

Lista de Exercícios Vd

CONSULTAS WITH QUERIES e VISÕES em SQL no BD SEE

João Batista de Sousa Paula, 11911BCC008

John Vitor da Silva Cunha, 11821BCC005

Laís Saloum Deghaide, 11821BCC001

Vinícius Calixto Rocha, 11911BCC039

Yan Stivaletti e Souza, 11821BCC002

1.

a) CREATE VIEW nadadores

AS SELECT nomepes

FROM pessoa AS p, atleta AS a, inscricao AS i, categoria AS c, modalidade AS m

WHERE p.idpes=a.idpes

AND a.idcompetidor=i.idcompetidor

AND i.idcat=c.idcat

AND c.idmod=m.idmod

AND nomemod='Natacao'

SELECT * FROM nadadores

"nomepes"

"Adilson Pires"

"Artur A Moura"

"Cesar W Alvarenga"

"Claudia Araujo"

"Daniel N Guardieiro"

"Jaqueline A Papini"

b) CREATE VIEW participantes_modalidade1 AS

(SELECT nomemod, nomecat, nomepes AS nomeparticipante

FROM modalidade AS m, categoria AS c, inscricao AS i, atleta AS a, pessoa AS p

WHERE m.idmod=c.idmod

AND c.idcat=i.idcat

AND i.idcompetidor=a.idcompetidor

AND a.idpes=p.idpes)

UNION

(SELECT nomemod, nomecat, nomeequ AS nomeparticipante

FROM modalidade AS m, categoria AS c, inscricao AS i, equipe AS e

WHERE m.idmod=c.idmod

AND c.idcat=i.idcat

AND i.idcompetidor=e.idcompetidor)

```
SELECT * FROM participantes_modalidade1
```

```
"nomemod","nomecat","nomeparticipante"
"Atletismo","100mt rasos","Adilson Pires"
"Atletismo","100mt rasos","Artur A Moura"
"Atletismo","100mt rasos","Cesar W Alvarenga"
"Atletismo","100mt rasos","Claudia Araujo"
"Atletismo","100mt rasos","Daniel N Guardieiro"
"Atletismo","100mt rasos","Danilo Ribeiro"
"Atletismo","100mt rasos","Diego C Bessa"
"Atletismo","100mt rasos","Diego M Oliveira"
"Atletismo","100mt rasos","Diogo Mendes"
"Atletismo","100mt rasos","Edislei C Reis"
"Atletismo","400mt barreiras","Eduardo I Sa"
"Atletismo","400mt barreiras","Eli R Silva"
"Atletismo","400mt barreiras","Everson F Giacomelli Jr."
"Atletismo","400mt barreiras","Fabiano S Terra"
"Atletismo","400mt barreiras","Fabricio S Araujo"
"Atletismo","400mt barreiras","Gabriel"
"Atletismo","400mt barreiras","Gil Victor"
"Atletismo","400mt barreiras","Guilherme H G Silva"
"Atletismo","400mt barreiras","Helder Linhare"
"Atletismo","400mt barreiras","Humberto E Cruz Jr."
"Futebol","Masculino","Fut periodo 1"
"Futebol","Masculino","Fut periodo 2"
"Futebol","Masculino","Fut periodo 3"
"Futebol","Masculino","Fut periodo 4"
"Futebol","Masculino","Fut periodo 5"
"Futebol","Masculino","Fut periodo 6"
"Futebol","Masculino","Fut periodo 7"
"Futebol","Masculino","Fut periodo 8"
"Futebol","Masculino","Fut Professores"
"Natacao","100mt Borboleta","Daniel N Guardieiro"
"Natacao","100mt Costas","Cesar W Alvarenga"
"Natacao","100mt Peito","Artur A Moura"
"Natacao","100mt Peito","Jaqueline A Papini"
"Natacao","50mt Livre","Adilson Pires"
"Natacao","50mt Livre","Claudia Araujo"
"Volei","Feminino","Volei time 1"
"Volei","Feminino","Volei time 2"
```

- c) CREATE OR REPLACE VIEW patrocínios_evento (SomaPat) AS
(SELECT SUM(valorpat)

```
FROM see.patevento);
```

```
SELECT SomaPat  
FROM patrocinios_evento
```

```
"somapat"  
15000.00
```

```
d) CREATE OR REPLACE VIEW patrocinios_atletas1(nomepes,QuantPat) AS  
    (SELECT nomepes, COUNT(idemp)  
        FROM see.pessoa NATURAL JOIN see.atleta,see.patrocínio  
        NATURAL JOIN see.patcompetidor  
        WHERE atleta.idcompetidor = patrocínio.idcompetidor  
        GROUP BY nomepes  
    );
```

```
"nomepes","quantpat"  
"Diogo Mendes",1  
"Edislei C Reis",1  
"Daniel N Guardieiro",1  
"Diego M Oliveira",1  
"Diego C Bessa",1  
"Eduardo I Sa",1  
"Danilo Ribeiro",1  
"Adilson Pires",1  
"Claudia Araujo",1  
"Cesar W Alvarenga",1  
"Artur A Moura",1
```

```
e)  
CREATE VIEW campeoes_evento AS SELECT nomepes as nomecampeao,  
nomemod, nomecat  
FROM (categoria NATURAL JOIN inscricao)  
    JOIN competicao ON fasecom='F' AND  
categoria.idcat=competicao.idcat  
    JOIN pessoa ON inscricao.idcompetidor=pessoa.idpes  
    JOIN modalidade ON categoria.idmod=modalidade.idmod  
    JOIN participacao pa1 ON  
pa1.idcompeticao=competicao.idcompeticao  
    AND pa1.idcompetidor=inscricao.idcompetidor  
WHERE ((pa1.unidade IN ('metros','centimetros','milimetros',  
    'pontos','sets','gols'))
```

```

        AND pa1.escor > ALL (SELECT escor FROM participacao
pa2 WHERE pa1.idcompeticao=pa2.idcompeticao AND pa1.idcompetidor !=
pa2.idcompetidor)
    )

```

```

    OR
    (pa1.unidade IN ('segundos', 'milissegundos')
    AND pa1.escor < ALL (SELECT escor FROM participacao pa3
        WHERE pa1.idcompeticao=pa3.idcompeticao
        AND pa1.idcompetidor!=pa3.idcompetidor)
    )

```

```

    )
    )
UNION
SELECT nomeequ as nomecampeao, nomemod, nomecat
FROM (categoria NATURAL JOIN inscricao)
    JOIN competicao ON fasecom='F' AND
categoria.idcat=competicao.idcat
    JOIN equipe ON inscricao.idcompetidor=equipe.idequ
    JOIN modalidade ON categoria.idmod=modalidade.idmod
    JOIN participacao pa1 ON

```

```

pa1.idcompeticao=competicao.idcompeticao
    AND pa1.idcompetidor=inscricao.idcompetidor
WHERE ((pa1.unidade IN ('metros','centimetros','milimetros',
    'pontos','sets','gols')
    AND pa1.escor > ALL (SELECT escor FROM participacao
pa2 WHERE pa1.idcompeticao=pa2.idcompeticao AND pa1.idcompetidor !=
pa2.idcompetidor)
    )

```

```

    OR
    (pa1.unidade IN ('segundos', 'milissegundos')
    AND pa1.escor < ALL (SELECT escor FROM participacao pa3
        WHERE pa1.idcompeticao=pa3.idcompeticao AND
        pa1.idcompetidor!=pa3.idcompetidor)
    )

```

```

    )
    )

```

```

"nomecampeao","nomemod","nomecat"
"Artur A Moura","Atletismo","100mt rasos"
"Eli R Silva","Atletismo","400mt barreiras"
"Fut periodo 1","Atletismo","100mt rasos"
"Natacao Comp. masculino","Atletismo","400mt barreiras"

```

2.

Disponibilidade:

```
CREATE VIEW atletas AS SELECT nomepes, datanas, sexo, nomecat,  
nomemod  
FROM pessoa, atleta, categoria, modalidade, competidor, inscricao  
WHERE pessoa.idpes=atleta.idpes  
AND competidor.idcompetidor=inscricao.idcompetidor  
AND inscricao.idcat=categoria.idcat  
AND categoria.idmod=modalidade.idmod  
AND atleta.idcompetidor=competidor.idcompetidor
```

A visão acima permite ao usuário consultar com um comando simples (SELECT * FROM atletas) o nome, a data de nascimento, o sexo, o nome da categoria que o atleta participa e sua modalidade, simplificando e centralizando essa consulta o que atende aos requisitos de disponibilidade da visão.

Confidencialidade:

```
CREATE VIEW pessoas AS SELECT nomepes, datanas, sexo FROM pessoa
```

A visão acima restringe ao usuário da visão o acesso a alguns atributos confidenciais das pessoas cadastradas como seu id pessoal e seu cpf, retornando apenas os nomes, as datas de nascimento e os sexos das pessoas.

Integridade:

```
CREATE VIEW atletas_equipas AS SELECT nomepes, nomeequ  
FROM pessoa, atleta, atlequ, equipe  
WHERE pessoa.idpes=atleta.idpes  
AND atleta.idpes=atlequ.idpes  
AND equipe.idequ=atlequ.idequ
```

A visão acima exibe os atletas e suas respectivas equipes e por não possuir nenhuma chave primária não se pode fazer alterações (facilmente) nas tabelas através desta visão, garantindo assim a integridade das tabelas que estão na visão.

3. 1º restrição de junção

```
CREATE VIEW emphours AS (SELECT lname, essn, hours FROM employee JOIN  
works_on ON ssn=essn) UPDATE emphours SET lname = 'Silva' WHERE hours >  
20;
```

OBS: caso a atualização fosse aplicada, seria no lname da tabela employee, mas o mesmo empregado poderia trabalhar menos de 20 horas em outro projeto, portanto esta visão não pode ser atualizável

2º restrição Função agregação

CREATE VIEW empsumhours AS (SELECT essn, sum(hours) FROM works_on GROUP BY essn) OBS: não há correspondência direta da soma de horas com um atributo da tabela base, portanto esta visão não pode ser atualizável

3º Restrições para visões atualizáveis Subconsultas com tabelas da cláusula FROM
CREATE VIEW supervisors AS (SELECT ssn, lname, superssn FROM employee WHERE ssn IN (SELECT superssn FROM employee)) UPDATE supervisors SET superssn = '123456789' WHERE superssn = '987654321']

OBS: superssn = '123456789' inicialmente pode não fazer da visão, então haveria uma inserção? E os outros atributos? portanto esta visão não pode ser atualizável

4º Restrições para visões atualizáveis - DISTINCT

CREATE VIEW empname AS (SELECT DISTINCT lname FROM employee)
UPDATE empname SET lname='S.' WHERE lname = 'Silva'

OBS: cada tupla de empname pode corresponder a várias tuplas de employee, então não há correspondência direta de um atributo da visão com um atributo da tabela base, portanto esta visão não pode ser atualizável

CREATE VIEW seg(ssn, name, sex) AS (SELECT ssn, fname, sex FROM employee) UNION (SELECT essn, name, sex FROM dependent) WITH CHECK OPTION; INSERT INTO seg ('123456789', 'Jose', 'M');

4.

a)

WITH camp_comp (idcat,idcompeticao, fasecom, nomecompetidor, escore, unidade) AS
((SELECT idcat,idcompeticao,fasecom, nomepes AS nomecompetidor, escore, unidade

```

        FROM categoria NATURAL JOIN competicao NATURAL JOIN
participacao NATURAL JOIN competidor NATURAL JOIN atleta NATURAL JOIN
pessoa
    )
UNION
(SELECT idcat,idcompeticao,fasecom, nomequ AS nomecompetidor, escore, unidade
    FROM categoria NATURAL JOIN competicao NATURAL JOIN
participacao NATURAL JOIN competidor NATURAL JOIN equipe
    )
)

SELECT nomemod,nomecat,fasecom,nomecompetidor,escore,unidade FROM
    (SELECT nomemod,nomecat,fasecom,nomecompetidor,escore,unidade,(-1*escore)
AS escoreaux
        FROM modalidade NATURAL JOIN categoria NATURAL JOIN
camp_comp
        WHERE unidade IN ('metros','centimetros','milimetros','pontos','sets','gols')
    UNION
    (SELECTnomemod,nomecat,fasecom,nomecompetidor,escore,unidade, escore AS
escoreaux
        FROM modalidade NATURAL JOIN categoria NATURAL JOIN
camp_comp
        WHERE unidade IN ('segundos','milisegundos')
    )
ORDER BY escoreaux
) AS table_aux

```

```

"nomemod","nomecat","fasecom","nomecompetidor","escore","unidade"
"Volei","Feminino","F","Volei time 1","3","sets"
"Volei","Feminino","F","Volei time 2","2","sets"
"Atletismo","100mt rasos","F","Artur A Moura","11.6","segundos"
"Atletismo","100mt rasos","0","Adilson Pires","12","segundos"
"Atletismo","100mt rasos","F","Adilson Pires","12","segundos"
"Atletismo","100mt rasos","1","Adilson Pires","12.5","segundos"
"Atletismo","100mt rasos","1","Artur A Moura","12.6","segundos"
"Atletismo","100mt rasos","1","Cesar W Alvarenga","12.7","segundos"
"Atletismo","100mt rasos","1","Diogo Mendes","12.8","segundos"
"Atletismo","100mt rasos","0","Artur A Moura","13","segundos"
"Atletismo","100mt rasos","0","Cesar W Alvarenga","14","segundos"
"Atletismo","100mt rasos","0","Diogo Mendes","15","segundos"
"Atletismo","100mt rasos","0","Daniel N Guardieiro","16","segundos"
"Atletismo","100mt rasos","0","Danilo Ribeiro","17","segundos"
"Atletismo","100mt rasos","0","Diego C Bessa","17.2","segundos"
"Atletismo","100mt rasos","0","Diego M Oliveira","17.3","segundos"

```

"Atletismo"	"400mt barreiras"	"F"	"Eli R Silva"	"51"	"segundos"	"
"Atletismo"	"400mt barreiras"	"F"	"Eduardo I Sa"	"52"	"segundos"	"
"Atletismo"	"400mt barreiras"	"1"	"Eduardo I Sa"	"54"	"segundos"	"
"Atletismo"	"400mt barreiras"	"1"	"Eli R Silva"	"55"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Eduardo I Sa"	"55"	"segundos"	"
"Atletismo"	"400mt barreiras"	"1"	"Everson F Giacomelli Jr."	"56"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Eli R Silva"	"56"	"segundos"	"
"Atletismo"	"400mt barreiras"	"1"	"Fabiano S Terra"	"57"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Everson F Giacomelli Jr."	"57"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Fabiano S Terra"	"58"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Fabricio S Araujo"	"59"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Gabriel"	"60"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Gil Victor"	"60.1"	"segundos"	"
"Atletismo"	"400mt barreiras"	"0"	"Guilherme H G Silva"	"60.2"	"segundos"	"

b)

```

WITH RECURSIVE arbcoo(nivel,idcoo,idpes) AS (
    SELECT 1,raiz.idcoo,raiz.idpes
    FROM arbitro raiz
    WHERE raiz.idpes = '5'
    UNION ALL
    SELECT pai.nivel+1,filho.idcoo,filho.idpes
    FROM arbcoo pai, arbitro filho
    WHERE filho.idcoo = pai.idpes
    AND filho.idpes <> '5'
)
SELECT DISTINCT nivel,idcoo,idpes
FROM arbcoo
ORDER BY nivel,idcoo,idpes
"nivel","idcoo","idpes"
1,5,5
2,5,6
2,5,7
3,6,8
3,6,9

```

c) WITH RECURSIVE arbcoo(nivel,idcoo,idpes) AS (

```

SELECT 0,raiz.idcoo,raiz.idpes
FROM arbitro raiz
WHERE raiz.idpes = raiz.idcoo

```



```
UNION ALL
SELECT pai.nivel+1,filho.idcoo,filho.idpes
      FROM arbcoo pai, arbitro filho
      WHERE filho.idcoo = pai.idpes
            AND filho.idpes <> pai.idpes
    )
SELECT DISTINCT nivel,idcoo,idpes
      FROM arbcoo
      ORDER BY nivel,idcoo,idpes
```

"nivel","idcoo","idpes"

0,5,5

1,5,6

1,5,7

2,6,8

2,6,9