

**UNIVERSIDADE FEDERAL FLUMINENSE**  
**PÓLO UNIVERSITÁRIO DE RIO DAS OSTRAS**  
**FACULDADE FEDERAL DE RIO DAS OSTRAS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**V. S. de Banco de Dados – 1º. Sem de 2008**

**Prof.: Carlos Bazilio**

**Aluno:**

**Matrícula:**

1. (5.0 pts) Considere o diagrama ER da Figura 1, representando o esquema conceitual, bastante simplificado, de um banco de dados da Agência Reguladora de Petróleo, onde:



**Empresa**, esquema de entidade representando empresas, com os seguintes atributos:

**CNPJ**, que é a chave

**Nome**

**Ano**, indica o ano de fundação da empresa

**Leilao**, esquema de entidade representando os leilões já realizados, com os seguintes atributos:

**Codigo**, que é chave

**Data**

**Bloco**, indica o bloco de exploração licitado

**Participou**, esquema de relacionamento entre **Leilao** e **Empresa**, indicando os leilões que uma empresa já participou, com 2 atributos

**Vencedor**, indica se a empresa foi vencedora

Vencedor = 1 : empresa venceu o leilão

Vencedor = 0 : empresa não venceu o leilão

Vencedor = null : empresa foi desclassificada

**Lance**, indica o valor do lance da empresa no leilão

- a) (0.5) Apresente operações de criação de tabelas **em SQL** para o modelo acima. Defina estas tabelas de tal forma a capturar todas as restrições de integridade do esquema ER.

```

create table Empresa (
    CNPJ numeric(15) not null,
    Nome char(20) not null,
    Ano date not null,
    /* chave de Empresa */
    primary key(CNPJ)
)
create table Leilao (
    Cod char(7) not null,
    Data date not null,
    Bloco char(10) not null,
    /* chave de Leilao */
    primary key(Cod)
)
create table Participou (
    CNPJ numeric(15) not null,
    Cod char(7) not null,
    Vencedor numeric(1)
    /* domínio restrito */ check(value in (0, 1)),
    Lance numeric(10) not null,
    /* chave de Participou */
    primary key(CNPJ, Cod),

    /* chaves estrangeiras para Empresa e Leilão */
    foreign key(CNPJ) references Empresa,
    foreign key(Cod) references Leilao
)

```

- b) (0.5) É possível representar a entidade Participou e a entidade Empresa em uma única relação sem violar as formas normais? **Explique sua resposta, com detalhes, através de um exemplo.** Não, pois criaríamos uma tabela que viola a 2NF. De fato, criaríamos a tabela

EP (CNPJ, Cod, Nome, Ano, Vencedor, Lance)

com chave {CNPJ,Cod}, onde vale a dependência funcional EP: CNPJ@Nome. Logo, o atributo Nome depende parcialmente da chave.

EP	CNPJ	Cod	Nome	Ano	Vendedor	Lance
	0001	1	ABC	2005	1	5000
	0001	2	ABC	2005	2	8000

Para mudar o nome da empresa com CNPJ 0001 de ABC para ACME, seria necessário alterar 2 linhas da tabela, o que caracteriza uma anomalia.

- c) (0.8) Formule a seguinte consulta em SQL: “Liste o nome da empresa e o valor do lance, para cada empresa que participou de leilões em 2006.”

```
select E.Nome, P.Lance
from Empresa E, Participou P, Leilao L
where E.CNPJ = P.CNPJ
      and P.Cod = L.Cod
      and L.Data = '2006';
```

- d) (0.8) Formule a seguinte consulta em SQL: “Liste o código e a data dos leilões para os quais todas as empresas participantes foram desclassificadas.”

```
select L.Cod, L.Data
from Leilao L
where not exists (
    select *
    from Participou P
    where L.Cod = P.Cod
          and P.Vencedor <> null);
```

- e) (0.8) Formule a seguinte consulta em SQL: “Para os leilões realizados em 2007 e com quantidade de lances maior que 5, liste o código do leilão, a quantidade de lances e a respectiva diferença entre o maior e menor lance feito para este leilão.”.

```
select P.Cod, count(*) as participacoes, max(P.Lance) -
min(P.Lance) as diferenca
from Leilao L, Participou P
where L.Data = '2007'
      and L.Cod = P.Cod
group by P.Cod
having (participacoes > 5);
```

- f) (0.8) Formule a seguinte consulta em SQL: “Liste os nomes das empresas que participaram de todos os leilões”.

```
select E.Nome
from Empresa E
where not exists (
    select *
    from Leilao L
    where not exists (
        select *
        from Participou P
        where E.CNPJ = P.CNPJ and L.Cod = P.Cod ) );
```

- g) (0.8) Formule a(s) seguinte(s) operação(ões) em SQL, tomando cuidado para que o banco de dados seja consistente sempre que possível: “Remova da base de dados todos os leilões que ocorreram antes de 2000. Em seguida, empresas sem participação em leilões também devem ser removidas”. Lembre que operações de inserção e remoção podem ser combinadas com operações de consulta.

```
delete from Participou P
where P.Cod IN (
    select P1.Cod
    from Leilao L, Participou P1
    where L.Cod = P1.Cod
        and L.Data < '2000' );
delete from Leilao L
where L.Data < '2000';
delete from Empresa E
where E.CNPJ NOT IN (
    select P.CNPJ
    from Participou P );
```

2. (2.0 pts) Formule a consulta da 1.c) em Álgebra Relacional de duas formas distintas e argumente qual das duas deverá ser mais eficiente. **Justifique** exemplificando (fornecendo valores para tuplas como exemplo) e/ou fornecendo dados quantitativos (suponha uma quantidade de tuplas nas relações envolvidas).

```
(Pi (Sigma (Leilao x (Empresa x Participou) Empresa.CNPJ=Participou.CNPJ) Participou.Cod=Leilao.Cod)
Leilao.Data='2006) Empresa.Nome, Participou.Lance
```

```
(Pi (Empresa x (Pi (Participou x (Pi (Sigma
(Leilao) Leilao.Data='2006) Leilao.Cod) Participou.Cod=Leilao.Cod) Participou.CNPJ, Participou.Lance) Participou.CNPJ
=Empresa.CNPJ)) Empresa.Nome, Participou.Lance
```

A segunda forma é mais eficiente pois reduz o tamanho das tabelas que participam das junções. Neste ponto, a exemplificação poderia ser dada com algum exemplo fornecido com quantidades de tuplas para as relações Empresa, Leilão e Participou, e o tamanho das tabelas geradas durante o processo.

3. (3.0 pts) Seja o esquema de relação abaixo (a chave está sublinhada):

Filme (Nome, Cinema, Distribuidora, País)

onde Nome é o nome do filme, Cinema é o cinema de exibição, Distribuidora é a distribuidora do filme e País é o país da distribuidora. Para a relação filme temos o conjunto de dependências funcionais abaixo:

Nome → Distribuidora

Distribuidora → País

- a) Uma **anomalia** num banco de dados ocorre quando alguma operação realiza mais trabalho do que o necessário por erro de projeto do banco de dados. Por exemplo, se mantemos uma base de dados de alunos e guardamos o seu endereço em 2 pontos diferentes num banco de dados, uma alteração do endereço deveria ser realizada nestes 2 pontos. Isto caracterizaria uma **anomalia de alteração**. Também podemos ter anomalias de inclusão e exclusão. Há anomalias para a relação Filme? **Justifique sua resposta com exemplos.**

Sim. Suponha os seguintes valores para a relação Filme:

('Xuxa', 'Petrópolis', 'Globo Filmes', 'BR')

('Didi', 'Rio das Ostras', 'Globo Filmes', 'BR')

Se alterarmos o país de uma distribuidora, todas as ocorrências da distribuidora deverão ser modificadas. O mesmo vale para o nome da distribuidora.

- b) Em qual forma normal está a relação filme? Caso não esteja na 3a. Forma Normal, realize decomposições, **justificando cada passo**, para que a relação respeite esta Forma Normal. Ainda podemos encontrar algum tipo de anomalia nesta relação? **Justifique a sua resposta com exemplos.**

1FN, pois não há atributos multivalorados. Além disso, há uma dependência parcial da chave primária {Nome @ Distribuidora}, o que determina que esta relação não está na 2FN.

Para que ela esteja, realizamos a decomposição:

Filme (Nome, Cinema)

Dist (Nome, Distribuidora, País)

Estas estão na 2FN, mas Dist não está na 3FN pois existe uma dependência transitiva com respeito à chave primária {Nome @ Distribuidora @ País}. Para adequá-la à 3FN realizamos uma nova decomposição:

Filme (Nome, Cinema)

Distribuição (Nome, Distribuidora)

Dist (Distribuidora, País)

Finalmente as relações respeitam a 3FN. Não há anomalias pois todas as redundâncias foram excluídas.