

Exercícios de fixação

Deseja-se informatizar uma clínica médica, armazenando as informações de seus **pacientes**, dos **convênios** utilizados por estes **pacientes**, e dos exames realizados por eles.

- É necessário armazenar o nome, CPF, telefone e endereço de cada paciente;
- Os convênios possuem nome, CNPJ, nome e telefone da pessoa de contato;
- Os exames possuem um código e um **tipo**, além de um diagnóstico (texto) associado;
- Cada paciente pode realizar vários exames na clínica, sendo que cada exame pode ser feito por um convênio diferente;
- O sistema deve permitir a geração de um relatório mensal de todos os exames realizados no mês por um determinado convênio.

1) Defina formalmente o esquema relacional incluindo todos os atributos declarados no problema e também os atributos que serão necessários para satisfazer a todos os requisitos.

⇒ indique a chave primária e as restrições de integridade referencial de cada esquema de relação;
⇒ use a notação vista de conjunto vista em aula (apenas texto), ou use um editor online de diagramas relacionais (como o <https://dbdiagram.io>, <http://draw.io>, <https://www.lucidchart.com>, dentre outros)

2) Para cada restrição de integridade que você definiu, exemplifique uma tupla inválida, isto é, que não satisfaça às regras estabelecidas em projeto;

3) Escreva o correspondente código DDL para definição do esquema de dados;

4) Escreva o código DML para inserir, pelo menos, duas tuplas em cada relação.

⇒ Execute o seu código SQL em sua base de dados Oracle (ou PostgreSQL, ou MySQL).

Considere os dados da Copa do Mundo (WorldCupMatches.csv e WorldCups.csv). Escreva o correspondente Esquema Relacional definindo o correspondente relacionamento entre os dois conjuntos de dados.

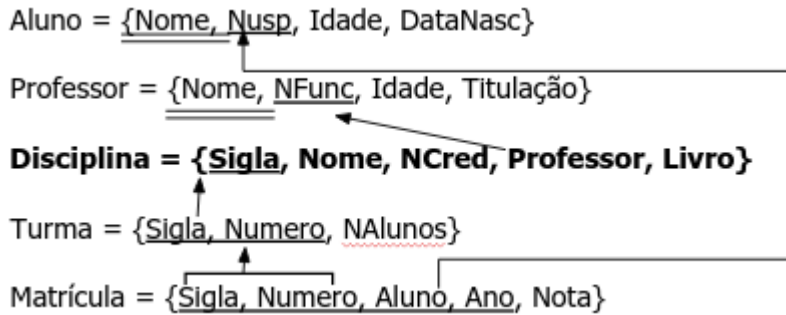
5) Defina formalmente o esquema relacional;

⇒ indique a chave primária e as restrições de integridade referencial de cada esquema de relação;
⇒ use a notação vista nas aulas (apenas texto) ou use um editor online de diagramas relacionais (como o <https://dbdiagram.io>, <http://draw.io>, <https://www.lucidchart.com>, dentre outros)

6) Escreva o correspondente código DDL para definição do esquema de dados;

7) Escreva um programa python que lê os dados Copa do Mundo, e escreve os correspondentes comandos INSERT; em seguida, execute os comandos gerados em sua base de dados Oracle (ou PostgreSQL, ou MySQL).

8) Usando Python ou o SQLDeveloper (ou o PGAdmin), carregue o esquema e os dados do Esquema Universidade.



```
1. Descompactar "EsquemaUniversidade.zip"
2. Rodar no SQLDeveloper "Universidade-Create.sql"
3. Rodar no SQLDeveloper "Universidade-Insert.sql"

-- CHECK create table e inserts (dados de apoio)
SELECT 'aluno', COUNT(*) FROM ALUNO UNION
SELECT 'PROFESSOR', COUNT(*) FROM PROFESSOR UNION
SELECT 'Disciplina', COUNT(*) FROM Disciplina UNION
SELECT 'Turma', COUNT(*) FROM Turma UNION
SELECT 'Matricula', COUNT(*) FROM Matricula;

/* Contagens Colhidas
'ALUNO'          COUNT(*)
-----
Disciplina        10
Matricula         46
PROFESSOR         5
Turma            20
aluno            10*/
```

Faça alterações no esquema

9) Adicione uma entidade Laboratório(id_lab, nome_lab, capacidade, especialidade);

10) Altere a entidade Turma para que cada turma tenha um laboratório associado, sendo que um mesmo laboratório pode estar associado a mais de uma turma;

11) Após as alterações, desenhe o diagrama relacional de todo o projeto.

12) Insira 4 laboratórios no banco de dados;

13) Associe as turmas existentes aos laboratórios que foram criados usando o comando UPDATE; deixe algumas turmas sem laboratório; e, pelo menos, um laboratório não vinculado a qualquer turma;

14) Formule uma consulta que relate quais turmas estão vinculadas a quais laboratórios, incluindo as turmas sem nenhum laboratório vinculado, e os laboratórios sem nenhuma turma vinculada;

15) Formule uma consulta para computar o total de alunos vinculados a cada laboratório; por exemplo, se duas turmas estão vinculadas ao laboratório 1, uma com 50 alunos e outra com 40, o total de alunos vinculados ao laboratório é de 90 alunos;

Ordene o resultado pelo total de alunos.

⇒ Sua consulta terá uma junção, uma agregação, e uma ordenação.

16) Formule uma consulta que retorne, para cada aluno, o nome da disciplina, o ano de todas as suas turmas e a nota obtida, o nome do respectivo professor, e o nome do laboratório vinculado. Desconsidere os alunos que não frequentam nenhum laboratório, isto é, basta um inner join.

Consultas aninhadas.

Liste a sigla e o nome das disciplinas que não possuem alunos matriculados:

17) Usando junção;

18) Usando consulta aninhada correlacionada (EXISTS).

19) Usando consulta aninhada não-correlacionada (IN).

Em uma conexão MongoDB

20) Crie uma coleção denominada Universidade;

21) Crie um documento que Inclua todas as informações do esquema Universidade, isto é:

```
Aluno{
  Nome
  NUSP
  Idade
  DataNasc
  Matricula{
    Disciplina{
      Sigla
      Nome
      NCred
      Livro
      Professor{
        NFunc
        Nome
        Idade
        Titulacao
      }
    }
    Numero_da_turma
    Num_alunos_da_turma
    Ano
    Nota
  }
}
```

22) Observando a estrutura do documento Aluno, identifique duas situações nas quais uma situação de inconsistência pode ocorrer com relação a outros documentos que possam ser inseridos na base de dados.

23) Em MongoDB, e em outras soluções NoSQL, a escalabilidade horizontal é conseguida por meio de:

- a) Múltiplos discos rígidos organizados por meio de tecnologia RAID;
- b) Pela instalação de mais pentes de memória, espaço em disco, processadores com múltiplos cores, e largura de banda ampliada;
- c) Pela técnica denominada Sharding, a qual prevê o particionamento horizontal (subconjuntos de documentos) dos dados, com cada partição sendo associada a um nó de armazenamento e processamento;
- d) Contratando-se mais administradores de bancos de dados, os quais deverão ser alocados em uma mesma unidade física das instalações da empresa.