

# Prova de Fundamentos de Bancos de Dados

## 2ª Prova

Prof. Carlos A. Heuser

Novembro de 2010

Prova *com* consulta – duas horas de duração

Considere a base de dados abaixo (base de dados idêntica a da 1ª prova, exceto no que tange ao tratamento de vazio). Esta base de dados é usada por uma oficina de manutenção de automóveis.

*/\* tabela de clientes cadastrados na oficina – cpf\_conjuge é o código do cônjuge do marido, caso este esteja na base de dados \*/*

```
CLIENTE (cpf, nome_cli, cpf_conjuge)
```

*/\* tabela com dados dos automóveis dos clientes da oficina \*/*

```
AUTOMOVEL (placa, no_chassis, modelo, cpf);  
          (cpf) references CLIENTE
```

*/\* tabela com as revisões periódicas programadas e feitas – para cada automóvel, a oficina cadastra todas revisões programadas –*

*Km e data\_programada são a quilometragem e a data em que deve ser feita a revisão*

*data\_ultim\_telef serve para informar quando o pessoal da oficina ligou para o cliente lembrando da provável necessidade de fazer a revisão – caso o cliente não tenha sido chamado, este campo contém vazio (NULL)*

*data\_executada e Km\_executada informa a data e a quilometragem de uma revisão que já foi executada – caso a revisão não tenha sido executada, estes campos contém vazio (NULL) \*/*

```
REVISAO (placa, Km, data_programada, data_ultim_telef,  
         data_executada, Km_executada)  
        (placa) references AUTOMOVEL
```

*/\* tabela com as peças usadas em cada revisão executada \*/*

```
PECA_REVISAO (placa, Km, cod_peca, quantidade)  
              (placa, Km) references REVISAO  
              (cod_peca) references PECA
```

*/\* tabela com as descrições das peças \*/*

```
PECA (cod_peca, descricao_peca)
```

Nome do aluno:

1. (Peso 1,5)

Sobre a base de dados acima, resolver a seguinte consulta usando *cálculo relacional*:

Obter as datas programadas de revisões do automóvel com número de chassis 'BNF1945', nas quais tenha sido usada ao menos uma peça de descrição 'Mancal 123'.

**Solução:**

```
{r|
  ∃ rev ∈ REVISAO (
    rev.data_programada = r.data_programada ∧
    ∃ aut ∈ AUTOMOVEL (
      aut.placa = rev.placa ∧
      aut.no_chassis = 'BNF1945'
    )
    ∧
    ∃ pec_rev ∈ PECA_REVISAO (
      pec_rev.placa=rev.placa ∧
      pec_rev.Km=rev.Km ∧
      ∃ peca ∈ PECA (
        peca.cdo_peca=pec_rev.cod_peca ∧
        peca.descricao_peca='Mancal 123'
      )
    )
  )
}
```

2. Sobre a base de dados acima, resolver as consultas abaixo usando *SQL*:

2.a (*Peso 1,5*)

Obter as placas dos automóveis com modelo 'Arabutã', de propriedade de uma pessoa de nome 'José Silva' e que ainda não tenham sofrido nenhuma revisão (ainda não têm nenhuma revisão na qual a quilometragem executada esteja preenchida).

**Solução:**

```
SELECT aut.placa
FROM AUTOMOVEL AS aut NATURAL JOIN CLIENTE as cli
WHERE cli.nome='José Silva' AND
      aut.modelo='Arabutã'
MINUS
SELECT rev.placa
FROM REVISAO AS rev
WHERE rev.KM_executada IS NOT NULL
```

ou

```
SELECT aut.placa
FROM AUTOMOVEL AS aut NATURAL JOIN CLIENTE as cli
WHERE cli.nome='José Silva' AND
      aut.modelo='Arabutã' AND
      aut.placa NOT IN
      (SELECT rev.placa
       FROM REVISAO AS rev
       WHERE rev.KM_executada IS NOT NULL
      )
```

2.b (*Peso 1,5*)

(ANULADA)

2.c (*Peso 1,5*)

Obter o código e a descrição de cada peça utilizada em todas revisões.

**Solução:** Obter o código e descrição de cada peça, tal que não exista revisão, na qual a peça não tenha sido usada.

```
SELECT cod_peca, descricao_peca
FROM PECA
WHERE
  NOT EXISTS (
    SELECT * FROM REVISAO
    WHERE
      NOT EXISTS (
        SELECT * FROM PECA_REVISAO
        WHERE placa=REVISAO.placa AND
              Km=REVISAO.Km AND
              cod_peca=PECA.cod_peca
      )
  )
```

2.d (*Peso 1*)

Obter o número de clientes que possuem automóveis.

**Solução:**

```
SELECT COUNT (distinct cpf)
FROM AUTOMOVEL
```

2.e (*Peso 1,5*)

(anulada)

2.f (*Peso 1,5*)

Por um erro de programação, foram introduzidos casamentos artificiais na base de dados. A característica dos casamentos incorretos é a transitividade. Mais precisamente, a tabela CLIENTE foi alterada no campo cpf\_conjuge, de forma a que ocorram situações como: A é casado com B, que por sua vez é casado com C. Este erro deverá ser corrigido manualmente. Para tal, é necessário obter uma tabela com duas colunas, relacionando o cpf do cliente com o cpf de seu cônjuge, o cpf do cliente com o cpf do cônjuge do cônjuge e assim por diante, recursivamente. Caso um cliente não tenha cônjuges ele não deve constar do resultado.

```
WITH RECURSIVE casal_n(cpf, cpf_conjuge)
(
    (SELECT cpf,
             cpf_conjuge
     FROM CLIENTE
     WHERE cpf_conjuge IS NOT NULL
    )
UNION
    (SELECT cli.cpf,
            casal_n.cpf_conjuge
     FROM CLIENTE AS cli,
            casal_n
     WHERE cli.cpf_conjuge=
            casal_n.cpf
    )
)
(SELECT * FROM casal_n)
```