

Material on-line encontra-se disponível para todos os exercícios deste capítulo, em inglês, na página web do livro em

<http://www.cs.wisc.edu/~dbbook>

O material inclui scripts para criar as tabelas de cada exercício para uso com o Oracle, o IBM DB2, o Microsoft SQL Server e o MySQL.

Exercício 5.1 Considere as seguintes relações:

Aluno(nroAlun: integer, nomeAlun: string, formacao: string, nível: string, idade: integer)

Curso(nome: string, horário: time, sala: string, idProf: integer)

Matriculado(nroAlun: integer, nomeCurso: string)

Professor(idProf: integer, nomeProf: string, idDepto: integer)

O significado dessas relações é direto; por exemplo, Matriculado tem um registro por par aluno-curso tal que o aluno está matriculado no curso.

Escreva as seguintes consultas em SQL. Nenhuma duplicata deve ser impressa em qualquer uma das respostas.

1. Encontre os nomes de todos os Juniors (nível = JR) que estão matriculados em um curso ministrado por I. Teach.
2. Encontre o nome do aluno mais velho que é formado em História ou matriculado em um curso ministrado por I. Teach.
3. Encontre os nomes de todos os cursos que são ministrados na sala R128 ou que têm cinco ou mais alunos matriculados.
4. Encontre os nomes de todos os alunos que estão matriculados em dois cursos que são ministrados no mesmo horário.
5. Encontre os nomes dos professores que ministram cursos em todas as salas em que algum curso é ministrado.
6. Encontre os nomes dos professores para os quais a lista de matriculados dos cursos que eles ministram é menor do que cinco.
7. Imprima o nível e a idade média dos alunos desse nível, para cada nível.
8. Imprima o nível e a idade média dos alunos desse nível, para todos os níveis exceto JR.
9. Para cada professor que ministra cursos apenas na sala R128, imprima seu nome e o número total de cursos que ele ou ela ministra.
10. Encontre os nomes dos alunos matriculados no número máximo de cursos.
11. Encontre os nomes dos alunos não matriculados em nenhum curso.
12. Para cada valor de idade que aparece em Aluno, encontre o valor do nível que aparece com mais frequência. Por exemplo, se houver mais alunos no nível FR com idade 18 do que os alunos com idade 18 dos níveis SR, JR ou SO, você deve imprimir o par (18, FR).

Exercício 5.2 Considere o seguinte esquema:

Fornecedores(idForn: integer, *nomeForn: string*, *endereço: string*)

Peças(idPeça: integer, *nomePeça: string*, *cor: string*)

Catálogo(idForn: integer, idPeça: integer, *custo: real*)

A relação Catálogo lista os preços das Peças cobradas pelos Fornecedores. Escreva as seguintes consultas em SQL:

1. Encontre os *nomePeças* das peças para as quais há algum fornecedor.
2. Encontre os *nomeForns* dos fornecedores que fornecem todas as peças.
3. Encontre os *nomeForns* dos fornecedores que fornecem todas as peças vermelhas.
4. Encontre os *nomePeças* das peças fornecidas pela Acme Widget Suppliers e por nenhum outro fornecedor.
5. Encontre os *idForns* dos fornecedores que cobram mais por alguma peça do que o custo médio dessa peça (calculado com base em todos os fornecedores que fornecem essa peça).
6. Para cada peça, encontre o *nomeForn* do fornecedor que cobra mais por essa peça.
7. Encontre os *idForns* dos fornecedores que fornecem apenas peças vermelhas.
8. Encontre os *idForns* dos fornecedores que fornecem uma peça vermelha e uma peça verde.
9. Encontre os *idForns* dos fornecedores que fornecem uma peça vermelha ou uma peça verde.
10. Para todo fornecedor que fornece apenas peças verdes, imprima o nome do fornecedor e o número total de peças que ele fornece.
11. Para todo fornecedor que fornece uma peça verde e uma peça vermelha, imprima o nome e o preço da peça mais cara que ele fornece.

Exercício 5.3 As seguintes relações controlam as informações sobre empresas aéreas:

```
Vôos(nroVoo: integer, de: string, para: string, distância: time,  
partida: time, chegada: time, preco: integer)  
Aeronave(idAero: integer, nomeAero: string, distLimite: integer)  
Certificado(idFuncion: integer, idAero: integer)  
Funcionários(idFuncion: integer, nomeFuncion: string, salário: integer)
```

Observe que a relação Funcionários descreve os pilotos, assim como os outros tipos de funcionários; todo piloto é certificado para alguma aeronave, e somente os pilotos são certificados para voar. Escreva cada uma das seguintes consultas em SQL. (*Consultas adicionais usando o mesmo esquema encontram-se listadas nos exercícios do Capítulo 4.*)

1. Encontre os nomes das aeronaves tais que todos os pilotos certificados para operá-las ganhem mais do que \$ 80.000.
2. Para cada piloto que é certificado para mais do que três aeronaves, encontre o *idFuncion* e a *distLimite* máxima das aeronaves para as quais ele ou ela é certificado.
3. Encontre os nomes dos pilotos cujos salários são menores do que o preço da rota mais barata de Los Angeles a Honolulu.
4. Para todas as aeronaves com *distLimite* maior do que 1.000 milhas, encontre o nome da aeronave e o salário médio de todos os pilotos certificados para essa aeronave.
5. Encontre os nomes dos pilotos certificados para alguma aeronave Boeing.
6. Encontre os *idAeros* de todas as aeronaves que podem ser usadas nas rotas de Los Angeles a Chicago.
7. Identifique as rotas que podem ser pilotadas por todo piloto que ganha mais do que \$ 100.000.
8. Imprima os *nomeFuncions* dos pilotos que podem operar aeronaves com *distLimite* maior do que 3.000 milhas, mas que não são certificados em nenhuma aeronave Boeing.
9. Um cliente deseja viajar de Madison a Nova York com no máximo duas escalas de voo. Liste a escolha de horários de partida de Madison se o cliente quer chegar em Nova York antes de 18 h.
10. Compute a diferença entre o salário médio de um piloto e o salário médio de todos os funcionários (incluindo os pilotos).
11. Imprima o nome e o salário de todo não piloto cujo salário é maior do que o salário médio dos pilotos.
12. Imprima os nomes dos funcionários que são certificados apenas em aeronaves com distância limite maior do que 1.000 milhas.
13. Imprima os nomes dos funcionários que são certificados apenas em aeronaves com distância limite maior do que 1.000 milhas, mas que sejam certificados para, no mínimo, duas aeronaves desse tipo.
14. Imprima os nomes dos funcionários que são certificados apenas em aeronaves com distância limite maior do que 1.000 milhas e que são certificados em alguma aeronave Boeing.

Exercício 5.4 Considere o seguinte esquema relacional. Um funcionário pode trabalhar em mais de um departamento; o campo *pct_tempo* da relação *Trabalha* ilustra a porcentagem de tempo que determinado funcionário trabalha em determinado departamento.

```
Funcion(idFuncion: integer, nomeFuncion: string, idade: integer, salário: real)
Trabalha(idFuncion: integer, idDepto: integer, pct_tempo: integer)
Depto(idDepto: integer, orçamento: real, idGerente: integer)
```

Escreva as seguintes consultas em SQL:

1. Imprima os nomes e as idades de cada funcionário que trabalha em ambos os departamentos de Hardware e de Software.
2. Para cada departamento com mais do que 20 funcionários equivalentes a tempo integral (isto é, onde a soma dos funcionários de tempo parcial e de tempo integral seja no mínimo essa quantidade de funcionários de tempo integral), imprima o *idDepto* juntamente com o número de funcionários que trabalham nesse departamento.
3. Imprima o nome de cada funcionário cujo salário exceda o orçamento de todos os departamentos em que ele ou ela trabalhe.
4. Encontre os *idGerentes* dos gerentes que administram apenas os departamentos com orçamentos maiores do que US\$ 1 milhão.
5. Encontre os *nomeFuncions* dos gerentes que administram os departamentos que possuem os maiores orçamentos.
6. Se um gerente administra mais do que um departamento, ele ou ela *controla* a soma de todos os orçamentos desses departamentos. Encontre os *idGerentes* dos gerentes que controlam mais do que \$ 5 milhões.
7. Encontre os *idGerentes* dos gerentes que controlam as maiores quantias.
8. Encontre os *nomeFuncions* dos gerentes que administram apenas os departamentos com orçamentos maiores do que \$ 1 milhão, mas no mínimo um departamento com orçamento menor do que \$ 5 milhões.

Exercício 5.5 Considere a instância da relação *Marinheiros* ilustrada na Figura 5.22.

1. Escreva consultas em SQL para computar a avaliação média, usando AVG; a soma das avaliações, usando SUM; e o número de avaliações, usando COUNT.

2. Se você dividir a soma recém-calculada pelo número de avaliações, o resultado seria o mesmo do que a média? Como sua resposta se alteraria se essas etapas fossem calculadas com relação ao campo idade no lugar de *avaliação*?
3. Considere a seguinte consulta: *encontre os nomes dos marinheiros com uma avaliação maior do que todos os marinheiros com idade < 21*. As duas consultas em SQL seguintes tentam obter a resposta a esta questão. Ambas computam o mesmo resultado? Se não, explique por quê. Sob quais condições elas computariam o mesmo resultado?

```

SELECT  M.nome-marin
FROM    Marinheiros M
WHERE   NOT EXISTS (SELECT *
                    FROM    Marinheiros M2
                    WHERE   M2.idade < 21
                    AND M.avaliação <= M2.avaliação )

SELECT  *
FROM    Marinheiros M
WHERE   M.avaliação > ANY ( SELECT  M2.avaliação
                           FROM    Marinheiros M2
                           WHERE   M2.idade < 21)

```

4. Considere a instância de Marinheiros ilustrada na Figura 5.22. Vamos definir a instância M1 de Marinheiros consistindo nas duas primeiras tuplas, a instância M2 como as duas últimas tuplas e M como a instância dada.
 - (a) Mostre a junção externa esquerda de M com ela mesma, com a condição de junção *id-marin=id-marin*.
 - (b) Mostre a junção externa direita de M com ela mesma, com a condição de junção *id-marin=id-marin*.
 - (c) Mostre a junção externa completa de M com ela mesma, com a condição de junção *id-marin=id-marin*.
 - (d) Mostre a junção externa esquerda de M1 com M2, com a condição de junção *id-marin=id-marin*.
 - (e) Mostre a junção externa direita de M1 com M2, com a condição de junção *id-marin=id-marin*.
 - (f) Mostre a junção externa completa de M1 com M2, com a condição de junção *id-marin=id-marin*.

Exercício 5.7 Considere o seguinte esquema relacional e responda sucintamente as questões a seguir:

```

Funcion(idFuncion: integer, nomeFuncion: string, idade: integer, salário: real)
Trabalha(idFuncion: integer, idDepto: integer, pct_tempo: integer)
Depto(idDepto: integer, orçamento: real, idGerente: integer)

```

1. Defina uma restrição de tabela em Funcion que assegurará que todo funcionário ganhe no mínimo \$ 10.000.
2. Defina uma restrição de tabela em Depto que assegurará que todos os gerentes tenham *idade* > 30.
3. Defina uma assertiva em Depto que assegurará que todos os gerentes tenham *idade* > 30. Compare essa assertiva com a restrição de tabela equivalente. Explique qual é melhor.
4. Escreva comandos SQL para excluir todas as informações sobre os funcionários cujos salários excedam o salário do gerente de um ou mais departamentos onde eles trabalham. Certifique-se de assegurar que todas as restrições de integridade relevantes sejam satisfeitas após suas atualizações.

Exercício 5.8 Considere as seguintes relações:

Aluno(nroAlun: integer, nomeAlun: string, formação: string, nível: string, idade: integer)

Curso(nome: string, horário: time, sala: string, idProf: integer)

Matriculado(nroAlun: integer, nomeCurso: string)

Professor(idProf: integer, nomeProf: string, idDepto: integer)

O significado dessas relações é direto; por exemplo, Matriculado tem um registro por par aluno-curso tal que o aluno está matriculado no curso.

1. Escreva os comandos SQL exigidos para criar essas relações, incluindo as versões adequadas de todas as restrições de integridade de chave primária e de chave estrangeira.
2. Expresse cada uma das seguintes restrições de integridade em SQL a menos que ela esteja implícita pela restrição de chave primária e chave estrangeira; nesse caso, explique como ela está implícita. Diga se a restrição não pode ser expressa em SQL. Para cada restrição, afirme quais operações (inserções, exclusões e atualizações em relações específicas) devem ser monitoradas para forçar a restrição.
 - (a) Todo curso tem um número mínimo de 5 alunos matriculados e um máximo de 30 alunos.
 - (b) No mínimo um curso usa cada sala.
 - (c) Todo professor deve ministrar no mínimo dois cursos.
 - (d) Apenas professores no departamento com *idDepto=33* ministram mais do que três cursos.
 - (e) Todo aluno deve ser matriculado no curso chamado Math 101.
 - (f) A sala na qual o curso agendado para o horário mais cedo (isto é, o curso com menor valor de *horário*) não deve ser a mesma sala na qual ocorre o curso agendado para o horário mais tarde.
 - (g) Dois cursos não podem ser na mesma sala no mesmo horário.
 - (h) O departamento com a maioria dos professores deve ter menos do que duas vezes o número de professores do departamento com o menor número de professores.
 - (i) Nenhum departamento pode ter mais do que 10 professores.
 - (j) Um aluno não pode acrescentar mais do que dois cursos de uma vez (isto é, em uma única atualização).
 - (k) O número de formados em CS deve ser maior do que o número de formados em Math.
 - (l) O número de cursos distintos nos quais os formados em CS estão matriculados é maior do que o número de cursos distintos nos quais os formados em Math estão matriculados.
 - (m) O total de matriculados dos cursos ministrados pelos professores do departamento com *idDepto=33* é maior do que o número de formados em Math.
 - (n) Deve haver no mínimo um formado em CS se houver algum aluno.
 - (o) Professores de departamentos diferentes não podem ministrar na mesma sala.

Exercício 5.10 Considere o seguinte esquema relacional. Um funcionário pode trabalhar em mais de um departamento; o campo *pct_tempo* da relação *Trabalha* ilustra a porcentagem de tempo que determinado funcionário trabalha em determinado departamento.

```
Funcion(idFuncion: integer, nomeFuncion: string, idade: integer, salário: real)
Trabalha(idFuncion: integer, idDepto: integer, pct_tempo: integer)
Depto(idDepto: integer, orçamento: real, idGerente: integer)
```

Escreva restrições de integridade em SQL-92 (domínio, chave, chave estrangeira ou restrições CHECK; ou assertivas) ou gatilhos do SQL:1999 para assegurar cada um dos requisitos seguintes, considerados independentemente.

1. Os funcionários devem ter um salário mínimo de \$ 1.000.
2. Todo gerente também deve ser um funcionário.
3. A porcentagem total de todo o tempo de dedicação de um funcionário deve ser menor do que 100%.
4. Um gerente deve sempre ter um salário maior do que qualquer funcionário que ele ou ela gerencia.
5. Sempre que um funcionário ganha um aumento, o salário do gerente deve ser aumentado para ser no mínimo igual ao novo salário.
6. Sempre que um funcionário ganha um aumento, o salário do gerente deve ser aumentado para ser no mínimo igual ao novo salário. Além disso, sempre que um funcionário ganha um aumento, o orçamento do departamento deve ser aumentado para ser maior do que a soma dos salários de todos os funcionários do departamento.