

**EIC0023 - Base de Dados**

## **Gestão de um Jardim Zoológico**

**Mestrado Integrado em Engenharia Informática e Computação**

**Prof:** Carla Alexandra Teixeira Lopes

**Porto, 01 de Junho de 2014**

**Elementos do Grupo:**

Joana Lopes Beleza – ei12029@fe.up.pt

Mário André Ferreira Macedo – ei12105@fe.up.pt

Pedro Miguel Martins de Lemos da Cunha Faria – ei12097@fe.up.pt

Como nos foi solicitado, para esta terceira e última entrega do Projecto de Base de Dados, cujo tema é “Gestão de um Jardim Zoológico”, apresentamos dez interrogações e dois *triggers* plenamente funcionais e relevantes no contexto do trabalho.

Para além do pedido para esta entrega, aproveitamos também para corrigir alguns erros em termos da criação da base de dados e fizemos uma ligeira alteração do *UML* de forma a otimizá-lo. Esta alteração passou por adicionar uma nova tabela “Jaula” que se associa a um “Animal” e também a “TipoDeJaula” e “Localização”, como se pode ver pelo ficheiro em anexo com o novo *UML*. Em anexo encontram-se também todos os *scripts* prontos a correr em *sqlite*, nomeadamente estruturação da base de dados e inserção de dados nesta, interrogações e *triggers*, como se pode ver de seguida:

## INTERROGAÇÕES

1. Listagem de todos animais, focando as informações mais importantes acerca destes, espécie, género e tratador responsável:

```
Select nome, nomeEspecie, genero, DataDeNascimento, Peso, nomeTratador
FROM Animal, Especie, Genero, Tratador
Where (Animal.idEspecie = Especie.idEspecie AND Animal.idGenero = Genero.idGenero AND
Animal.idTratador = Tratador.idTratador)
GROUP BY Animal.nome
ORDER BY Animal.nome;
```

2. Listar os animais de uma determinada espécie, sendo que esta interrogação ser útil em caso de se ter de realizar tratamentos/cuidados a todos os animais dessa espécie. Um exemplo concreto desta situação é a mudança de pele das cobras que ocorre em determinadas alturas do ano e é um fenómeno de toda a espécie:

```
Select nomeEspecie, Animal.nome
From Animal, Especie
Where Especie.idEspecie = Animal.idEspecie
ORDER BY Especie.nomeEspecie;
```

3. Listagem de animais por jaula. Apresenta a sua utilidade em situações como as dos peixes, visto que podemos ter diferentes espécies de peixes no mesmo aquário:

```
Select Animal.idJaula, nomeEspecie
From Jaula, Animal, Especie
Where Animal.idJaula = Jaula.idJaula AND Animal.idEspecie = Especie.idEspecie
ORDER BY Animal.idEspecie;
```

4. Listagem de espécies de animais por zona (Norte, Sul, Este ou Oeste):

```
Select localizacao, nomeEspecie
FROM Localizacao, Especie, Animal, Jaula
Where (Especie.idEspecie = Animal.idEspecie AND Jaula.idLocalizacao =
Localizacao.idLocalizacao AND Animal.idJaula = Jaula.idJaula)
ORDER BY Localizacao.localizacao;
```

5. Lista dos tipos de jaula existentes por localização:

```
Select DISTINCT localizacao, tipoJaula
From Localizacao, TipoDeJaula, Jaula
Where Localizacao.idLocalizacao = Jaula.idLocalizacao
ORDER BY Localizacao.idLocalizacao;
```

6. Lista de todos os funcionários (tratadores e veterinários) do Zoo:

```
Select nomeTratador AS Funcionarios
From Tratador
UNION
Select nomeVeterinario
From Veterinario;
```

7. Todos os veterinários que estão a tratar um determinado animal:

```
Select nomeVeterinario, nome, nomeCondicao
From Veterinario, Animal, Condicao, DataTratamento, Tratamento
Where (Condicao.nomeCondicao = 'em tratamento' AND
      DataTratamento.idCondicao = Condicao.idCondicao AND
      DataTratamento.idTratamento = Tratamento.idTratamento AND
      Tratamento.idVeterinario = Veterinario.idVeterinario AND
      DataTratamento.idAnimal = Animal.idAnimal)
ORDER BY Veterinario.nomeVeterinario;
```

8. Lista de todos os animais cuja condição é diferente de saudável e ainda não estão em tratamento:

```
Select nome, nomeEspecie, nomeCondicao
From Animal, Condicao, Especie
Where Animal.idEspecie = Especie.idEspecie AND Animal.idCondicao = Condicao.idCondicao
GROUP BY Condicao.nomeCondicao
HAVING (Condicao.nomeCondicao != 'saudavel' AND Condicao.nomeCondicao != 'em tratamento'
AND Condicao.nomeCondicao != 'morto');
```

9. Retorna qual a espécie existente no Zoo que necessitou de mais tratamentos:

```
Select nomeEspecie, count(*) AS n_trat
From Tratamento, Especie, DataTratamento, Animal
Where (Animal.idEspecie = Especie.idEspecie AND DataTratamento.idAnimal = Animal.idAnimal
AND
      Tratamento.idTratamento = DataTratamento.idTratamento)
GROUP BY Especie.idEspecie
ORDER BY n_trat DESC
LIMIT 1;
```

10. Lista de pesos médios por espécie:

```
Select nomeEspecie AS especie, avg (Peso) AS Pesomedio
From Animal, Especie
Where (Animal.idEspecie = Especie.idEspecie)
```

group by Animal.idEspecie;

## TRIGGERS

1. Caso um animal morra (a sua condição é alterada para 'morto') é apagado da tabela:

```
DROP TRIGGER if exists AnimalMorre;  
create trigger AnimalMorre after update on Animal  
for each row  
when (Animal.Condicao = 'morto')  
begin  
delete from Animal where idAnimal = old.idAnimal;  
end;
```

2. Caso um animal apresente alguma condição que não seja saudável e passe a existir um tratamento para essa problema, a sua condição passa para 'tratamento':

```
DROP TRIGGER if exists TratarAnimal;  
create trigger TratarAnimal after insert on DataTratamento  
FOR EACH ROW  
When Animal.idAnimal = DataTratamento.idAnimal AND Animal.idCondicao != 0  
Begin  
Update Animal  
set idCondicao = (Select idCondicao from Animal  
                  where idCondicao = new.idCondicao)  
  
Where idDataTratamento = new.idDataTratamento;  
End;
```