

Sistemas de Informação e Bases de Dados

Enunciado da Etapa 3 do Projeto de 2022/2023

Ponto de partida

Considere para esta etapa do projeto o seguinte esquema relacional sobre a gestão de uma loja, especificado em SQL-DDL, e *inspirado* nos conceitos das etapas 1 e 2.

```
-----  
DROP TABLE linhafatura;  
DROP TABLE fatura;  
DROP TABLE produto;  
DROP TABLE cliente;  
  
-----  
  
CREATE TABLE cliente (  
    nif          NUMBER (9),  
    nome         VARCHAR (20) CONSTRAINT nn_cliente_nome      NOT NULL,  
    genero       CHAR (1)  CONSTRAINT nn_cliente_genero       NOT NULL,  
    nascimento   NUMBER (4)  CONSTRAINT nn_cliente_nascimento NOT NULL, -- Ano.  
    localidade   VARCHAR (20) CONSTRAINT nn_cliente_localidade NOT NULL,  
--  
    CONSTRAINT pk_cliente  
        PRIMARY KEY (nif),  
--  
    CONSTRAINT ck_cliente_nif -- RIA 16.  
        CHECK (nif BETWEEN 100000000 AND 999999999),  
--  
    CONSTRAINT ck_cliente_genero -- RIA 17.  
        CHECK (genero IN ('F', 'M')),  
--  
    CONSTRAINT ck_cliente_nascimento  
        CHECK (nascimento >= 1900) -- Valor razoável.  
);  
  
-----
```

```
CREATE TABLE produto (  
    ean13        NUMBER (13),  
    nome         VARCHAR (20)  CONSTRAINT nn_produto_nome      NOT NULL,  
    categoria    CHAR (7)     CONSTRAINT nn_produto_categoria  NOT NULL,  
    preco        NUMBER (6,2)  CONSTRAINT nn_produto_preco      NOT NULL,  
    stock        NUMBER (4)    CONSTRAINT nn_produto_stock      NOT NULL,  
--  
    CONSTRAINT pk_produto  
        PRIMARY KEY (ean13),  
--  
    CONSTRAINT ck_produto_ean13 -- RIA 11.  
        CHECK (ean13 BETWEEN 1000000000000 AND 9999999999999),  
--
```

```

CONSTRAINT ck_produto_categoria -- RIA 13.
    CHECK (categoria IN ('Comida', 'Roupa', 'Beleza', 'Animais')),
--
CONSTRAINT ck_produto_preco -- RIA 14.
    CHECK (preco > 0.0),
--
CONSTRAINT ck_produto_stock
    CHECK (stock >= 0)
);

```

```

CREATE TABLE fatura (
    numero    NUMBER (6),
    data      DATE          CONSTRAINT nn_fatura_data    NOT NULL,
    cliente    CONSTRAINT nn_fatura_cliente NOT NULL,
--
    CONSTRAINT pk_fatura
        PRIMARY KEY (numero),
--
    CONSTRAINT fk_fatura_cliente
        FOREIGN KEY (cliente)
        REFERENCES cliente (nif),
--
    CONSTRAINT ck_fatura_numero -- RIA 18.
        CHECK (numero >= 1)
);

```

```

CREATE TABLE linhafatura (
    fatura,
    produto,
    unidades NUMBER (4) CONSTRAINT nn_linhafatura_unidades NOT NULL,
--
    CONSTRAINT pk_linhafatura
        PRIMARY KEY (fatura, produto),
--
    CONSTRAINT fk_linhafatura_fatura
        FOREIGN KEY (fatura)
        REFERENCES fatura (numero),
--
    CONSTRAINT fk_linhafatura_produto
        FOREIGN KEY (produto)
        REFERENCES produto (ean13),
--
    CONSTRAINT ck_linhafatura_unidades -- RIA 19.
        CHECK (unidades > 0)
);

```

Objetivo

Traduzir para *interrogações SQL* os seguintes pedidos de dados:

1. NIF, nome, e idade das clientes femininas com apelido Dias, que compraram um ou mais produtos de Beleza durante o ano de 2021. O EAN-13 e nome do(s) produto(s) também devem ser mostrados, bem como o número e data da(s) respetiva(s) fatura(s) de compra. O resultado deve vir ordenado de forma ascendente pela idade e nome das clientes, e de forma descendente pela data das faturas e nome dos produtos. *Nota:* a extração do ano a partir de uma data pode ser feita usando `TO_CHAR(data, 'YYYY')`.

Variante com menor cotação: sem o cálculo da idade das clientes.

2. NIF e nome dos clientes masculinos que nunca compraram produtos de Beleza (independentemente do ano), e que, considerando apenas 2021, ou não compraram Roupa nesse ano ou compraram Roupa em até duas ocasiões. Assuma que cada fatura com algum produto de Roupa representa uma ocasião de compra, não interessando se foi comprada pouca ou muita roupa. O resultado deve vir ordenado pelo nome dos clientes de forma ascendente e pelo NIF dos clientes de forma descendente. *Nota:* a extração do ano a partir de uma data pode ser feita usando `TO_CHAR(data, 'YYYY')`.

Variantes com menor cotação: a) sem a verificação dos clientes nunca terem comprado produtos de Beleza; e b) sem a verificação do número de ocasiões em que compraram Roupa em 2021.

3. Produtos de Comida com preço abaixo da média dos preços de todos os produtos (independentemente da categoria), e que tenham sido alguma vez comprados por *todos* os clientes do Porto na parte da manhã dos dias, isto é, entre as 8h e as 12h. O resultado deve vir ordenado pelo preço dos produtos de forma descendente e pelo EAN-13 dos produtos de forma ascendente. *Nota:* a extração da hora do dia a partir de uma data pode ser feita usando `TO_CHAR(data, 'HH24')`.

Variantes com menor cotação: a) sem a verificação do preço dos produtos de Comida ser inferior à média dos preços de todos os produtos; e b) sem as verificações da localidade dos clientes e da hora das compras.

4. NIF e nome dos clientes que gastaram mais dinheiro em compras em cada ano, separadamente para clientes femininos e masculinos, devendo o género dos clientes e o total gasto em cada ano também aparecer no resultado. A ordenação do resultado deve ser pelo ano de forma descendente e pelo género dos clientes de forma ascendente. No caso de haver mais do que um(a) cliente com o mesmo máximo de dinheiro gasto num ano, devem ser mostrados todos esses clientes.

Variantes com menor cotação: a) mostrar o total de dinheiro gasto em compras por cada cliente em cada ano, sem verificar se foram os que mais gastaram; e b) sem a distinção entre clientes femininos e masculinos.

A ter em conta

- É fundamental que as interrogações possam ser *executadas sem erros*.
- Em caso de dificuldade em cumprir a totalidade de um pedido de dados, podem ser adotadas uma ou mais das *variantes com menor cotação*.
- O resultado de qualquer interrogação *não deve ter linhas repetidas*. Contudo, só deve ser usada a cláusula de remoção de duplicados se esta for mesmo necessária.
- As interrogações devem ser *inteligíveis* e estar bem alinhadas, de acordo com a apresentação dos exemplos das aulas teóricas, e não devem ter mudanças bruscas de linha.
- As interrogações devem ser as mais *simples* possíveis, evitando, por exemplo, a inclusão de tabelas desnecessárias na cláusula FROM ou o uso excessivo de subinterrogações.
- A título meramente sugestivo, podem ser usadas as seguintes *iniciais* para referir de forma abreviada cada uma das tabelas fornecidas: cliente C, produto P, fatura F, e linhafatura L.

Estrutura do relatório

- O relatório deve ser um *script* SQL contendo as interrogações SQL devidamente anotadas com o número e texto dos pedidos de dados, e eventuais variantes com menor cotação.
- No início do *script* SQL deve estar a sigla da unidade curricular e o ano letivo, a etapa do projeto, o número do grupo, e o nome, número, e turma dos alunos.

Entrega do relatório

- O relatório deve ser entregue num ficheiro de texto simples com nome SIBD-2223-GXX-E3.SQL, onde XX representa o número do grupo, com dois dígitos (ex. 01).
- A entrega do relatório é feita através de uma atividade na página de entrada do Moodle da unidade curricular.
- O prazo de entrega do relatório é **2 de dezembro de 2022** (sexta-feira).

Bom trabalho nesta etapa do projeto!