

Projeto de Bases de Dados

2021/2022

3ªEntrega

Docente: [Pedro Miguel Leão Veloso Dias](#)
Grupo 61

Aluno	Esforço
Francisco Sanchez (99071)	9 horas (33%)
Inês Alves (99xxx)	9 horas (33%)
João Costa (99088)	9 horas (33%)

SQL

```
-- Qual o nome do retalhista (ou retalhistas) responsáveis pela reposição
do maior número de categorias?
SELECT retalhista.nome FROM responsavel_por
NATURAL JOIN retalhista
GROUP BY retalhista.nome
HAVING COUNT(retalhista.nome) >= ALL (
    SELECT COUNT(retalhista.nome) FROM responsavel_por
    NATURAL JOIN retalhista
    GROUP BY retalhista.nome
)

--Qual o nome do ou dos retalhistas que são responsáveis por todas as
categorias simples?
SELECT retalhista.nome FROM responsavel_por
NATURAL JOIN retalhista
JOIN categoria_simples ON responsavel_por.nome_cat = categoria_simples.nome

--Quais os produtos (ean) que nunca foram repostos?
SELECT produto.ean FROM produto
GROUP BY produto.ean
HAVING ean NOT IN (
    SELECT produto.ean FROM evento_reposicao
    NATURAL JOIN produto
)

--Quais os produtos (ean) que foram repostos sempre pelo mesmo retalhista?
SELECT ean FROM produto
GROUP BY ean
HAVING ean NOT IN (
    SELECT ean FROM evento_reposicao
    GROUP BY ean
)
```

Restrições de Integridade

```
--(RI-1) Uma Categoria não pode estar contida em si própria

CREATE OR REPLACE FUNCTION check_category_is_valid()
RETURNS TRIGGER AS
$$
BEGIN
    IF NEW.super_categoria == NEW.categoria THEN
        RAISE EXCEPTION 'Uma Categoria não pode estar contida em si própria'
    END IF;
    RETURN NEW;
END;

$$ LANGUAGE plpgsql;

--(RI-4) O número de unidades repostas num Evento de Reposição
-- não pode exceder o número de unidades especificado no Planograma

CREATE OR REPLACE FUNCTION check_product_replacement()
RETURNS TRIGGER AS
$$
BEGIN
    IF NEW.evento_reposicao.unidades > NEW.planograma.unidades THEN
        RAISE EXCEPTION 'O número de unidades repostas não pode exceder o
valor especificado no planograma'
    END IF;
    RETURN NEW;
END;

$$ LANGUAGE plpgsql;

-- (RI-5) Um Produto só pode ser repostado numa Prateleira que apresente (pelo
menos) uma das
-- Categorias desse produto

CREATE OR REPLACE FUNCTION check_shelf_category()
RETURNS TRIGGER AS
$$
BEGIN
    IF NEW.prateleira.categoria != NEW.produto.categoria THEN
        RAISE EXCEPTION 'Um Produto só pode ser repostado numa Prateleira que
apresente (pelo menos) uma das categorias desse produto '
    END IF;
    RETURN NEW;
END;

$$ LANGUAGE plpgsql;
```

Aplicação

Todos os ficheiros da aplicação estão contidos na pasta web sendo que os ficheiros html estão inseridos na subdiretoria “templates”.

A página principal, criada com o root.html, contém o menu principal da aplicação, o qual dá acesso ao utilizador a todas as ferramentas exigidas da mesma (gerir categorias, gerir retalhistas, listar reposições e listar subcategorias).

Ao escolher uma das ferramentas, caso a operação tenha sucesso, o utilizador é reencaminhado para o website da ferramenta escolhida. Caso aconteça algum erro, o utilizador nunca tem acesso a qualquer informação sobre o mesmo, e é redirecionado para uma página de erro.

Nas duas primeiras operações (gerir categorias e gerir retalhistas), o utilizador depara-se com uma tabela contendo as categorias e utilizadores já existentes, respetivamente, com a opção de poder remover os mesmos.

Abaixo da tabela, em ambos os casos, existe um submenu onde o utilizador poderá adicionar uma nova categoria ou retalhista, dependendo da ferramenta que escolher. No caso da ferramenta gerir categorias, na tabela principal, é-lhe dada também a possibilidade de adicionar uma subcategoria a uma das super-categorias já criadas anteriormente, esta subcategoria terá de ser uma categoria já previamente criada, caso se tente adicionar uma subcategoria com uma categoria inexistente, o utilizador é redirecionado para uma página de erro.

Nas outras duas ferramentas (listar reposições e listar subcategorias) é apresentado ao utilizador uma tabela, em que nesta estão listados os eventos de reposição e as super-categorias, respetivamente, pertencentes à base de dados. Na operação listar reposições o utilizador tem a opção de obter mais informação sobre um certo evento de reposição (categorias e unidades repostas), se seleccionar o número de série de tal evento, sendo redirecionado para uma nova página onde lhe será disponibilizada uma tabela com tais informações. Na operação listar subcategorias o utilizador pode consultar todas as subcategorias de qualquer super-categoria já existente na base de dados. Ao seleccionar uma super categoria o utilizador é redirecionado para uma nova página que contém uma tabela preenchida por todas as subcategorias da super-categoria seleccionada previamente.

link: <http://web2.ist.utl.pt/ist199088/app.cgi/>

Análise de Dados

1.

```
SELECT dia_semana, concelho, SUM(unidades)
FROM vendas
WHERE ano BETWEEN EXTRACT(YEAR FROM data_inicial) AND EXTRACT(YEAR
FROM data_final)
WHERE mes BETWEEN EXTRACT(MONTH FROM data_inicial) AND
EXTRACT(MONTH FROM data_final)
WHERE dia_mes BETWEEN EXTRACT(DOM FROM data_inicial) AND
EXTRACT(DOM FROM data_final)
GROUP BY CUBE (dia_semana, concelho);
```

2.

```
SELECT concelho, categoria, dia_semana, SUM(unidades)
FROM vendas
WHERE distrito = "Lisboa"
GROUP BY CUBE (concelho, categoria, dia_semana);
```

Índices

7.1-

Na tabela retalhista não há necessidade de criar um índice dado que o tin já é chave primária e, desse modo, já tem associado um índice do tipo Btree. No entanto, por se tratar de uma igualdade, o índice mais eficiente seria o hash.

```
create index tin_index on retalhista USING hash(tin);
```

Também não há necessidade de criar um índice para o atributo nome pois o facto de ser único, pelas restrições de integridade, já lhe atribuí um índice Btree.

Para a tabela responsavel_por, sugerimos um índice hash para o atributo nome_cat, visto que é chave estrangeira e não cria índice e que é usado em igualdades.

```
create index nome_index on responsavel_por USING hash(nome_cat);
```

7.2-

Na tabela tem_categoria, sugerimos um índice no atributo nome para facilitar o GROUP BY e condição associada ao mesmo atributo.

```
create index nome_index2 on tem_categoria(nome) USING btree;
```

Para a tabela produto, sugerimos a criação de um índice hash no atributo cat porque é mais seletivo que a desc e facilitará a igualdade

```
create index cat_incx on produto USING hash(cat);
```