

DI-FCT-NOVA

9 de maio de 2018

Bases de Dados

2º teste, 2017/18 (versão A)

Duração: 1,5 horas (consulta limitada)

Considere uma base de dados relacional para armazenar informação sobre serviços de telecomunicações da empresa FCTel, criada pelos seguintes comandos em SQL no dialecto do sistema estudado nas práticas:

<pre> create table Pessoas(idPessoa number primary key, nome varchar(100), dataNasc date, idMorada number not null, foreign key (idMorada) references Moradas); create table Contratos(idContrato number primary key, dataInicio date not null, dataFim date, idPessoa references Pessoas, idFunc references Pessoas); create table Serviços(idServ number primary key, nome varchar(50), preço number not null, check (preço >= 0)); create table ServiçosInternet(idServ number primary key, download integer, upload integer, foreign key (idServ) references Serviços); </pre>	<pre> create table Moradas(idMorada number primary key, rua varchar(100) not null, numandar varchar(20) not null, localidade varchar(100) not null); create table Assinaturas(idContrato number, idServ number, idMorada number, primary key (idContrato, idServ), foreign key (idContrato) references Contratos, foreign key (idServ) references Serviços, foreign key (idMorada) references Moradas); create table ServiçosTV(idServ number primary key, canais integer, idServINT integer not null, foreign key (idServ) references Serviços, foreign key (idServINT) references ServiçosInternet); </pre>
---	---

A base de dados tem: uma tabela de pessoas, com um identificador único, para armazenar clientes e/ou funcionários, tendo ainda uma chave estrangeira para a sua morada representada na tabela Moradas; uma tabela de contratos com os identificadores do cliente e do funcionário que o realizou, assim como a data de início do contrato e sua data de fim (pode ser NULL quando é por tempo ilimitado); uma tabela de assinaturas com os serviços assinados com o contrato e seu local de instalação (pode ser NULL quando não foi instalado ou não necessita de instalação física); uma tabela de serviços com o seu identificador, nome e preço mensal assim como as suas tabelas de especialização para serviços Internet e de TV (não esquecer que os serviços TV têm sempre um serviço Internet associado).

NOTA: Para encurtar a escrita das restrições de integridade referencial utiliza-se a notação do Oracle em que por omissão é referida a chave primária da tabela pai. Por exemplo, “**foreign key** (idServINT) **references** ServiçosInternet” é equivalente a “**foreign key** (idServINT) **references** ServiçosInternet(idServ)” e “idPessoa **references** Pessoas” é equivalente a “idPessoa **references** Pessoas(idPessoa)”.

Página propositadamente deixada em branco (rascunho)

Grupo 1

1. Apresente uma expressão em **álgebra relacional e uma consulta em SQL** para cada uma das perguntas:
 - a) **[3 valores (1.5 álgebra+1.5 SQL)]** Quais os nomes dos clientes de Almada que assinaram serviços por mais de 50€/mês?
 - b) **[3]** Quais os contratos que assinaram todos os serviços disponíveis ?
 - c) **[3]** Quais os funcionários que celebraram contratos que incluam serviços que não sejam nem de Internet nem de TV?
2. Apresente **uma consulta SQL** para cada uma das perguntas:
 - a) **[2]** Qual o montante mensal a pagar por cada contrato?
 - b) **[2]** Qual é o serviço mais rentável para a empresa FCTel (pode existir mais do que um) ?

3. **[2]** Considere a seguinte *view* e a seguinte consulta SQL:

```
create view A as
( select idPessoa, nome
  from Contratos natural inner join Pessoas
 group by idPessoa, nome
 having count(dataInicio) <> count(dataFim)
 );
```

Diga qual é, intuitivamente, o resultado da *view* e se devolve alguns resultados ou não. Indique se podem ser devolvidos tuplos repetidos, justificando a sua resposta.

Grupo 2

1. Como já reparou, a base de dados acima tem uma série de problemas de desenho e omissão de algumas restrições. Apresente o código SQL (de alteração de tabelas e/ou introdução de novas restrições e/ou introdução de asserções) para impor cada uma das seguintes restrições de integridade:
 - a) **[1]** Não existem serviços com o mesmo nome e mesmo preço.
 - b) **[1]** Um funcionário não pode celebrar um contrato consigo mesmo.
 - c) **[1]** Não pode haver contratos com serviço de TV assinado sem o respetivo serviço de Internet também assinado.
2. **[2]** A tabela PacoteServiços (ver comando de criação abaixo) regista serviços “virtuais” que agregam outros serviços, com a indicação do desconto sobre o preço base do serviço incluído. No máximo um dos serviços pode ser oferecido gratuitamente, colocando o valor de 100 na coluna desconto. Não há pacotes de pacotes e um cliente não pode assinar um pacote e um serviço que esteja incluído nesse pacote.

```
create table PacoteServiços(
  idServPac number,
  idServInc number,
  desconto number not null default 10.0,
  primary key (idServPac, idServInc)
  foreign key (idServPac) references Serviços,
  foreign key (idServInc) references Serviços
);
```

Crie *trigger(s)* garantindo que na inserção/atualização de linhas na tabela PacoteServiços:

- a) ao incluir-se um serviço de TV no pacote também se inclui no mesmo pacote (caso não esteja já incluído) o serviço de Internet que é oferecido (gratuito).
- b) quando se tenta inserir um novo serviço gratuito já existindo um outro, o desconto a aplicar é de 10% ao novo a que se está inserir/alterar (esta regra tem prioridade sobre a anterior);
- c) Um cliente não assina simultaneamente um pacote e um serviço lá incluído.

FIM