BD: Guião 6

Problema 6.1

a) Todos os tuplos da tabela autores (authors);

```sql
SELECT *
    FROM authors
```

b) O primeiro nome, o último nome e o telefone dos autores;

```sql
SELECT authors.au_fname, authors.au_lname, authors.phone
    FROM authors
```

c) Consulta definida em b) mas ordenada pelo primeiro nome (ascendente) e depois o último nome (ascendente);

```sql
SELECT authors.au_fname, authors.au_lname, authors.phone
    FROM authors
    ORDER BY authors.au_fname, authors.au_lname
```

d) Consulta definida em c) mas renomeando os atributos para (first_name, last_name, telephone);

```sql
SELECT authors.au_fname AS first_name, authors.au_lname AS last_name,
authors.phone AS telephone
    FROM authors
    ORDER BY authors.au_fname, authors.au_lname
```

e) Consulta definida em d) mas só os autores da Califórnia (CA) cujo último nome é diferente de 'Ringer';

```sql
SELECT authors.au_fname AS first_name, authors.au_lname AS last_name,
authors.phone AS telephone
    FROM authors
    WHERE authors.au_lname != 'Ringer'
    ORDER BY authors.au_fname, authors.au_lname
```

f) Todas as editoras (publishers) que tenham 'Bo' em qualquer parte do nome;

```
SELECT pub_name
    FROM publishers
    WHERE pub_name LIKE '%Bo%'
```

g) Nome das editoras que têm pelo menos uma publicação do tipo ‘Business’;

```
SELECT publishers.pub_name
    FROM publishers, titles
    WHERE publishers.pub_id=titles.pub_id AND titles.type='business'
    GROUP BY publishers.pub_name
```

h) Número total de vendas de cada editora;

```
SELECT publishers.pub_name, SUM(sales.qty) as total_sales
    FROM sales
        INNER JOIN titles ON sales.title_id = titles.title_id
        INNER JOIN publishers ON titles.pub_id = publishers.pub_id
    GROUP BY publishers.pub_name
    ORDER BY publishers.pub_name
```

i) Número total de vendas de cada editora agrupado por título;

```
SELECT publishers.pub_name, titles.title, SUM(sales.qty) as total_sales
    FROM sales
        INNER JOIN titles ON sales.title_id = titles.title_id
        INNER JOIN publishers ON titles.pub_id = publishers.pub_id
    GROUP BY publishers.pub_name, titles.title
    ORDER BY publishers.pub_name, titles.title
```

j) Nome dos títulos vendidos pela loja ‘Bookbeat’;

```
SELECT titles.title
    FROM titles, stores, sales
    WHERE stores.stor_name='Bookbeat' AND stores.stor_id=sales.stor_id AND
sales.title_id=titles.title_id
```

k) Nome de autores que tenham publicações de tipos diferentes;

```
SELECT authors.au_fname, authors.au_lname, COUNT(*) AS types_c
    FROM authors
```

```
        INNER JOIN titleauthor ON authors.au_id = titleauthor.au_id
        INNER JOIN titles ON titleauthor.title_id = titles.title_id
    GROUP BY authors.au_fname, authors.au_lname
    HAVING COUNT(DISTINCT titles.type) > 1
```

l) Para os títulos, obter o preço médio e o número total de vendas agrupado por tipo (type) e editora (pub_id);

```
SELECT [type], pub_id, AVG(price) AS average_price, SUM(ytd_sales) AS
all_time_sales
    FROM titles
    WHERE [type]!='UNDECIDED'
    GROUP BY [type], pub_id
```

m) Obter o(s) tipo(s) de título(s) para o(s) qual(is) o máximo de dinheiro "à cabeça" (advance) é uma vez e meia superior à média do grupo (tipo);

```
SELECT titles.type
    FROM titles
    GROUP BY titles.type
    HAVING MAX(titles.advance) > 1.5 * AVG(titles.advance)
```

n) Obter, para cada título, nome dos autores e valor arrecadado por estes com a sua venda;

```
SELECT titles.title, (authors.au_fname +' '+  authors.au_lname) AS [author], (
(titles.price*titles.ytd_sales * titles.royalty / 100) * titleauthor.royaltyper /
100) AS [money]
    FROM titles,titleauthor,authors
    WHERE titles.ytd_sales IS NOT NULL AND titles.title_id=titleauthor.title_id
AND titleauthor.au_id=authors.au_id
    ORDER BY titles.title
```

o) Obter uma lista que incluía o número de vendas de um título (ytd_sales), o seu nome, a faturação total, o valor da faturação relativa aos autores e o valor da faturação relativa à editora;

```
SELECT titles.title, titles.ytd_sales, titles.ytd_sales * titles.price AS profit,
                                        titles.ytd_sales * titles.price *
titles.royalty / 100 AS auth_revenue,
                                        titles.price * titles.ytd_sales -
titles.price * titles.ytd_sales * titles.royalty / 100 AS pub_revenue
    FROM titles
    ORDER BY titles.title
```

p) Obter uma lista que incluía o número de vendas de um título (ytd_sales), o seu nome, o nome de cada autor, o valor da faturação de cada autor e o valor da faturação relativa à editora;

```
SELECT titles.title, (authors.au_fname +' '+  authors.au_lname) AS [author], (
(titles.price*titles.ytd_sales * titles.royalty / 100) * titleauthor.royaltyper /
100) AS auth_revenue, (titles.price*titles.ytd_sales *( 100- titles.royalty) /
100) AS publisher_revenue
    FROM titles,titleauthor,authors, publishers
    WHERE titles.ytd_sales IS NOT NULL AND titles.title_id=titleauthor.title_id
AND titleauthor.au_id=authors.au_id AND titles.pub_id = publishers.pub_id
    ORDER BY titles.title,titles.ytd_sales, titles.price, titles.royalty,
au_fname, au_lname
```

q) Lista de lojas que venderam pelo menos um exemplar de todos os livros;

```
SELECT stores.stor_name
    FROM stores,sales,titles
    WHERE stores.stor_id = sales.stor_id AND sales.title_id = titles.title_id
    GROUP BY stores.stor_name
    HAVING (COUNT(stores.stor_name)) = (SELECT COUNT(*) FROM titles WHERE
titles.ytd_sales IS NOT NULL)
```

r) Lista de lojas que venderam mais livros do que a média de todas as lojas;

```
SELECT stor_name
    FROM sales
        INNER JOIN stores ON stores.stor_id=sales.stor_id
    GROUP BY stores.stor_name
    HAVING SUM(sales.qty)>(SELECT AVG(sales.qty) FROM sales);
```

s) Nome dos títulos que nunca foram vendidos na loja “Bookbeat”;

```
SELECT title FROM titles
    EXCEPT
    SELECT DISTINCT title
        FROM titlesINNER JOIN sales ON sales.title_id=titles.title_id
            INNER JOIN stores ON stores.stor_id=sales.stor_id
        WHERE stor_name='Bookbeat'
```

t) Para cada editora, a lista de todas as lojas que nunca venderam títulos dessa editora;

```
SELECT pub_name, stor_name
FROM publishers
    JOIN stores ON stor_id NOT IN (SELECT stor_id FROM sales INNER JOIN titles ON
sales.title_id = titles.title_id)
    ORDER BY pub_name
```

Problema 6.2

5.1

a) SQL DDL Script

a) SQL DDL File

b) Data Insertion Script

b) SQL Data Insertion File

c) Queries

a)

```
SELECT Pname, Ssn, Fname, Lname
    FROM project
        INNER JOIN works_on ON Pno=Pnumber
        INNER JOIN employee ON Essn=Ssn
```

b)

```
SELECT e.Fname, e.Lname
    FROM Company.Employee e
        JOIN Company.Employee s ON e.Super_ssn = s.Ssn
    WHERE s.Fname = 'Carlos' AND s.Minit = 'D' AND s.Lname = 'Gomes';
```

c)

```
SELECT Pname, SUM(Hours) AS THours
    FROM project
        INNER JOIN works_on ON Pnumber=Pno
    GROUP BY Pname
```

d)

```
SELECT e.Fname, e.Minit, e.Lname
    FROM employee e
        JOIN works_on w ON e.Ssn = w.Essn
        JOIN project p ON w.Pno = p.Pnumber
        JOIN department d ON p.Dnum = d.Dnumber
    WHERE d.Dnumber = 3 AND w.Hours > 20 AND p.Pname = 'Aveiro Digital'
```

e)

```
SELECT Fname, Minit, Lname
    FROM  employee
        LEFT outer JOIN works_on ON Ssn=Essn
    WHERE Pno IS NULL
```

f)

```
SELECT department.Dname, AVG(employee.Salary) AS AvgSalary
    FROM department
        JOIN employee ON department.Dnumber = employee.Dno
    WHERE employee.Sex = 'F'
    GROUP BY department.Dname;
```

g)

```
SELECT Fname, Minit, Lname FROM employee
INNER JOIN (
            SELECT Essn, COUNT(Essn) AS quantos FROM dependent
            GROUP BY Essn
            HAVING quantos > 2
            ) AS dependentes
ON Ssn=Essn
```

h)

```
SELECT Fname, Minit, Lname
    FROM department
        INNER JOIN employee ON Ssn=Mgr_ssn
        LEFT outer JOIN dependent ON Essn=Ssn
    WHERE Dependent_name IS NULL
```

i)

```sql
SELECT Fname, Minit, Lname, Address FROM employee
INNER JOIN (
            SELECT *
                FROM project
                    INNER JOIN dept_location ON Dnum=Dnumber
                WHERE Dlocation!='Aveiro' AND Plocation='Aveiro'
            ) AS PROJECT_LST
ON Dno=Dnum
```

5.2

a) SQL DDL Script

a) SQL DDL File

b) Data Insertion Script

b) SQL Data Insertion File

c) Queries

a)

```sql
SELECT GestStock_fornecedor.nome, GestStock_fornecedor.nif
    FROM GestStock_fornecedor
    LEFT JOIN GestStock_encomenda ON GestStock_fornecedor.nif =
GestStock_encomenda.fornecedor
    WHERE GestStock_encomenda.numero IS NULL;
```

b)

```sql
SELECT codProd, AVG(unidades) AS nMed
    FROM GestStock_encomenda
    JOIN GestStock_item ON GestStock_encomenda.numero = GestStock_item.numEnc
    GROUP BY codProd;
```

c)

```sql
SELECT AVG(produtos) AS mediaProd
FROM (
    SELECT numEnc, COUNT(codProd) AS produtos
    FROM GestStock_encomenda
    JOIN GestStock_item ON GestStock_encomenda.numero = GestStock_item.numEnc
    GROUP BY numEnc
) AS subquery;
```

d)

```
SELECT GestStock_fornecedor.nome, GestStock_produto.nome,
SUM(GestStock_item.unidades) AS unidades
    FROM GestStock_fornecedor
    INNER JOIN GestStock_encomenda ON GestStock_fornecedor.nif =
GestStock_encomenda.fornecedor
    INNER JOIN GestStock_item ON GestStock_encomenda.numero =
GestStock_item.numEnc
    INNER JOIN GestStock_produto ON GestStock_item.codProd =
GestStock_produto.codigo
    GROUP BY GestStock_fornecedor.nome, GestStock_produto.nome;
```

5.3

a) SQL DDL Script

a) SQL DDL File

b) Data Insertion Script

b) SQL Data Insertion File

c) Queries

a)

```
SELECT PRESCRICAO_Paciente.nome
FROM PRESCRICAO_Paciente
LEFT JOIN PRESCRICAO_Prescricao ON PRESCRICAO_Paciente.numUtente =
PRESCRICAO_Prescricao.numUtente
WHERE PRESCRICAO_Prescricao.numPresc IS NULL;
```

b)

```
SELECT PRESCRICAO_Medico.especialidade, COUNT(PRESCRICAO_Medico.especialidade) AS
Npresc
FROM PRESCRICAO_Medico
```

```
JOIN PRESCRICAO_Prescricao ON PRESCRICAO_Medico.numSNS =
PRESCRICAO_Prescricao.numMedico
GROUP BY PRESCRICAO_Medico.especialidade;
```

c)

```
SELECT PRESCRICAO_Farmacia.nome, COUNT(PRESCRICAO_Prescricao.numPresc) AS N_presc
FROM PRESCRICAO_Prescricao
JOIN PRESCRICAO_Farmacia ON PRESCRICAO_Prescricao.farmacia =
PRESCRICAO_Farmacia.nome
GROUP BY PRESCRICAO_Farmacia.nome;
```

d)

```
SELECT PRESCRICAO_Farmaco.nome
FROM PRESCRICAO_Farmaco
LEFT JOIN PRESCRICAO_Presc_farmaco ON PRESCRICAO_Farmaco.numRegFarm =
PRESCRICAO_Presc_farmaco.numRegFarm AND PRESCRICAO_Farmaco.nome =
PRESCRICAO_Presc_farmaco.nomeFarmaco
WHERE PRESCRICAO_Presc_farmaco.numPresc IS NULL
AND PRESCRICAO_Farmaco.numRegFarm = 906;
```

e)

```
SELECT PRESCRICAO_Prescricao.farmacia, numReg, COUNT(PRESCRICAO_Farmaco.nome) AS
Number
    FROM PRESCRICAO_Farmaceutica
    JOIN PRESCRICAO_Farmaco ON numReg = numRegFarm
    JOIN PRESCRICAO_Presc_farmaco ON PRESCRICAO_Farmaco.numRegFarm =
PRESCRICAO_Presc_farmaco.numRegFarm
    JOIN PRESCRICAO_Prescricao ON PRESCRICAO_Presc_farmaco.numPresc =
PRESCRICAO_Prescricao.numPresc
    WHERE PRESCRICAO_Prescricao.farmacia IS NOT NULL
    GROUP BY PRESCRICAO_Prescricao.farmacia, numReg
```

f)

```
SELECT nome FROM
    (
        SELECT nome, COUNT(PRESCRICAO_Prescricao.numMedico) AS medicoCount
            FROM PRESCRICAO_Paciente
            JOIN PRESCRICAO_Prescricao ON PRESCRICAO_Paciente.numUtente = PRESCRICAO_Prescricao.numUtente
            GROUP BY nome
    ) AS P
    WHERE medicoCount > 1
```