	FNome	FPapel
1	Caetano Veloso	21-06-2019	22:00	1	Coliseu	3000	1	Sérgio Abreu	Promotor
1	Caetano Veloso	21-06-2019	22:00	1	Coliseu	3000	2	Maria Menezes	Relações Públicas
1	Caetano Veloso	21-06-2019	22:00	1	Coliseu	3000	3	Carlos Roberto	Técnico de Som
2	Antónia Variations	25-06-2019	23:30	2	Maus Hábitos	300	1	Sérgio Abreu	Relações Públicas
2	Antónia Variations	25-06-2019	23:30	2	Maus Hábitos	300	4	Filipe Marques	Promotor
2	Antónia Variations	25-06-2019	23:30	2	Maus Hábitos	300	3	Carlos Roberto	Técnico de Som
3	Iggy Carvalho	26-06-2019	2:30	2	Maus Hábitos	300	4	Filipe Marques	Promotor
3	Iggy Carvalho	26-06-2019	2:30	2	Maus Hábitos	300	3	Carlos Roberto	Técnico de Som

1. Identifique as dependências funcionais entre atributos em `CONCERTO` e o conjunto de atributos que formam a chave primária.

2. Explique porque é que o esquema não é 2NF, e normalize-o para 2NF.

3. Explique se o esquema que obteve na questão anterior é ou não também 3NF. Se não for o caso, normalize-o para 3NF.

GRUPO D (15 %) — Use folhas de exame.

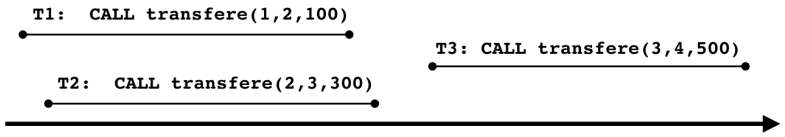
Considere uma tabela `CONTA` para uma conta bancária com atributos `Num` para o número da conta e `Saldo` para o saldo da conta, ambos inteiros. Suponha ainda que está definido um "stored procedure" `transfere(IN c1 INT, IN c2 INT, IN v INT)` de tal forma que `transfere(c1,c2,v)` executa a seguinte sequência de passos:

— Inicia uma transação com `START TRANSACTION`.
— Se `c1` e `c2` forem números de contas válidas e o saldo da conta `c1` for igual ou superior a `v` (código omitido) então é executado:

```
UPDATE CONTA SET Saldo = Saldo - v WHERE Num = c1;
UPDATE CONTA SET Saldo = Saldo + v WHERE Num = c2;
COMMIT;
```

isto é, o saldo de `c1` é decrementado (debitado) no montante de `v`, o de `c2` é incrementado (creditado) no montante de `v`, e a transação toma efeito permanente com `COMMIT`.
— Caso contrário, a transação iniciada é abortada com `ROLLBACK`.

1. Considere que as contas 1, 2, 3, 4 estão definidas na BD com saldos de respectivamente, 100, 200, 300, e 400. Na execução de 3 transações da forma ilustrada abaixo, temos que `T1` e `T2` executam em simultâneo, e que `T3` executa apenas após `T1` e `T2` terem terminado. Assumindo que o SGDB executa as transações de forma serializável, justifique quais são os estados possíveis da BD no final para a tabela `CONTA`, e que transações completam com sucesso ou são abortadas para cada possibilidade.



2. Ilustre um escalonamento possível para uma execução serializável de `T1` e `T2`. Explique o escalonamento passo-a-passo em termos de leituras e escritas de registos, uso de "locks" segundo uma estratégia de "two-phase locking", e de tal forma que uma das transações `T1` ou `T2` bloqueie momentaneamente.

GRUPO E (20 %) — Use uma folha de exame SEPARADA para este grupo.

x	2	$]2,4]$	$]4,8]$	$]8,16]$	$]16,32]$	$]32,64]$	$]64,128]$	$]128,256]$	$]256,512]$	$]512,1024]$	$]1024,2048]$	$]2048,4096]$
$\lceil \log_2(x) \rceil$	1	2	3	4	5	6	7	8	9	10	11	12

Neste grupo justifique as suas respostas apresentando todos os cálculos relevantes que efectuou. Assuma em todos os casos que um registo está inteiramente guardado num único bloco de disco (i.e., um registo não "atravessa" blocos), que cada bloco de disco tem um tamanho de 4096 (2^{12}) bytes, e que uma referência a um bloco de disco tem um tamanho de 8 bytes.

1. Considere uma tabela `T` com $5,000$ registos de tamanho fixo igual a 200 bytes, armazenada num ficheiro ordenado pela chave primária `K` de `T`, onde o tamanho de `K` é de 8 bytes. Indique o número de blocos em disco necessários para o armazenamento de `T`, e o número máximo de acessos a blocos de disco para localizar um registo com base num valor para `K`.

2. Se criarmos um índice primário sobre o ficheiro para a tabela `T`, quantos blocos em disco ocupará o índice? Usando o índice, qual será o número máximo de acessos a blocos de disco para localizar um registo com base num valor para `K` ?

3. Considere agora que `T` tem também um atributo `U` que é uma chave secundária (i.e., `UNIQUE`) com um tamanho de 8 bytes. Indique quantos blocos em disco teremos de aceder sem recorrer a qualquer indexação para localizar um registo com base num valor para `U`. Caracterize de seguida um índice multi-nível para `U` de tal forma que o índice de último nível ocupe apenas 1 bloco em disco. Para cada nível i, identifique o número de blocos em disco necessários, e o número máximo de acessos a disco para localizar um registo a partir do nível i.