

Banco de Dados 1

Prof. André Britto

Exercícios

SQL - Consultas - JOIN

1 - Leitura e pesquisa

A atividade 5 consiste no estudo de comandos de junção em SQL (DML) e na resolução de exercícios

Caso o aluno tenha acesso, a leitura recomendada é capítulo 4 do livro texto da disciplina “Elmasri R., Navathe S. B. Sistemas de Banco de Dados. 6 edição. 2011.”

Caso não tenha acesso ao livro, o aluno pode fazer buscas por material buscando pelos termos:

Tipos de junção

INNER JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

FULL OUTER JOIN

NATURAL

ON <condição>

USING(A1, A2, ..., An)

Fiquem livres para buscar fontes de maior interesse. Pode ser material de outras disciplinas, livros disponíveis de forma gratuita, vídeos no youtube. O importante é que a busca e leitura seja bem completa.

O aluno também pode acompanhar as explicações dos comandos nas notas de aula enviadas para o SIGAA.

2 - Material complementar

Slides de aula anexos ao tópico de aula no SIGAA

3 - Atividade

Questões

1 - Esquema universidade

- 1.1 Liste o nome das disciplinas que possuem turmas. - INNER JOIN
- 1.2 Liste o primeiro nome e sobrenome do estudante e o nome de cada disciplina das turmas que ele está cursando. - INNER JOIN
- 1.3 Liste o primeiro nome e sobrenome do aluno e o primeiro nome e sobrenome de seu orientador. - INNER JOIN
- 1.4 Considerando todas as turmas, liste o primeiro nome e sobrenome do professor, o nome da disciplina que ele está lecionando, além disso liste o primeiro nome e sobrenome de todos os alunos cursando essas turmas. - INNER JOIN
- 1.5 Liste o nome da disciplina e o nome do seu pré-requisito. - INNER JOIN
- 1.6 Liste o primeiro nome e sobrenome do professor e o primeiro nome e sobrenome do seu chefe. - INNER JOIN
- 1.7 Liste o nome da disciplina, turma e primeiro nome e sobrenome do professor, das disciplinas que não são pré-requisito - OUTER JOIN
- 1.8 Liste o primeiro nome e sobrenome de todos os alunos e primeiro nome e sobrenome do orientador, inclusive os alunos sem orientador. - OUTER JOIN
- 1.9 Liste o primeiro nome e sobrenome de todos os professores e o primeiro nome e sobrenome de seus orientandos (alunos que são orientados), inclusive os professores sem orientandos. - OUTER JOIN
- 1.10 Liste o primeiro nome e sobrenome de todos alunos e o primeiro nome e sobrenome dos professores, independente da orientação. - OUTER JOIN
- 1.11 Liste o primeiro nome e sobrenome dos professores que não lecionam turmas. - OUTER JOIN
- 1.12 Liste o nome das disciplinas das turmas que não tem alunos cursando. - OUTER JOIN
- 1.13 Refaça a consulta 1.7 da atividade 4, usando OUTER JOIN - OUTER JOIN
- 1.14 Refaça a consulta 1.8 da atividade 4, usando JOIN - INNER JOIN
- 1.15 Refaça a consulta 1.9 da atividade 4, usando OUTER JOIN - OUTER JOIN

2 - Esquema - hospital/clinica (Há questões para usar o comando LIKE nesse exercício)

- 2.1 Listar o CPF e sobrenome de todos os usuários que possuem a sílaba "sa" em seu sobrenome.
- 2.2 Listar a quantidade de pessoas que possuem sobrenome com a inicial "S".
- 2.3 Listar o nome e CPF de todos os pacientes. - INNER JOIN
- 2.4 Listar a soma de todos os salários dos médicos, porém apenas os que possuem cadastro ativo. - INNER JOIN
- 2.5 Listar o nome e CPF dos pacientes, seguido do nome e CPF dos seus acompanhantes. - INNER JOIN

2.6 Listar o nome e CPF de todos os pacientes que realizaram consulta. - INNER JOIN

2.7 Listar o nome, CPF e número de prontuário de todos os pacientes que realizaram consulta e não houve prescrição de medicamento. - OUTER JOIN

2.8 Listar o nome, CPF, número de prontuário e o nome do medicamento de todos os pacientes que realizaram consulta e houve prescrição de medicamento. - OUTER JOIN

Respostas

-- Questão 1.1

```
SELECT DISTINCT d.nome  
FROM turma t JOIN disciplina d ON (t.cod_disc = d.cod_disc)
```

-- Questão 1.2

```
SELECT u.primeiro_nome, u.sobrenome, d.nome  
FROM turma t JOIN disciplina d ON (t.cod_disc = d.cod_disc)  
      JOIN cursa c ON (c.id_turma = t.id_turma)  
      JOIN estudante e ON (c.mat_estudante = e.mat_estudante)  
      JOIN usuario u ON (e.cpf = u.cpf)
```

-- Questão 1.3

```
SELECT ue.primeiro_nome, ue.sobrenome, up.primeiro_nome, up.sobrenome  
FROM estudante e JOIN plano pl ON (pl.mat_estudante = e.mat_estudante)  
      JOIN usuario ue ON (e.cpf = ue.cpf)  
      JOIN professor p ON (pl.mat_professor = p.mat_professor)  
      JOIN usuario up ON (p.cpf = up.cpf)
```

-- Questão 1.4

```
SELECT *  
FROM turma t JOIN disciplina d ON (t.cod_disc = d.cod_disc)  
      JOIN cursa c ON (c.id_turma = t.id_turma)  
      JOIN estudante e ON (c.mat_estudante = e.mat_estudante)  
      JOIN usuario u ON (e.cpf = u.cpf)  
      JOIN leciona l ON (l.id_turma = t.id_turma)  
      JOIN professor p ON (l.mat_professor = p.mat_professor)  
      JOIN usuario up ON (p.cpf = up.cpf)
```

-- Questão 1.5

```
SELECT d.nome, pr.nome  
FROM disciplina d JOIN disciplina pr ON (pr.pre_req = d.cod_disc)
```

-- Questão 1.6

```
SELECT up.primeiro_nome, up.sobrenome, uc.primeiro_nome, uc.sobrenome
FROM universidade.professor p JOIN universidade.usuario up ON (p.cpf=up.cpf)
      JOIN universidade.departamento d ON (p.departamento = d.cod_depto)
      JOIN universidade.professor pc ON (d.chefe = pc.mat_professor)
      JOIN universidade.usuario uc ON (pc.cpf = uc.cpf)
```

-- Questão 1.7

```
SELECT pr.nome, t.turma, u.primeiro_nome, u.sobrenome
FROM disciplina d RIGHT JOIN disciplina pr ON (d.pre_req = pr.cod_disc)
      JOIN turma t ON (t.cod_disc = pr.cod_disc)
      JOIN leciona l ON (l.id_turma = t.id_turma)
      JOIN professor p ON (l.mat_professor = p.mat_professor)
      JOIN usuario u ON (p.cpf = u.cpf)
WHERE d.cod_disc IS NULL
```

-- Questão 1.8

```
SELECT ue.primeiro_nome, ue.sobrenome, up.primeiro_nome, up.sobrenome
FROM universidade.plano pl JOIN universidade.professor p USING(mat_professor)
      JOIN universidade.usuario up ON (p.cpf=up.cpf)
      RIGHT JOIN (universidade.estudante e
                  JOIN universidade.usuario ue ON (e.cpf = ue.cpf))
USING(mat_estudante)
```

-- Questão 1.9

```
SELECT ue.primeiro_nome, ue.sobrenome, up.primeiro_nome, up.sobrenome
FROM universidade.plano pl JOIN universidade.estudante e USING(mat_estudante)
      JOIN universidade.usuario ue ON (e.cpf = ue.cpf)
      RIGHT JOIN (universidade.professor p
                  JOIN universidade.usuario up ON (p.cpf=up.cpf)) USING(mat_professor)
```

-- Questão 1.10

```
SELECT up.primeiro_nome, up.sobrenome, ue.primeiro_nome, ue.sobrenome
FROM universidade.plano pl RIGHT JOIN( universidade.professor p
JOIN universidade.usuario up ON (p.cpf=up.cpf)) USING(mat_professor)
FULL JOIN (universidade.estudante e
JOIN universidade.usuario ue ON (e.cpf = ue.cpf)) USING(mat_estudante)
```

-- Questão 1.11

```
SELECT up.primeiro_nome, up.sobrenome
FROM universidade.liciona l RIGHT JOIN universidade.professor p USING(mat_professor)
      JOIN universidade.usuario up USING(cpf)
WHERE id_turma IS NULL
```

-- Questão 1.12

```
SELECT d.nome FROM universidade.turma t LEFT JOIN universidade.cursa c
USING(id_turma) JOIN universidade.disciplina d USING(cod_disc) WHERE mat_estudante
IS NULL
```

-- Questão 1.13

```
SELECT DISTINCT mat_professor, chefe
FROM plano pl FULL JOIN departamento d ON (pl.mat_professor = d.chefe)
WHERE id_projeto IS NOT NULL OR chefe IS NOT NULL
```

-- Questão 1.14

```
SELECT DISTINCT mat_professor
FROM plano pl JOIN departamento d ON (pl.mat_professor = d.chefe)
```

-- Questão 1.15

```
SELECT DISTINCT mat_professor
FROM plano pl LEFT JOIN departamento d ON (pl.mat_professor = d.chefe)
WHERE chefe IS NULL
```

-- Questão 2.1

```
SELECT cpf,  
       sobrenome  
FROM   hospital.usuario  
WHERE  sobrenome LIKE '%sa%';
```

-- Questão 2.2

```
SELECT count(sobrenome)  
FROM   hospital.usuario  
WHERE  sobrenome LIKE 'S%';
```

-- Questão 2.3

```
SELECT pe.primeironome,  
       pe.sobrenome,  
       pe.cpf  
FROM   hospital.usuario AS pe natural  
JOIN   hospital.paciente
```

-- Questão 2.4

```
SELECT sum(me.salario)  
FROM   hospital.medico AS me natural  
JOIN   hospital.usuario AS pe natural  
JOIN   hospital.perfil AS per  
WHERE  per.ativo='S'
```

-- Questão 2.5

```
SELECT   pe.cpf,  
         pe.primeironome,  
         pe.sobrenome,  
         pea.cpf,  
         pea.primeironome,  
         pea.sobrenome  
FROM     hospital.paciente AS pa  
INNER JOIN hospital.usuario AS pe
```

```
ON      (pa.cpf = pe.cpf)
INNER JOIN hospital.acompanhante AS ac
ON      (ac.cpf = pa.cpfacomp)
INNER JOIN hospital.usuario AS pea
ON      (pea.cpf = ac.cpf)
```

-- Questão 2.6

```
SELECT DISTINCT pe.primeironome,
                pe.sobrenome,
                pe.cpf
FROM      hospital.usuario pe
INNER JOIN hospital.paciente pa
ON        (pa.cpf = pe.cpf)
INNER JOIN hospital.consulta co
ON        (co.numprontuario = pa.numprontuario)
```

-- Questão 2.7

```
SELECT      pe.primeironome,
            pe.sobrenome,
            pe.cpf,
            pa.numprontuario
FROM        hospital.paciente pa
INNER JOIN  hospital.consulta co
ON          (co.numprontuario = pa.numprontuario)
LEFT OUTER JOIN hospital.prescricao AS pr
ON          (pr.numprontuario = co.numprontuario)
INNER JOIN  hospital.usuario pe
ON          (pe.cpf = pa.cpf)
WHERE       pr.idconsultaprescricao IS NULL
```

-- Questão 2.8

```
SELECT      pe.primeironome,
            pe.sobrenome,
            pe.cpf,
            pa.numprontuario,
            me.nome
FROM        hospital.paciente pa
INNER JOIN  hospital.consulta co
```



```
ON          (co.numprontuario = pa.numprontuario)
LEFT OUTER JOIN hospital.prescricao AS pr
ON          (pr.numprontuario = co.numprontuario)
INNER JOIN   hospital.medicamento me
ON          (pr.idmedicamento = me.idmedicamento)
INNER JOIN   hospital.usuario pe
ON          (pe.cpf = pa.cpf)
WHERE       (
            pr.idconsultaprescricao IS NOT NULL
            AND pr.idmedicamento IS NOT NULL
            AND pr.numprontuario IS NOT NULL)
```