

Bases de Dados (CC2005), 2019/20, DCC/FCUP

Exame de época normal — 06/07/2020 — Duração: 2:00 ★

Número:  Nome:

**GRUPO A (32.5 %) — Responda na folha do enunciado.**

**A cotação individual deste grupo pode substituir a nota do teste SQL (caso seja melhor).**

**Em qualquer caso é obrigatória a sua resolução (contará para a nota global do exame).**

1. Complete o seguinte fragmento SQL de 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. Tenha em conta as tabelas `FUNCIONARIO` e `DEPARTAMENTO` referidas na questão 1. Escreva uma consulta SQL para obter o identificador de departamentos (`DEPARTAMENTO.DepId`) que têm 5 ou mais funcionários sem supervisor e o correspondente nº de funcionários em causa. Os resultados devem estar ordenados de forma decrescente pelo nº de funcionários sem supervisor e depois pelo identificador do departamento (para igual nº de funcionários sem supervisor).

3. Expresse a consulta anterior através de uma sequência equivalente de operações de álgebra relacional (AR). Não precisa de ter em conta o critério de ordenação (que não pode ser expresso em AR).

Notação para alguns operadores AR –  $\sigma$ : seleção;  $\pi$ : projeção;  $\mathcal{F}$ : agregação;  $\bowtie$ : junção;  $*$ : junção natural.

4. [escolha múltipla] Tenha em conta novamente as tabelas FUNCIONARIO DEPARTAMENTO da questão 1. Considere a execução da instrução DELETE FROM DEPARTAMENTO WHERE DepId = 1, sujeita à definição de ON DELETE CASCADE para FUNCIONARIO.DepId ou não. Seja  $d_1$  o registo do departamento 1 e  $F_1$  o conjunto de registos em FUNCIONARIO que se referem ao departamento 1 via FUNCIONARIO.DepId. Que registos são removidos pela instrução DELETE ?

( $\emptyset$  denota o conjunto vazio)

- Apenas  $d_1$ , se  $F_1 = \emptyset$  independentemente da ON DELETE CASCADE estar definido ou não.
- $d_1$  e todos os registos em  $F_1$ , se ON DELETE CASCADE não está definido.
- $d_1$  e todos os registos em  $F_1$ , em qualquer caso.
- $d_1$  e todos os registos em  $F_1$ , se ON DELETE CASCADE está definido.
- Apenas  $d_1$ , se  $F_1 \neq \emptyset$  e ON DELETE CASCADE não está definido.
- Nenhum, se  $F_1 \neq \emptyset$  e ON DELETE CASCADE não está definido.

5. [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.

- O “trigger” só executa se usarmos uma instrução CALL para o efeito antes da instrução UPDATE.
- 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 modificados pela instrução UPDATE.
- O “trigger” é executado para todos os registos da tabela FUNCIONARIO.
- A instrução UPDATE tem sempre efeito de forma independente da execução do “trigger”.
- O “trigger” executa de forma automática antes da instrução UPDATE ter efeito.

6. [escolha múltipla] Considere os fragmentos de código (1) e (2) de uma aplicação Python que acede a uma base de dados com a tabela FUNCIONARIO:

```
# (1)
dados = db.execute(
    'SELECT * FROM FUNCIONARIO WHERE Nome = ' + nome + '
').fetchall()
...
# (2)
dados = db.execute(
    'SELECT * FROM FUNCIONARIO WHERE Nome = %s',
    nome
).fetchall()
```

Suponha que nome é um valor arbitrário obtido de uma fonte externa à aplicação (ex. através de uma ligação à rede).

- O código em (1) é passível de injeção SQL.
- O código em (2) é passível de injeção SQL.
- O código em (1) usa uma instrução parameterizada que inibe injeção de SQL.
- O código em (2) usa uma instrução parameterizada que inibe injeção de SQL.
- Uma instrução parameterizada não deixa que código SQL arbitrário seja injectado dinamicamente a partir de valores “maliciosos” passados à aplicação.
- Uma instrução parameterizada altera o esquema da BD dinamicamente por forma a inibir a injeção de SQL.

**Bases de Dados (CC2005), 2019/20, DCC/FCUP**

**Exame de época normal — 06/07/2020 — Duração: 2:00 ★**

Número:  Nome:

**GRUPO B (12.5 %) — Responda no enunciado.**

Considere uma tabela `ACCOUNT` para uma conta bancária com atributos inteiros `Num` para o número da conta e `Value` para o saldo da conta, e que está definido um “stored procedure”

```
transfer(IN a INT, IN b INT, IN v INT)
```

de tal forma que `transfer(a, b, v)` executa internamente a seguinte sequência de passos:

— Inicia uma transação com `START TRANSACTION`.

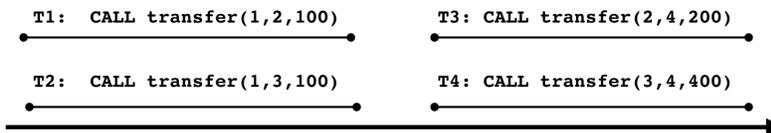
— Se `a` e `b` forem números de contas válidas e o saldo da conta `a` for igual ou superior a `v` (o código é omitido para o teste destas condições) então são executadas as seguintes instruções:

```
UPDATE ACCOUNT SET Value = Value - v WHERE Num = a;  
UPDATE ACCOUNT SET Value = Value + v WHERE Num = b;  
COMMIT;
```

isto é, o saldo de `a` é decrementado (debitado) no montante de `v`, o de `b` é incrementado (creditado) no montante de `v`, e a transação toma efeito permanente com `COMMIT`.

— Caso contrário, a transação iniciada é abortada com a execução de uma instrução `ROLLBACK`.

1. [escolha múltipla] Considere que as contas 1, 2, 3, e 4 estão definidas na BD com saldos iniciais de, respectivamente, 100, 200, 300, e 400. Na execução de 4 transações da forma ilustrada abaixo, temos que T1 e T2 executam em simultâneo, seguidas de T3 em simultâneo com T4.



Quais das seguintes opções são possíveis tendo em conta execuções serializáveis das transações, em termos de transações que completam com sucesso (i.e., terminam com `COMMIT`) e correspondentes estado da BD no para os saldos finais das contas 1 a 4?

	Transações completadas	Conta 1	Conta 2	Conta 3	Conta 4
<input type="checkbox"/>	Todas	0	100	300	600
<input type="checkbox"/>	Todas menos T1	0	0	0	1000
<input type="checkbox"/>	Todas	-100	100	0	1000
<input type="checkbox"/>	T1 e T3	0	100	300	600
<input type="checkbox"/>	Todas menos T2	0	0	0	1000
<input type="checkbox"/>	Apenas T3	100	0	300	600

2. [escolha múltipla] Em protocolos “two-phase locking” para execução de transações concorrentes de bases de dados:

- Cada transação passa por uma fase de “expansão”, durante a qual só adquire “locks”, e depois por outra de “contração”, durante a qual só liberta “locks”.
- Cada transação alterna entre fases de “expansão”, durante as quais só adquire “locks”, e “contração”, durante as quais só liberta “locks”.
- É garantida a ausência de “deadlocks”.
- Podem ocorrer “deadlocks” durante a execução.
- É garantida uma execução serializável de transações concorrentes.
- É apenas garantida a ausência de “leituras sujas” (“dirty reads”) durante a execução de transações concorrentes.

3. [escolha múltipla] Durante a execução de uma transação, uma anomalia de “leitura suja” (“dirty read”):

- É caracterizada pela leitura de dois valores diferentes para o mesmo registo durante a execução, em virtude do valor escrito por outra transação que terá completado entre as duas leituras.
- É caracterizada pela leitura de um valor escrito por outra transação ainda não completada e que acaba por abortar.
- É caracterizada pela leitura de um valor de uma tabela cujo acesso se encontra bloqueado por um “lock”.
- Uma execução serializável da transação e outras simultaneamente activas pode ser comprometida.
- Uma execução serializável da transação e outras simultaneamente activas nunca é comprometida.

**GRUPO C (10 %) — Responda no enunciado.**

$x$	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.**

Considere uma tabela  $T$  com  $N = 16,000$  registos de tamanho fixo igual a  $R = 400$  bytes, armazenada num ficheiro ordenado pela chave primária  $K$  de  $T$ , onde o tamanho tomado por um valor de  $K$  é de 24 bytes. Assuma ainda que cada bloco de disco tem um tamanho de  $B = 4096$  ( $2^{12}$ ) bytes, e que uma referência a um bloco de disco tem um tamanho de 8 bytes.

1. 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  $K$  ?

**Bases de Dados (CC2005), 2019/20, DCC/FCUP**

**Exame de época normal — 06/07/2020 — Duração: 2:00 ★**

Número:  Nome:

**GRUPO D (32.5 %) — Responda em folha de exame.**

Considere os seguintes requisitos para uma base de dados de um serviço de partilha de documentos “online” (ex. análogo a Dropbox):

- Um utilizador tem um identificador de login único, um nome, e um endereço de email.
- Um documento tem um identificador único, um nome, um conjunto de “hash-tags”, e também um tipo que pode ser ou “pasta” ou “ficheiro”.
- Um documento (ficheiro ou pasta) pode ser parte de um documento do tipo pasta. Uma pasta pode conter vários outros documentos, mas também estar vazia.
- Um documento é pertença de apenas um utilizador, mas pode ser partilhado por esse utilizador com outros. Por cada partilha com um utilizador é identificado o nível de acesso dado (ex. escrita ou apenas leitura).
- Um documento tem várias versões, em que cada versão tem um número de ordem que é apenas único por documento, e ainda associado um tamanho e data de criação.
- Uma versão de um documento é criada por um utilizador que pode ser o dono do documento ou outro utilizador que tenha acesso ao documento via partilha.
- Podem haver utilizadores sem qualquer documento criado ou acesso a documento partilhado, e também documentos que não são partilhados.

1. Apresente um modelo ER na forma de um diagrama para a base de dados. **Nota:** Não deverá usar entidades-tipo distintas para distinguir documentos do tipo “pasta” ou “ficheiro”; use apenas uma entidade-tipo para documento com um atributo para o tipo de documento.
2. Apresente um esquema relacional correspondente à tradução do modelo ER.

**GRUPO E (12.5 %) — Responda em folha de exame.**

Considere uma base de dados para livros, em que cada livro é caracterizado por:

- um código ISBN único (ISBN), um título (Título), ano de edição (Ano);
- um ou mais autores, em que cada autor tem um identificador único (AId) e um nome (ANome);
- e uma editora, em que cada editora tem um identificador único (EId) e um nome (ENome).

Considere o seguinte esquema INF para a BD e exemplos de entradas correspondentes na BD:

LIVRO (ISBN, Título, Ano, AId, ANome, EId, ENome)

ISBN	Título	Ano	AId	ANome	EId	ENome
9789722709620	Os Lusíadas	2009	1	Luís de Camões	1	Porto Editora
9789722526289	Sonetos	2009	1	Luís de Camões	2	Edições Asa
9789720049773	Sonetos	2019	2	Florbela Espanca	1	Porto Editora
9789724138695	Astérix o Gaulês	2019	3	René Goscinny	2	Edições Asa
9789724138695	Astérix o Gaulês	2019	4	Albert Uderzo	2	Edições Asa
9789724146911	Lucky Luke - O Tenrinho	2006	3	René Goscinny	2	Edições Asa
9789724146911	Lucky Luke - O Tenrinho	2006	5	Morris	2	Edições Asa

1. Identifique as dependências funcionais entre atributos em LIVRO 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.