



Lista de Exercícios – Álgebra Relacional e SQL

1. Tendo como base de dados o esquema de banco de dados abaixo, especifique as consultas solicitadas usando operadores da álgebra relacional e comandos SQL.

EMPREGADO

PNOME	UNAME	<u>SSN</u>	DNASC	ENDER	SEXO	SALARIO	SUPERSSN	DNO
-------	-------	------------	-------	-------	------	---------	----------	-----

DEPARTAMENTO

<u>DNOME</u>	<u>DNUMERO</u>	GERSSN	GERDATAINICIO
--------------	----------------	--------	---------------

DEPTO_LOCALIZACOES

<u>DNUMERO</u>	<u>DLOCALIZACAO</u>
----------------	---------------------

PROJETO

PJNOME	<u>PNUMERO</u>	PLOCALIZACAO	DNUM
--------	----------------	--------------	------

TRABALHA_EM

<u>ESSN</u>	<u>PNO</u>	HORAS
-------------	------------	-------

DEPENDENTE

<u>ESSN</u>	<u>NOME_DEPENDENTE</u>	SEXO	DATANASC	PARENTESCO
-------------	------------------------	------	----------	------------

- a) Recupere os nomes e endereços de todos os empregados do departamento 5 que trabalham mais de dez horas por semana no projeto ‘Produto X’.

$\Pi_{PNOME, UNOME, ENDER} (\sigma_{PJNOME = 'PRODUTO X' \wedge HORAS > 10 \wedge DNO = 5}$
 $((EMPREGADO \bowtie_{SSN=ESSN} TRABALHA_EM) \bowtie_{PNO=PNUMERO} PROJETO))$

SELECT PNOME, UNOME, ENDER
FROM ((EMPREGADO **JOIN** TRABALHA_EM **ON** SSN=ESSN)
 JOIN PROJETO **ON** PNO=PNUMERO)
WHERE PJNOME = 'PRODUTO X' ^ HORAS > 10 ^ DNO = 5

- b) Encontre os nomes de todos os empregados que são diretamente supervisionados por ‘João da Silva’ (SUPERSSN = 123456).

$\Pi_{PNOME, UNOME} (\sigma_{SUPERSSN = 123456} (EMPREGADO))$

SELECT PNOME, UNOME
FROM EMPREGAD
WHERE SUPERSSN = 123456



- c) Para cada empregado, recupere o primeiro e o último nome do empregado e o primeiro e o último nome de seu supervisor imediato.

$E1 \leftarrow \text{EMPREGADO}$

$E2 \leftarrow \text{EMPREGADO}$

$\Pi_{E1.PNOME, E1.UNOME, E2.PNOME, E2.UNOME} (E1 \bowtie_{E1.SUPERSSN=E2.SSN} E2)$

SELECT E1.PNOME, E1.UNOME, E2.PNOME, E2.UNOME
FROM EMPREGADO **AS** E1, EMPREGADO **AS** E2
WHERE E1.SUPERSSN = E2.SSN

- d) Para cada projeto, recupere o número do projeto, o nome do projeto e o número de empregados do departamento 5 que trabalha no projeto.

$R1 \leftarrow \sigma_{DNO=5} (\text{EMPREGADO})$

$R2 \leftarrow \Pi_{PNUMERO, PJNOME, ESSN} ((R1 \bowtie_{SSN=ESSN} \text{TRABALHA_EM})$
 $\quad \bowtie_{PNO=PNUMERO} \text{PROJETO}))$

PNUMERO, PJNOME **G** **COUNT(ESSN)** (**R2**)

SELECT PNO, PJNOME, COUNT(ESSN)
FROM TRABALHA_EM **JOIN** PROJETO **ON** PNO = PNUMERO
GROUP BY PNO

- e) Para cada projeto, liste o nome do projeto e o total de horas por semana (de todos os empregados) gastas no projeto.

PJNOME **G** **SUM(HORAS)** (**TRABALHA_EM** $\bowtie_{PNO=PNUMERO}$ **PROJETO**)

SELECT PJNOME, SUM(HORAS)
FROM TRABALHA_EM **JOIN** PROJETO **ON** PNO = PNUMERO
GROUP BY PJNOME

- f) Recupere os nomes de todos os empregados que trabalhem em todos os projetos.

$P \leftarrow \Pi_{PNUMERO} (\text{PROJETO})$

$T \leftarrow \Pi_{ESSN, PNO} (\text{TRABALHA_EM})$

$\Pi_{PNOME, UNOME} (\text{EMPREGADO} \bowtie_{SSN=ESSN} (T \div P))$

Ou



$$\Pi_{\text{PNOME, UNOME}} (\text{EMPREGADO} \bowtie_{\text{SSN=ESSN}} (\Pi_{\text{ESSN, PNO}} (\text{TRABALHA_EM}) \div \Pi_{\text{PNUMERO}} (\text{PROJETO})))$$

```
SELECT PNOME, UNOME
FROM EMPREGADO
WHERE NOT EXISTS
  ((SELECT PNUMERO FROM PROJETO)
  EXCEPT
  (SELECT PNO FROM TRABALHA_EM WHERE SSN=ESSN));
```

Observação:

A primeira subconsulta (que não está correlacionada) seleciona todos os projetos e a segunda (que está correlacionada) seleciona todos os projetos nos quais o empregado em particular, que está sendo considerado, trabalhe. Se a diferença dos conjuntos, o da primeira subconsulta MINUS (EXCEPT) e o da segunda, for vazio, isso significa que o empregado trabalha em todos os projetos e então é selecionado. Lembre-se de que EXCEPT é o operador da diferença entre os conjuntos.

g) Recupere os nomes de todos os empregados que não trabalham em nenhum projeto.

$$E \leftarrow \Pi_{\text{SSN}} (\text{EMPREGADO})$$

$$T \leftarrow \Pi_{\text{ESSN}} (\text{TRABALHA_EM})$$

$$R \leftarrow E - T$$

$$\Pi_{\text{PNOME, UNOME}} (\text{EMPREGADO} \bowtie_{\text{EMPREGADO.SSN=R.SSN}} (R))$$

Ou

$$\Pi_{\text{PNOME, UNOME}} (\text{EMPREGADO} \bowtie_{\text{EMPREGADO.SSN=E.SSN}} (E - T))$$

```
SELECT PNOME, UNOME
FROM EMPREGADO
WHERE SSN NOT IN
  (SELECT ESSN FROM TRABALHA_EM);
```

h) Faça uma lista de todos os números de projetos para aqueles projetos que envolvem um empregado cujo último nome é 'Smith'.

$$\Pi_{\text{PNUMERO}} (\sigma_{\text{UNOME} = \text{'SMITH'}} ((\text{EMPREGADO} \bowtie_{\text{SSN=ESSN}} \text{TRABALHA_EM}) \bowtie_{\text{PNO=PNUMERO}} \text{PROJETO}))$$

```
SELECT PNUMERO
FROM EMPREGADO JOIN TRABALHA_EM ON SSN=ESSN
WHERE UNOME = `Smith`
```



- i) Para cada departamento, recupere o nome do departamento e a média salarial de todos os empregados que trabalham nesse departamento.

$\Pi_{\text{DNOME}} \text{AVG}(\text{SALARIO}) (\text{EMPREGADO} \bowtie_{\text{DNO=DNUMERO}} \text{DEPARTAMENTO})$

```
SELECT DNOME, AVG(SALARIO)
FROM EMPREGADO JOIN DEPARTAMENTO ON DNO = DNUMERO
GROUP BY DNOME
```

- j) Recupere a média salarial de todos os empregados do sexo feminino.

$\Pi_{\text{AVG}(\text{SALARIO})} (\sigma_{\text{SEXO} = 'F'} (\text{EMPREGADO}))$

```
SELECT AVG(SALARIO)
FROM EMPREGADO
WHERE SEXO = 'F'
```

- k) Encontre os nomes e os endereços de todos os empregados que trabalhem em pelo menos um projeto localizado em Maringá, mas cujo departamento não se localiza em Maringá.

$\Pi_{\text{PNOME, UNOME, ENDER}} (\sigma_{\text{PLOCALIZACAO} = \text{'Maringá'} \wedge \text{DLOCALIZACAO} \neq \text{'Maringá'}} ((\text{EMPREGADO} \bowtie_{\text{DNO=DNUM}} \text{DEPARTAMENTO}) \bowtie_{\text{SSN=ESSN}} \text{TRABALHA_EM}) \bowtie_{\text{PNO=PNUMERO}} \text{PROJETO}))$

Ou

$\Pi_{\text{PNOME, UNOME, ENDER}} (((\text{EMPREGADO} \bowtie_{\text{DNO=DNUM}} (\sigma_{\text{DLOCALIZACAO} \neq \text{'Maringá'}} (\text{DEPARTAMENTO}))) \bowtie_{\text{SSN=ESSN}} \text{TRABALHA_EM}) \bowtie_{\text{PNO=PNUMERO}} (\sigma_{\text{PLOCALIZACAO} = \text{'Maringá'}} (\text{PROJETO})))$

```
SELECT PNOME, UNOME
FROM EMPREGADO
WHERE SSN IN
  (SELECT ESSN FROM TRABALHA_EM, DEPARTAMENTO, PROJETO
   WHERE SSN = ESSN AND PNO = PNUMERO AND
     DLOCALIZACAO NOT = 'Maringá' AND PLOCALIZACAO = 'Maringá');
```



UNIVERSIDADE ESTADUAL DE MARINGÁ

Departamento de Informática

Disciplina: Banco de Dados I – 6892/01

Profa. Dra. Maria Madalena Dias

- 1) Liste os últimos nomes de todos os gerentes de departamento que não tenham dependentes.

$DEPTO \leftarrow \Pi_{GERSSN}(DEPARTAMENTO)$

$DEPEN \leftarrow \Pi_{ESSN}(DEPENDENTE)$

$R \leftarrow DEPTO - DEPEN$

$\Pi_{UNOME}(EMPREGADO \bowtie_{SSN=GERSSN}(R))$

Ou

$\Pi_{UNOME}(EMPREGADO \bowtie_{SSN=GERSSN}(DEPTO - DEPEN))$

SELECT UNOME

FROM EMPREGADO

WHERE SSN NOT IN

**(SELECT GERSSN FROM DEPARTAMENTO JOIN DEPENDENTE ON
GERSSN = ESSN);**

2. Tendo como base de dados o esquema de banco de dados abaixo, especifique as consultas solicitadas usando operadores da álgebra relacional e comandos SQL.

AEROPORTO

<u>CODIGO_AEROPORTO</u>	NOME	CIDADE	ESTADO
-------------------------	------	--------	--------

VOO

<u>NUMERO</u>	COMPANHIA_AEREA	DIA_SEMANA	ID_AVIAO
---------------	-----------------	------------	----------

TRECHO_VOO

<u>NO_VOO</u>	<u>NUM_TRECHO</u>	CODIGO_AER_PARTIDA	HORA_PARTIDA_PROGRAMADA	CODIGO_AER_CHEGADA	HORA_CHEGA_PROGRAMADA
---------------	-------------------	--------------------	-------------------------	--------------------	-----------------------

PASSAGEM

<u>NO_VOO</u>	CODIGO_PASSAGEM	VALOR	RESTRICOES
---------------	-----------------	-------	------------

TIPO_AVIAO

<u>NOME_TIPO</u>	MAX_POLTRONAS	EMPRESA
------------------	---------------	---------

PODE_POUSAR

<u>NOME TIPO AVIAO</u>	<u>CODIGO_AEROPORTO</u>
------------------------	-------------------------



UNIVERSIDADE ESTADUAL DE MARINGÁ

Departamento de Informática

Disciplina: Banco de Dados I – 6892/01

Profa. Dra. Maria Madalena Dias

AVIAO

ID_AVIAO	NUMERO_TOTAL_POLTRONAS	TIPO_AVIAO
----------	------------------------	------------

RESERVA_POLTRONA

NO_VOO	NUM_TRECHO	DATA	NUM_POLTRONA	NOME_CLIENTE	FONE_CLIENTE
--------	------------	------	--------------	--------------	--------------

- a) Para cada voo, liste o número do voo, o aeroporto de chegada para o primeiro trecho de voo e o aeroporto de chegada para o último trecho de voo.

$GV(N_V, QT_TR) \leftarrow NO_VOO \text{ GROUP COUNT NUM_TRECHO (TRECHO_VOO)}$

$A1 \leftarrow \Pi_{NO_VOO, AER_CHEGADA} (\sigma_{NUM_TRECHO = 1} (TRECHO_VOO))$

$A2 \leftarrow \Pi_{NO_VOO, AER_CHEGADA} (\sigma_{NUM_TRECHO = QT_TR} (TRECHO_VOO \bowtie_{NO_VOO = N_V \wedge NUM_TRECHO = QT_TR} GV))$

$\Pi_{A1.NO_VOO, A1.AER_CHEGADA, A2.AER_CHEGADA} (A1 \bowtie_{A1.NO_VOO = A2.NO_VOO} A2)$

CREATE VIEW ULT_TRECHO AS

SELECT NO_VOO, COUNT(NUM_TRECHO) **AS** ULT_TR
FROM TRECHO_VOO
GROUP BY NO_VOO;

SELECT A1.NO_VOO, A1.AER_CHEGADA, A2.AER_CHEGADA
FROM ((**SELECT** A1.NO_VOO, A1.AER_CHEGADA
 FROM TRECHO_VOO **AS** A1 **WHERE** NUM_TRECHO = 1)
JOIN
 (**SELECT** A2.NO_VOO, A2.AER_CHEGADA
 FROM TRECHO_VOO **AS** A2, ULT_TRECHO
 WHERE A2.NO_VOO = ULT_TRECHO.NO_VOO
 AND A2.NUM_TRECHO = ULT_TRECHO.ULT_TR)
ON A1.NO_VOO = A2.NO_VOO);

- b) Liste os números dos voos e os dias da semana de todos os voos ou trechos de voo que partam do Aeroporto Internacional de São Paulo (código do aeroporto 'CGA') e cheguem ao Aeroporto Internacional de Recife (código do aeroporto 'GUA').

$\Pi_{NO_VOO, DIA_SEMANA} (\sigma_{AER_PARTIDA='CGA' \wedge AER_CHEGADA='GUA'} (TRECHO_VOO \bowtie_{NO_VOO=NUMERO_VOO}))$



```
SELECT NO_VOO, DIA_SEMANA
FROM TRECHO_VOO JOIN VOO ON NO_VOO = NUMERO
WHERE AER_PARTIDA = 'CGA' AND AER_CHEGADA = 'GUA'
```

- c) Liste o número do voo, o código do aeroporto de partida, o horário programado para a partida, o código do aeroporto de chegada, o horário programado para a chegada e os dias da semana de todos os voos ou trechos de voo que partam de algum dos aeroportos da cidade de São Paulo e cheguem em algum dos aeroportos da cidade de Recife.

$AP \leftarrow \Pi_{NO_VOO, AER_PARTIDA, HORA_PARTIDA_PROGRAMADA}$

$(\sigma_{CIDADE='Sao Paulo'} (TRECHO_VOO \bowtie_{AER_PARTIDA=CODIGO_AEROPORTO} AEROPORTO))$

$AC \leftarrow \Pi_{NO_VOO, AER_CHEGADA, HORA_CHEGA_PROGRAMADA}$

$(\sigma_{CIDADE='Recife'} (TRECHO_VOO \bowtie_{AER_CHEGADA=CODIGO_AEROPORTO} AEROPORTO))$

$\Pi_{AP.NO_VOO, AER_PARTIDA, HORA_PARTIDA_PROGRAMADA, AER_CHEGADA, HORA_CHEGA_PROGRAMADA, DIA_SEMANA} ((AP \bowtie_{AP.NO_VOO=AC.NO_VOO} AC) \bowtie_{AP.NO_VOO=VOO.NO_VOO} VOO)$

```
SELECT AP.NO_VOO, AP.AER_PARTIDA, AP.HORA_PARTIDA,
        AC.AER_CHEGADA, AC.HORA_CHEGA_PROGRAMADA,
        VOO_DIA_SEMANA
FROM (((SELECT AP.NO_VOO, AP.AER_PARTIDA,
        AP.HORA_PARTIDA_PROGRAMADA
FROM TRECHO_VOO AS AP JOIN AEROPORTO
        ON AP.AER_PARTIDA = CODIGO_AEROPORTO
WHERE CIDADE = 'Sao Paulo')
JOIN
(SELECT AC.NO_VOO, AC.AER_CHEGADA,
        AC.HORA_CHEGA_PROGRAMADA
FROM TRECHO_VOO AS AC JOIN AEROPORTO
        ON AER_CHEGADA = CODIGO_AEROPORTO
WHERE CIDADE = 'Recife')
ON AP.NO_VOO = AC.NO_VOO))
JOIN VOO ON AP.NO_VOO = VOO.NO_VOO)
```



d) Liste todas as informações dos passageiros do voo de número 'VO197'.

$\Pi_{\text{NUM_POLTRONA, NOME_CLIENTE, FONE_CLIENTE}} (\sigma_{\text{NO_VOO}='VO197'} (\text{RESERVA_POLTRONA}))$

e) Recupere o número de poltronas disponíveis para o voo de número 'VO197' em '2010-10-09'.

$T1 \leftarrow \Pi_{\text{NUMERO, NUMERO_TOTAL_POLTRONAS}} ((\sigma_{\text{NUMERO}='VO197'} (\text{VOO})))$

$\bowtie_{\text{VOO.ID_AVIAO=AVIAO.ID_AVIAO}} \text{AVIAO}$

$T2 (\text{NO_VOO, T_RES}) \leftarrow \Pi_{\text{NO_VOO, COUNT(*)}} (\sigma_{\text{NO_VOO}='VO197'} \wedge$
 $\text{DATA} = '2010-10-09' (\text{RESERVA_POLTRONA}))$

$\Pi_{\text{NO_VOO, (NUMERO_TOTAL_POLTRONAS - T_RES)}} (T1 \bowtie_{T1.NUMERO=T2.NO_VOO} T2)$

SELECT VOO.NO_VOO, (NUMERO_TOTAL_POLTRONAS – T_RES)
FROM ((**SELECT** NO_VOO, NUMERO_TOTAL_POLTRONAS,
 FROM VOO **JOIN** AVIAO **ON** VOO.ID_AVIAO = AVIAO.ID_AVIAO
 WHERE NO_VOO = 'VO197')
JOIN
 (**SELECT** RES.NO_VOO **AS** RES_VOO, COUNT(*) **AS** RES_TOTAL
 FROM RESERVA_POLTRONAS **AS** RES
 WHERE RES.NO_VOO = 'VO197' **AND** RES.DATA = '2010-10-09')
 ON NO_VOO = RES_VOO)