

Instituto Politécnico do Porto
Instituto Superior de Engenharia do Porto

Trabalho Prático – Parte I

Grupo 7

Licenciatura em Engenharia Informática

Base de Dados

José Cunha nº 1181494

Guilherme Oliveira nº 1170814

João Zarco nº 1170531

Índice

Introdução	3
Explicação das query's	4

Introdução

Este documento tem em vista a explicação das query's que utilizamos para a resolução dos exercícios propostos pelo o professor no âmbito do trabalho prático.

Todas as query's neste documento vão ser analisadas individualmente.

Cada explicação vai contar o enunciado, a query e a própria explicação.

Explicação das query's

- A) Liste a média de salários e o salário anual (considerando o subsídio de férias e Natal) para os empregados do armazém 'Parafusos'.

Query:

```
SELECT (SELECT AVG(E.salario_semanal)
        FROM empregado) AS media_salario_semanal
,(E.salario_semanal * 60) AS Salario_Anual
FROM empregado E
WHERE E.cod_armazem = (
        SELECT A.cod_armazem
        FROM armazem A
        WHERE A.nome like 'Parafusos');
```

Explicação: Primeiramente começamos por criar uma tabela onde retornasse o código do armazém onde o nome fosse "Parafusos". Depois, juntamos essa tabela a outra que retornava a media do salário, mais o salário anual. Por fim, fizemos as alterações necessárias para obter o resultado esperado.

- B) Liste as ordens de compras satisfeitas (estado = 3) que não possuem nenhum dos produtos do fornecedor que oferece o maior desconto.

Query:

```
SELECT *
FROM ordem_compra
WHERE estado = 3
AND cod_fornecedor NOT IN (
    SELECT cod_fornecedor
    FROM fornecedor_produto
    WHERE desconto = (
        SELECT MAX(desconto)
        FROM fornecedor_produto
    )
);
```

Explicação: Começamos por criar uma tabela onde retornasse o máximo desconto. Juntamos essa tabela a outra que fosse buscar esse desconto, e retornasse o código do fornecedor. Por fim, juntamos isso a uma outra tabela que retornasse a ordem de compra que estado fosse igual a três, e não fosse igual a esse código de fornecedor.

- C) Liste o nome dos armazéns que têm em stock todos os produtos que foram fornecidos ao armazém que possui o maior número de empregados.

Query:

```
SELECT DISTINCT A.nome
FROM   armazem_produto APA
INNER JOIN armazem A ON A.cod_armazem = APA.cod_armazem
WHERE NOT EXISTS(
    SELECT cod_produto
    FROM   armazem_produto
    WHERE cod_armazem IN (SELECT cod_armazem AS cod_armazem_empregados
                          FROM   empregado
                          GROUP BY cod_armazem
                          HAVING COUNT(cod_empregado) = (SELECT MAX(COUNT(cod_empregado))
                                                         FROM   empregado
                                                         GROUP BY cod_armazem))
)
MINUS
SELECT cod_produto
FROM   armazem_produto APB
WHERE APA.cod_armazem = APB.cod_armazem
);
```

Explicação: Esta query primeiramente seleciona o armazém com maior número de empregados, depois vai buscar o código dos produtos desse armazém. Utilizando o “MINUS” subtraímos os produtos do armazém com o maior número de empregados a todos os armazéns. Se o conjunto não for existente, então significa que este mesmo armazém tem os produtos todos, e vai retornar o nome desse armazém

- D) Indique quais os armazéns que, no período de 01/03/2018 a 15/10/2018, têm o número total de ordens de compra pendentes (estado = 2), maior do que qualquer armazém da cidade do Porto nesse período.

Query:

```
SELECT *
FROM   armazem WHERE cod_armazem IN (
    SELECT cod_armazem FROM(
        SELECT A.cod_armazem ,COUNT(OC.nr_ordem) AS num_vendas
        FROM   ordem_compra OC, armazem A, empregado EMP
        WHERE  OC.cod_empregado = EMP.cod_empregado
              AND EMP.cod_armazem = A.cod_armazem
              AND OC.data_compra BETWEEN TO_DATE('2018-03-01', 'YYYY-MM-DD')
              AND TO_DATE('2018-10-15', 'YYYY-MM-DD')
              AND OC.estado = 2
        GROUP BY A.cod_armazem
        HAVING COUNT(*) > (
            SELECT MAX(num_vendas)
            FROM(
                SELECT COUNT(*) AS num_vendas, A.cod_armazem
                FROM   ordem_compra OC, armazem A, empregado EMP
                WHERE  OC.cod_empregado = EMP.cod_empregado
                      AND EMP.cod_armazem = A.cod_armazem
                      AND A.cidade = 'Porto'
                      AND OC.data_compra BETWEEN TO_DATE('2018-03-01', 'YYYY-MM-DD')
                      AND TO_DATE('2018-10-15', 'YYYY-MM-DD')
                      AND OC.estado = 2
                GROUP BY A.cod_armazem
            )
        )
    )
);
```

Explicação: Primeiramente criamos uma tabela onde contasse o número de vendas onde o armazém fosse no Porto, o estado fosse 2, e fosse entre as datas pretendidas, agrupando pelo código do armazém e selecionávamos o máximo. De seguida fazíamos o mesmo para o resto dos outros armazéns, e quando um armazém tivesse maior numero de vendas superior à do Porto, retornavas esse mesmo armazém.

- E) Liste o nome do empregado que não é supervisor e que efetuou ordens de compra em maior número do que todos os supervisores que possuem um salário mensal entre 1000€ e 3000€

Query:

```
SELECT nome
FROM empregado
WHERE cod_empregado IN (
    SELECT cod_empregado
    FROM (
        SELECT COUNT(O.nr_ordem) as numero_de_ordens, E.cod_empregado
        FROM empregado E, ordem_compra O
        WHERE E.cod_supervisor IS NULL
        AND E.cod_empregado = O.cod_empregado
        GROUP BY E.cod_empregado
    )
    WHERE numero_de_ordens > (
        SELECT COUNT(*)
        FROM ordem_compra
        WHERE cod_empregado IN (
            SELECT cod_empregado
            FROM empregado
            WHERE cod_supervisor IS NOT NULL
            AND salario_semanal*4 BETWEEN 1000 AND 3000
        )
    )
);
```

Explicação: Primeiramente fizemos uma tabela de modo a ter todos os empregados que não são supervisores, e de seguida os supervisores que recebem entre 1000€ a 3000€ mensais. Depois criamos outra tabela em que contamos o número de ordem dos empregados não supervisores e os supervisores selecionados, e comparamos as suas ordens. Finalmente juntamos a informação de modo a retornar a solução pretendida.

- F) Liste o código do armazém, o número do corredor e o número da prateleira dos produtos que foram menos vezes comprados.

Query:

```
SELECT cod_armazem, corredor, prateleira
FROM armazen_produto
WHERE cod_produto IN (SELECT cod_produto
                      FROM ordem_compra_produto OP
                      GROUP BY cod_produto
                      HAVING COUNT(*) = (SELECT MIN(COUNT(*))
                                         FROM ordem_compra_produto
                                         GROUP BY cod_produto));
```

Explicação: Primeiramente formamos uma tabela com os produtos menos comprados. Fizemos uma query não correlacionada, que para cada produto diz a prateleira, o número do corredor, e o código do armazém.

- G) Liste todos os corredores de armazéns onde atualmente se encontram produtos que tiveram o maior número de ordens de compra com desconto acima dos 20%.

Query:

```
SELECT corredor
FROM armazen_produto AP
WHERE cod_produto IN (SELECT cod_produto
                     FROM ordem_compra_produto
                     WHERE desconto_pedido > 20
                     GROUP BY cod_produto
                     HAVING COUNT(nr_ordem) = (SELECT MAX(COUNT(nr_ordem))
                                                FROM ordem_compra_produto
                                                WHERE desconto_pedido > 20
                                                GROUP BY cod_produto));
```

Explicação: Primeiro selecionamos o máximo da contagem do número de ordens com desconto maior que 20% e agrupamos por código do produto. Depois juntamos uma tabela que selecionava o produto com o máximo numero de ordem de compra. Por fim juntamos uma query que retornasse o corredor com esse mesmo produto.

- H) Liste o produto e volume de compras mensal, para o ano 2018, dos produtos que estão em armazéns cujo stock está pelo menos 50% acima do stock mínimo.

Query:

```
SELECT COUNT(PP.cod_produto) as quantidade, PP.mes, PP.cod_produto, P.descricao, P.unidade_medida, P.preco
FROM (
    SELECT OCP.cod_produto, EXTRACT(month FROM OC.data_compra) as mes
    FROM ordem_compra OC, ordem_compra_produto OCP
    WHERE EXTRACT(year FROM OC.data_compra) = 2018
    AND OC.nr_ordem = OCP.nr_ordem
    AND OCP.cod_produto IN(
        SELECT cod_produto
        FROM produto
        WHERE cod_produto IN (
            SELECT cod_produto
            FROM armazem_produto
            WHERE stock*1.5 > stock_minimo
        )
    )
) PP, produto P
WHERE PP.cod_produto = P.cod_produto
GROUP BY PP.mes, PP.cod_produto, P.descricao, P.unidade_medida, P.preco
ORDER BY PP.mes;
```

Explicação: Começamos por fazer uma tabela que retornasse o código do produto que tem stock pelo menos acima dos 50% mínimos. De seguida, realizamos outra tabela que retornava esse produto em que a data de compra era em 2018. Por fim, realizamos a tabela final que listava o produto e o seu volume de compras, agrupando por mês, e ordenamos por mês.

- I) Liste as ordens de compra no estado satisfeito, entre o mês de Junho e Agosto de 2018, cuja hora de elaboração é inferior às 10 horas da manhã e com uma diferença entre a data de compra e a data da entrega superior a 10 dias.

Query:

```
SELECT *
FROM ordem_compra
WHERE TO_CHAR(data_compra, 'MM') BETWEEN 6 AND 8
    AND TO_CHAR(data_compra, 'YYYY') = 2018
    AND estado = 3
    AND (cast(data_entrega AS DATE) - cast(data_compra AS DATE)) > 10
    AND TO_CHAR(data_compra, 'HH') < 10;
```

Explicação: Inicialmente começamos por realizar uma tabela que retornava as ordens de compra entre os meses de junho e agosto, depois outra tabela que retornava a diferença entre a data de compra e data de entrega fosse superior a 10 dias, e a hora de elaboração fosse inferior às 10 da manhã. Finalmente juntamos a informação de modo a retornar a solução pretendida.

- J) Liste para o empregado com o código = 1223 a percentagem de ordens de compra satisfeitas em que foi responsável face às ordens de compra satisfeitas que pertencem ao armazém 'Tintas' e possuem um valor total compreendido entre 5.000€ e 10.000€.

Query:

```
SELECT
  ((SELECT COUNT(nr_ordem)
    FROM ordem_compra
    WHERE cod_empregado=1223 AND estado=3) /
  (SELECT COUNT(nr_ordem)
    FROM ordem_compra OC
    INNER JOIN Empregado E ON E.cod_empregado = OC.cod_empregado
    INNER JOIN Armazem A ON A.cod_armazem = A.cod_Armazem
    WHERE UPPER(A.nome) LIKE 'TINTAS'
    AND OC.valor_total BETWEEN 5000 AND 10000)) *100
FROM Empregado
FETCH FIRST 1 ROW ONLY;
```

Explicação: Primeiramente começamos por criar uma tabela onde contasse o número de ordens de compra do armazém "Tintas" e faz a percentagem. Depois fizemos outra tabela para o empregado 1223 e o estado igual a 3. Por fim, juntamos de modo a obter o resultado pretendido.