

Prova de Fundamentos de Bancos de Dados

1ª Prova

Prof. Carlos A. Heuser

Setembro de 2008

Nome do aluno:

Prova *sem* consulta – duas horas de duração

1. (*Peso 2*)

Deseja-se projetar uma base de dados para uma revenda de automóveis. Nesta base de dados estão armazenadas informações sobre os modelos de automóveis à venda, juntamente com suas configurações e seus preços.

Para cada modelo de automóvel, a base de dados informa a sua descrição (algo como "Volkswagen Gol Kit básico"). Cada modelo tem um código identificador.

Um modelo pode ser comercializado com diferentes motores. Um motor possui uma potência em CV, uma cilindrada em litros e um número de válvulas (algo como <106 CV, 1.8 l, 16 V>). Além disso, cada motor é identificado por um código. Um determinado modelo de automóvel pode ser oferecido com várias motorizações e uma motorização pode aparecer em diferentes modelos.

O preço do automóvel é fixado com base no modelo e na motorização escolhida e também deve estar armazenado na base de dados.

Projete uma base de dados relacional para armazenar os dados acima sem redundância de dados. Enumere as tabelas, suas colunas, as chaves primárias e as chaves estrangeiras. Não devem ser criadas colunas artificiais, além das apresentadas no enunciado. Apresente o esquema na notação textual ou diagramática vistas em aula.

Solução:

```
Modelo (cod_mod, descricao_mod)
Motor (cod_motor, cv, cc, valve)
Motorizacao (cod_mod, cod_motor, preco)
            (cod_mod) referencia Modelo
            (cod_motor) referencia Motor
```

2. Considere a seguinte base de dados, usada por uma oficina de manutenção de automóveis.

/ tabela de clientes cadastrados na oficina */*

CLIENTE (cpf, nome_cli)

/ 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 a cadeia vazia (")

*data_executada e Km_executada informa a data e a quilometragem de uma revisão que já foi executada – caso a revisão não tenha sido chamado, estes campos contém a cadeia vazia (") */*

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

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

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

/ tabela com as descrições das peças */*

PECA_REVISAO (cod_peca, descricao_peca)

Sobre esta base de dados, resolver as consultas que seguem usando *álgebra relacional*. Não usar mais tabelas que o estritamente necessário.

a) (*Peso 1,33...*)

Obter os nomes dos clientes cujo automóvel teve uma revisão executada da data 2009-10-07.

i. Resolver usando produto cartesiano.

Solução:

```
 $\pi$  CLIENTE.nome_cli  
( $\sigma$  CLIENTE.CPF = AUTOMOVEL.CPF AND  
AUTOMOVEL.placa = REVISAO.placa AND  
REVISAO.data_executada = 2009-10-07  
(CLIENTE  $\times$   
AUTOMOVEL  $\times$   
REVISAO  
)  
)
```

ii. Resolver usando junções. Se possível, usar junção natural, senão, usar equi-junção e em último caso usar theta-junção.

Solução:

```
 $\pi$  CLIENTE.nome_cli  
( $\sigma$  REVISAO.data_executada = 2009-10-07  
(CLIENTE  $\bowtie$   
(AUTOMOVEL  $\bowtie$   
REVISAO  
)  
)  
)  
)
```

b) (*Peso 1,33...*)

Obter as placas dos automóveis que não têm manutenções por fazer (não têm manutenções em que a `textttdata_executada` é a cadeia vazia (`''`)).

Solução:

```
 $\pi$  placa (AUTOMOVEL)
—
( $\pi$  placa
  ( $\sigma$  data_executada = ''
    (REVISAO)
  )
)
```

c) (*Peso 1,33...*)

Foi detetado um problema na base de dados, gerado por um erro de software. Para alguma revisões, erroneamente, está informada que elas usaram todas peças cadastradas na base de dados. Obter os identificadores destas revisões (`placa`, `Km`)

Solução:

```
( $\pi$  placa, Km, cod_peca (PECA_REVISAO))
÷
( $\pi$  cod_peca (PECA))
```

d) (*Peso 1,33...*)

Obter **uma** tabela contendo as seguintes colunas:

- i. Código e nome de cada peça;
- ii. placa do automóvel, quilometragem da revisão e data de cada revisão, na qual mais que 3 unidades (quantidade) da peça tenham sido usadas (se não houver revisão nesta condições para a peça, estas colunas devem aparecer em branco).

Solução:

```
 $\pi$  PECA.cod_peca,  
    PECA.descricao_peca,  
    PECA_REVISAO.placa,  
    PECA_REVISAO.Km,  
    REVISAO.data_executada  
    (PECA  
       $\bowtie$   
      ( (  $\sigma$  (quantidade > 3)  
          (PECA_REVISAO)  
        )  
       $\bowtie$   
      REVISAO  
    )  
  )
```

3. (*Peso 1,33...*)

Sobre a base de dados da questão precedente (Questão 2), expresse a seguinte consulta em SQL, *sem* o uso da sintaxe para junções explícitas, *nem* sub-consultas (ainda não vistas em aula). Na consulta, devem aparecer apenas as tabelas necessárias.

Para cada cliente que possui dois automóveis diferentes, obter o nome do cliente, seguido das placas de cada um dos dois automóveis.

Solução:

```
SELECT C.nom_cli,  
       A1.placa,  
       A2.placa  
FROM CLIENTE AS C,  
     AUTOMOVEL AS A1,  
     AUTOMOVEL AS A2  
WHERE  
    A1.placa <> A2.placa AND  
    A1.cpf = C.cpf AND  
    A2.cpf = C.cpf
```

4. (Peso 1,33...)

Considere a seguinte consulta em SQL:

```
SELECT AUTOMOVEL.modelo
FROM AUTOMOVEL,
      REVISAO,
      PECA_REVISAO
WHERE
  AUTOMOVEL.placa = REVISAO.placa AND
  PECA_REVISAO.placa = REVISAO.placa AND
  PECA_REVISAO.Km = REVISAO.Km AND
  REVISAO.data_executada = 2009-10-07 AND
  PECA_REVISAO.quantidade = 2
```

Mostre a consulta equivalente em álgebra relacional, depois, mostre a representação da consulta em forma de árvore e após, mostre cada um dos passos da otimização algébrica.

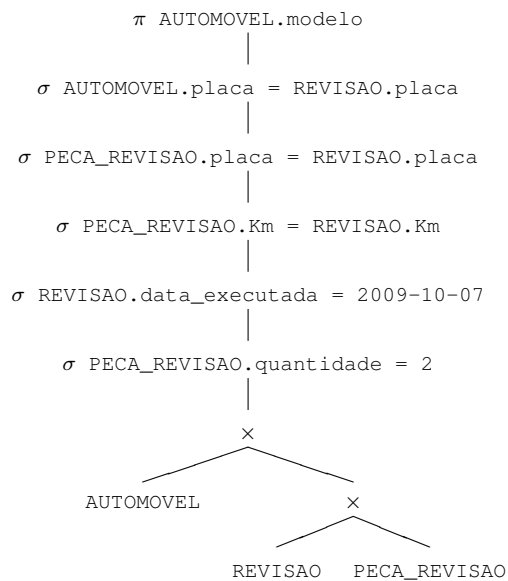
Resposta:

Solução:

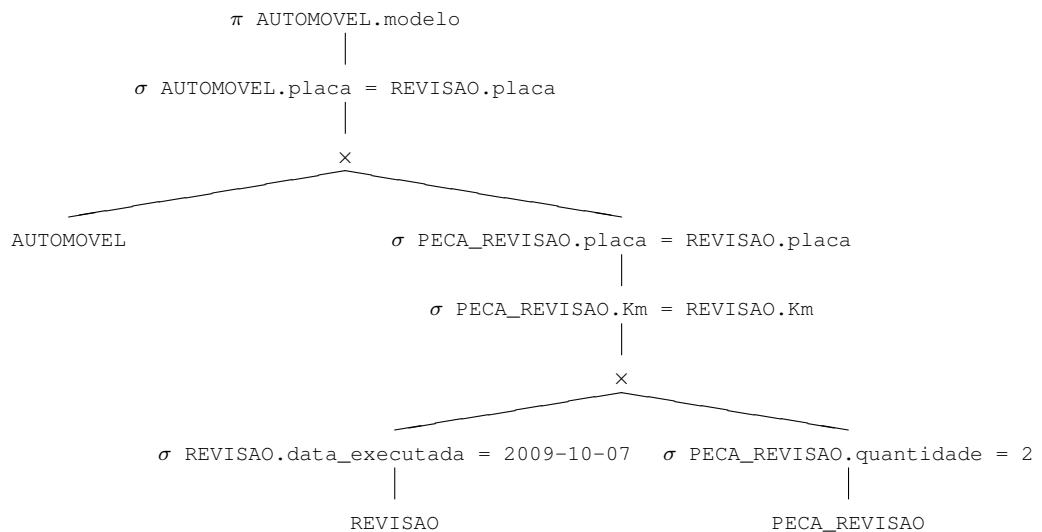
Consulta em álgebra relacional (a ordem dos produtos cartesianos foi arbitrada; outras ordens são equivalente e levam a uma consulta diferente com o mesmo resultado):

```
 $\pi$  AUTOMOVEL.modelo
 $\sigma$ 
  AUTOMOVEL.placa = REVISAO.placa AND
  PECA_REVISAO.placa = REVISAO.placa AND
  PECA_REVISAO.Km = REVISAO.Km AND
  REVISAO.data_executada = 2009-10-07 AND
  PECA_REVISAO.quantidade = 2
  ( AUTOMOVEL
    ×
    (REVISAO
      ×
      PECA_REVISAO
    )
  )
```

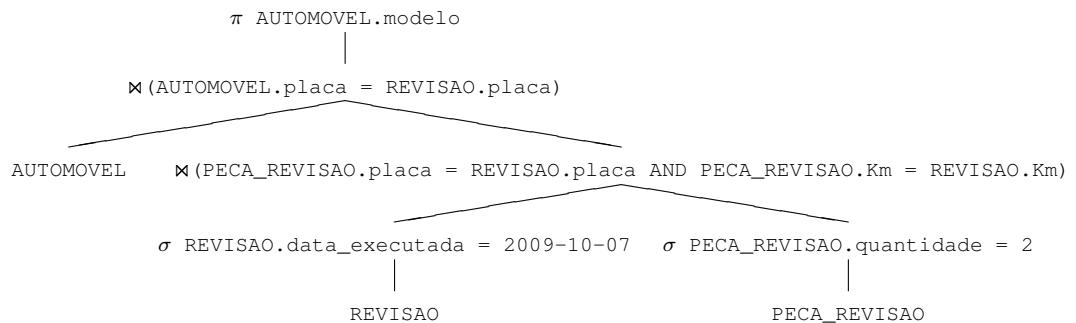
Representação da consulta em árvore:



Passo 1: Mover seleções para as folhas:



Passo 2: Substituir produtos cartesianos por junções:



Passo 3: Mover projeções para as folhas:

