



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AD2 2º semestre de 2012.

Nome: _____

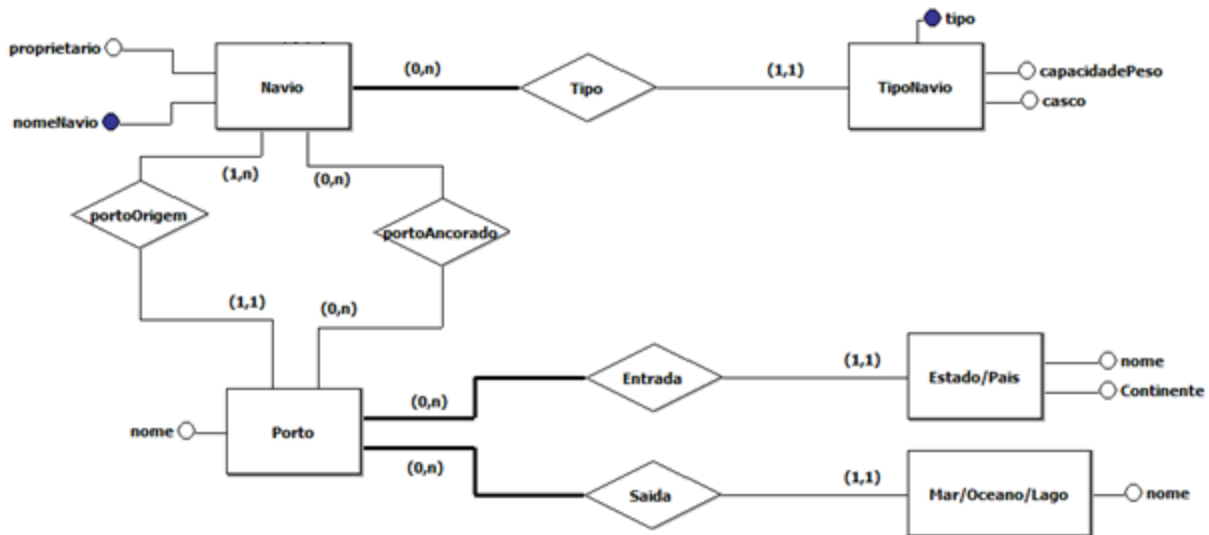
Observações:

1. Prova COM consulta.

Atenção: Como a avaliação à distância é individual, caso seja constatado que provas de alunos distintos são cópias umas das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual.

ADs enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.

Questão 1 (2,0 pontos). Considere o diagrama ER abaixo apresentado na AD1. Você já aprendeu a gerar um modelo ER, agora você deve projetar o esquema de uma base de dados relacional para o modelo ER em questão. A base de dados deve refletir exatamente o especificado no modelo conceitual. O esquema da base de dados relacional deve conter os nomes das tabelas, os nomes dos atributos, atributos que formam a chave primária e as chaves estrangeiras. Quando houver alternativas de projeto dizer que alternativa foi usada.



EstadoPais (nome, continente);
 MarOceanoLago (nome);
 Porto (nome, entrada, saida);
 entrada referencia EstadoPais;
 saida referencia MarOceanoLago;
 TipoNavio (tipo, capacidadePeso, casco);
 Navio (nomeNavio, proprietario, tipo, portoOrigem);
 tipo referencia TipoNavio;
 portoOrigem referencia Porto;
 NavioAcorado (nomeNavio, nomePorto);
 nomeNavio referencia Navio;
 nomePorto referencia Porto;

Questão 2 (3,0 pontos – 0,3 cada). Considere o seguinte esquema relacional:

Jogador (jid: integer, jnome: string, end: string)
 QuadraTennis (qtid: integer, qtnome: string, tid: integer, preço: real)
 tid referencia Tipo
 Aluguel (jid: integer, qtid: integer, horas: integer)
 jid referencia Jogador
 qtid referencia QuadraTennis
 Tipo (tid: integer, tnome: string)

No esquema acima, as chaves primárias estão sublinhadas. A tabela de Aluguel lista a quantidade de horas alugadas em uma determinada quadra de tênis por um jogador. Apresente, para cada consulta a seguir, as expressões em álgebra relacional correspondentes às consultas.

1. Obtenha o qtid da quadra de nome "Quadra de Saibro 14".

$\Pi_{qtid} (\sigma_{qtname="Quadra de Saibro 14"} QuadraTenis)$

2. Obtenha o nome das quadras que foram alugadas pelo menos uma vez.

$\pi_{qtname} (Quadra \bowtie Aluguel)$

3. Obtenha o nome dos jogadores que alugaram quadras do tipo "Grama" e o nome da quadra alugada.

$\pi_{jnome, qtname} (Jogador \bowtie Aluguel \bowtie QuadraTenis \bowtie \sigma_{tname="Grama"} Tipo)$

4. Obtenha o nome das quadras do tipo "Lisonda" que tenham preço superior a 150.

$\pi_{qtname} ((\sigma_{tname="Lisonda"} Tipo) \bowtie (\sigma_{preço > 150} QuadraTenis))$

5. Obtenha o qtid das quadras dos tipos "Grama" ou "DuraFlor".

$\pi_{qtid} ((\sigma_{tname="Grama"} \vee tname="DuraFlor" Tipo) \bowtie QuadraTenis)$

6. Obtenha o nome dos jogadores que alugaram alguma quadra do tipo "Grama" por mais de 10 horas.

$\pi_{jnome} ((\sigma_{horas > 10} Aluguel \bowtie (QuadraTenis \bowtie \sigma_{tname="Grama"} Tipo)) \bowtie Jogador)$

7. Obtenha o jid dos jogadores que alugaram por mais de 20 horas uma quadra ou que alugaram quadras do tipo "Lisonda".

$\rho(R1, \pi_{jid}(\sigma_{quantidade > 20} Aluguel))$

$\rho(R2, \pi_{jid}(\pi_{qtid}((\pi_{tid} \sigma_{tname="Lisonda"} Tipo) \bowtie QuadraTenis) \bowtie Aluguel))$

$R1 \cup R2$

8. Obtenha o nome dos jogadores que alugaram a quadra "Saibro verde 2" e a quadra "Grama K" por mais de 10 horas cada.

$\rho(R1, \pi_{jid}((\pi_{qtid} \sigma_{qtname="Saibro verde 2"} QuadraTenis) \sigma_{horas > 10} Aluguel))$

$\rho(R1, \pi_{jid}((\pi_{qtid} \sigma_{qtname="Grama K"} QuadraTenis) \sigma_{horas > 10} Aluguel))$

$\rho(R3, R1 \cap R2)$

$\pi_{jnome}(Jogador \bowtie R3)$

9. Obtenha o nome dos jogadores que alugaram alguma quadra de preço abaixo de 10.

$\pi_{jnome} ((\sigma_{preço < 100} QuadraTenis) \bowtie Aluguel) \bowtie Jogador)$

10. Obtenha os nomes dos tipos de quadra alugadas pelo cliente "Daniel de Oliveira" e que não foram alugadas pelo cliente "Roger Federer".

$\rho(R1, \pi_{tname} (((\sigma_{jnome="Daniel de Oliveira"} Jogador) \bowtie Aluguel) \bowtie QuadraTenis) \bowtie Tipo))$

$\rho(R2, \pi_{tname} (((\sigma_{jnome="Roger Federer"} Jogador) \bowtie Aluguel) \bowtie QuadraTenis) \bowtie Tipo))$

$R1 - R2$

Questão 3 (2,0 pontos). Considere o esquema relacional abaixo que representa o controle de publicações em eventos científicos. As chaves primárias estão sublinhadas.

| |
|---|
| Departamento (<u>CodDepto</u> , Nome) |
| Autor (<u>CodAutor</u> , nome, CodDepto) |
| CodDepto referencia Departamento |
| Conferência(<u>CodConf</u> , Nome) |
| Publicação (<u>CodPub</u> , título, ano, CodConf) |
| CodConf referencia Conferência |
| PublicaçãoAutor (<u>CodAutor</u> , <u>CodPub</u>) |
| CodAutor referencia Autor |
| CodPub REFERENCIA Publicação |

1. Escreva os comandos SQL para criar as tabelas Autor, Publicação e PublicaçãoAutor, incluindo as restrições de integridade que se aplicam. Assuma que quando um departamento é excluído, o código de departamento dos autores relacionados a ele deve ser alterado para NULL. Assuma ainda que ao excluir um autor, todas as suas publicações devem ser excluídas automaticamente. Assuma também que uma conferência não pode ser excluída se houver alguma publicação relacionada. Além disso, ao alterar o código de uma conferência, todas as publicações relacionadas devem ser alteradas automaticamente. Em uma publicação, assumo que o ano não pode ser nulo. Uma tupla na relação PublicaçãoAutor só deve existir se existir uma tupla correspondente na tabela Publicação. Se, por algum motivo, não for possível definir alguma restrição de integridade, justifique.

```
CREATE TABLE AUTOR (  
    CODAUTOR INT NOT NULL,  
    NOME VARCHAR(30),  
    CODDEPTO INT,  
    PRIMARY KEY (CODAUTOR),  
    FOREIGN KEY CODDEPTO REFERENCES DEPARTAMENTO (CODDEPT) ON DELETE SET  
    NULL  
)
```

```
CREATE TABLE PUBLICACAO (  
    CODPUB INT NOT NULL,  
    TITULO VARCHAR(30),  
    ANO INT NOT NULL,  
    CODCONF INT,  
    PRIMARY KEY (CODPUB),
```

```
FOREIGN KEY (CODCONF) REFERENCES CONFERENCIA (CODCONF) ON DELETE  
RESTRICT ON UPDATE CASCADE  
)
```

```
CREATE TABLE PUBLICACAOAUTOR (  
    CODAUTOR INT NOT NULL,  
    CODPUB INT NOT NULL,  
    PRIMARY KEY (CODAUTOR, CODPUB),  
    FOREIGN KEY (CODAUTOR) REFERENCES AUTOR (CODAUTOR) ON DELETE CASCADE,  
    FOREIGN KEY (CODPUB) REFERENCES PUBLICACAO (CODPUB) ON DELETE CASCADE  
)
```

É possível definir todas as restrições de integridade.

2. Escreva comandos SQL para incluir uma nova publicação no banco de dados, com os seguintes dados: Publicação código 111 do autor 05 (assuma que o autor já está cadastrado) na conferência 122 (assuma que a conferência já está cadastrada). Os detalhes da publicação são: título “Many task computing for large clusters”, ano 2010.

```
INSERT INTO PUBLICACAO (CODPUB, TITULO, ANO, CODCONF)  
VALUES (111, “Many task computing for large clusters”, 2010, 122);
```

```
INSERT INTO PUBLICACAOAUTOR (CODAUTOR, CODPUB)  
VALUES (05, 111);
```

3. Escreva um comando SQL para excluir todas as publicações do autor “Daniel de Oliveira”.

```
DELETE FROM PUBLICACAO  
WHERE CODPUB IN (SELECT CODPUB  
FROM PUBLICACAOAUTOR p, AUTOR a  
WHERE p.CODAUTOR=a.CODAUTOR AND  
a.NOME=“Daniel de Oliveira”)
```

NÃO é necessário excluir de PublicaçãoAutor devido à restrição DELETE CASCADE.

Questão 4 (2,0 pontos). Considere o esquema relacional abaixo de uma companhia de ônibus da cidade do Rio de Janeiro. As chaves primárias estão sublinhadas.

```
Bairro(cd_bairro, cd_municipio, ch_nome, dt_exclui)
      cd_municipio referencia Municipio
Municipio(cd_municipio, cd_pais, ch_nome, dt_exclui)
      cd_pais referencia Pais
Rodoviaria(sq_rod, cd_municipio, ch_nome, dt_exclui)
      cd_municipio referencia Municipio
Cliente(cd_cliente, cd_bairro, cd_telefone, ch_nome, dt_nasc, ch_endereco)
      cd_telefone referencia Telefone
Pais(cd_pais, ch_nome, dt_exclui)
Parada(cd_trecho, sq_rod_origem, sq_rod_destino, cd_viagem, dt_origem, dt_destino)
      sq_rod_origem referencia Rodoviaria
      sq_rod_destino referencia Rodoviaria
      cd_viagem referencia Viagem
Viagem(cd_viagem, nu_viagem, qtd_passagem, dt_partida, dt_chegada, dt_exclui)
Passagem(cd_passagem, cd_cliente, cd_viagem, valor)
      cd_cliente referencia Cliente
      cd_viagem referencia Viagem
Telefone(cd_telefone, cd_cliente, cd_tipo_telefone, nu_ddd, nu_telefone)
      cd_tipo_telefone referencia Tipo_Telefone
Tipo_Telefone(cd_tipo_telefone, ch_codigo, ch_nome)
```

1. Escreva um comando SQL para excluir todos os clientes cujo nome comece por "Marta" e termine com "Mattoso".

DELETE FROM CLIENTE WHERE CH_NOME LIKE "MARTA%MATTOSO"

2. Escreva um comando SQL que atualize o nome do município "Frorianopolis" para "Florianópolis".

UPDATE MUNICIPIO SET CH_NOME = "FRORIANOPOLIS" WHERE CH_NOME = "FLORIANÓPOLIS"

3. Escreva um comando SQL que lista todos os municípios que possuem mais do que uma rodoviária.

```

SELECT M.CH_NOME, COUNT(*)
FROM MUNICIPIO M, RODOVIARIA R
WHERE M.CD_MUNICIPIO = R. CD_MUNICIPIO
GROUP BY M.CH_NOME
HAVING COUNT(*) > 10

```

4. Escreva um comando SQL que lista todos os passageiros que possuem viagens não realizadas. Considere nesta questão que existe uma função date() que retorna a data atual do sistema.

```

SELECT DISTINCT C.CH_NOME FROM
CLIENTE C, VIAGEM V, PASSAGEM P
WHERE C.CD_CLIENTE = P. CD_CLIENTE
AND V.CD_VIAGEM = P. CD_VIAGEM
AND V.DT_PARTIDA > DATE()

```

5. Escreva um comando SQL que lista todos os passageiros que possuem TODAS as suas viagens não realizadas. Considere nesta questão que existe uma função date() que retorna a data atual do sistema.

```

SELECT C.CH_NOME FROM
CLIENTE C
WHERE NOT EXISTS (
SELECT * FROM PASSAGEM P, VIAGEM V
WHERE P.CD_VIAGEM = V. CD_VIAGEM
AND P. CD_CLIENTE = C.CD_CLIENTE
AND (V.DT_PARTIDA < DATE() OR DT_PARTIDA = DATE())
)

```

OU

```

SELECT C.CH_NOME FROM
CLIENTE C
WHERE C.CD_CLIENTE NOT IN(
SELECT C2.CD_CLIENTE FROM
CLIENTE C2, PASSAGEM P, VIAGEM V
WHERE P. CD_VIAGEM = V. CD_VIAGEM
AND P. CD_CLIENTE = C2.CD_CLIENTE
AND (V.DT_PARTIDA < DATE() OR DT_PARTIDA = DATE())
)

```

Questão 5 (1,0 ponto). Considere a seguinte tabela feita por um profissional que não foi aluno do CEDERJ, que não se encontra normalizada, de uma base de dados referente a uma rede de lanchonetes *fast food* (as chaves primárias estão sublinhadas):

Lanchonete (CodLanchonete, Nome, CodLocal, NomeLocal (CodSanduiche, NomeSanduiche, Tipo))

O significado de cada coluna é o seguinte:

- CodLanchonete: código da lanchonete
- Nome: nome da lanchonete
- CodLocal: código da localidade onde a lanchonete se localiza
- NomeLocal: nome da localidade onde a lanchonete se localiza
- CodSanduiche: código do sanduiche
- Nome: nome do sanduiche
- Hora: hora em que o sanduiche foi feito
- Validade: hora em que o sanduiche perde a validade
- Tipo: classificação do sanduiche

As dependências funcionais (podendo incluir dependências transitivas) que existem nesta tabela são as seguintes:

CodLanchonete → Nome

CodLocal → NomeLocal

CodLanchonete → CodLocal

(CodLanchonete, CodSanduiche) → Tipo

CodSanduiche → NomeSanduiche

1. Mostre a transformação da tabela para a terceira forma normal. Mostre cada forma normal intermediária, entre aquela em que a tabela se encontra e a terceira forma normal.

Passagem para a Primeira Forma Normal

Lanchonete (CodLanchonete, Nome, CodLocal, NomeLocal)
Sanduiche (CodSanduiche, NomeSanduiche, Tipo)

Passagem para a Segunda Forma Normal

Já está.

Passagem para a Terceira Forma Normal

Lanchonete (CodLanchonete, Nome, CodLocal)

Localidade (CodLocal, NomeLocal)

Sanduiche (CodSanduiche, NomeSanduiche, Tipo)