

Bases de Dados II

Ficha de Exercícios nº7

Engenharia Informática

Docentes

Casimiro Batista

José Casimiro Pereira

Objectivos

Resolução de exercícios envolvendo procedimentos, funções, declarações pré-preparadas e eventos.

Exercícios

1. Crie uma função, que dada uma determinada cadeia de caracteres, devolva qual o carácter mais frequente e o seu número de ocorrências. (para simplificação considere apenas caracteres minúsculos e em caso de igualdade devolve considerando a ordenação alfabética)
Exemplos:
o João foi à praia --> “o, 4 ocorrências”
bases de dados --> “d, 3 ocorrências”
2. Crie uma função que dado um valor numérico inteiro (um atributo de uma tabela, por exemplo), devolva o seu equivalente binário.
3. Utilizando a base de dados MUSICA, crie o procedimento que permita preencher a seguinte tabela. De notar que a tabela deve ser criada previamente.

Artista	Número de álbuns	Duração média por faixa	Duração média por álbum	Álbum mais longo
Os Os	3	196	2455	Isto demora a acabar
100 Nome	5	200	1648	Fomos!

Nota: os dados exemplificados não existem na BD.

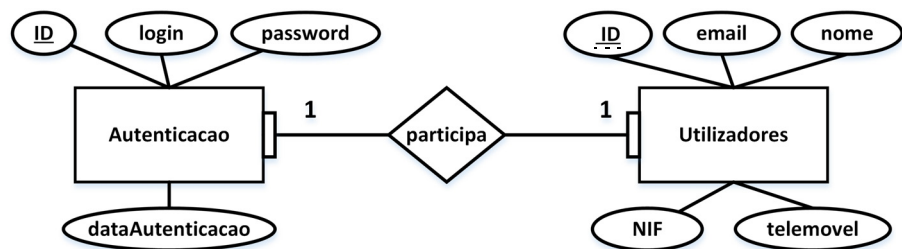
4. Utilizando a base de dados Veterinarios, crie o procedimento **apagaconsultas**, que permita apagar consultas da tabela *tbl_consultas*, dado um valor de *idconsulta*. Como a tabela *tbl_consultas* é referenciada por chaves forasteiras de outras tabelas, haverá casos em que poderá ocorrer um erro. Nesse caso, não se efectuará o apagamento, obviamente, mas deverá ser registado na tabela *tbl_error_log* o seguinte:
 - o código de erro
 - a mensagem: "Em <tempo corrente> não foi possível eliminar a consulta nº <idconsulta> devido a violação de integridade referencial"

5. Considere as duas tabelas apresentadas, pertencentes a duas empresas concorrentes da área do comércio de retalho. Crie um procedimento que crie uma nova tabela (*Produtos*) na base de dados **db3**, e copie os dados das duas tabelas para essa nova tabela. Devido ao facto das duas empresas venderem alguns produtos iguais, é natural aparecerem referências aos mesmos produtos em ambas as tabelas originais. Na operação de cópia, o procedimento deve copiar os dados de forma a criar registos unívocos. Sempre que for detectado um produto repetido, o preço a armazenar deverá ser o mais baixo.

```
CREATE TABLE db1.Produtos (
    idProduto int not null
        primary key,
    nome varchar(45) not null,
    descricao varchar (20),
    preco double,
    codBarras char(13) not null
) ENGINE=MyISAM;
```

```
CREATE TABLE db2.Produts (
    ProductID int not null,
    name varchar(35) not null,
    price int(10) unsigned,
    unitPrice varchar(5),
    BarCode varchar(20),
    PRIMARY KEY (EmployeeID)
) ENGINE=InnoDB;
```

6. Considere a tabela de autenticação, expressa na imagem seguinte:



- a. Crie um *procedimento* que quando alguém cria uma conta neste sistema, atribua automaticamente os respetivos dados às duas tabelas.
- O ID atribuído a um registo na tabela da ‘Autenticacao’ é igual ao atribuído ao registo correspondente na tabela ‘Utilizadores’.
 - O ID é obtido de forma sequencial, devendo ser primeiro pesquisado o último ID introduzido.
 - O *email* do utilizador é criado juntando ao ‘login’ o domínio ‘email.pt’.
 - O valor do atributo *dataAtualizacao* corresponde ao valor da hora, e data, atuais do sistema operativo.

Todos os atributos são de preenchimento obrigatório. Se algum não for fornecido, a operação de criação de conta deverá ser abortada.

- b. Crie um, ou mais, *triggers*, ou outras estruturas que considerar necessárias, que sempre que os dados dos utilizadores são atualizados efetuem um *backup* desses dados. Para tal considere uma tabela denominada ‘backupUtilizadores’, que além da estrutura das duas tabelas dos utilizadores, contém o ‘nome da tabela’ de origem dos dados, e outro atributo onde será registado o tipo de operação que deu origem à escrita dos dados (*update* ou *delete*). Não se esqueça que quando se apaga um registo numa das tabelas, o registo correspondente na outra tabela também deverá ser apagado.

7. Considere a seguinte tabela:

```
create table pessoa
(id int primary key,
 nome varchar(255),
 morada varchar(255),
 data_nasc date);
```

- a.** Utilizando uma declaração pré-preparada, prepare um comando que na tabela anterior devolva o número de pessoas que nasceram num determinado ano. O utilizador fornecerá o ano para o qual quer conhecer o resultado. Exemplifique como executaria a declaração.
- b.** Utilizando uma declaração pré-preparada, prepare um comando que na tabela anterior devolva o número de pessoas que nasceram em cada ano. O utilizador fornecerá o ano a partir do qual pretende o cálculo. Exemplifique como executaria a declaração.
- c.** Crie um evento que, com frequência diária, elimine registos cujo valor de data de nascimento seja superior à data do momento.
- d.** Crie um evento que, com frequência mensal, elimine os registos cuja data de nascimento tenha ocorrido há mais de 20 anos.
- e.** Crie um procedimento que analise qual o número de caracteres existentes nos registos da tabela, para o atributo *morada*. Com base nessa informação, o procedimento deverá alterar o tamanho do atributo *morada* para o maior valor registado, acrescido de mais 5 caracteres. Obviamente, os limites não deverão ser excedidos.