

SQL QUERIES

DataBase "LOJA":



-- Answer 1

```
SELECT * FROM loja.produto
LEFT JOIN loja.promocoies ON idproduto = idprodutoFK
```

-- Answer 2

```
SELECT idproduto, produto, preco, stock, ifnull(novopreco, ' - ') as 'promoção'
FROM loja.produto
LEFT JOIN loja.promocoies ON idproduto = idprodutoFK
```

-- Answer 3

```
SELECT produto,
       concat(ifnull(novopreco, preco), '€') as 'preço já com desconto aplicado!',
       stock
FROM loja.produto
LEFT JOIN loja.promocoies ON idproduto = idprodutoFK
```

-- or

```
SELECT produto,
       CASE
           WHEN ativo = 1 THEN concat(novopreco, '€')
           ELSE concat(preco, '€')
       END as 'preço já com desconto aplicado!',
       stock
FROM loja.produto
LEFT JOIN loja.promocoies ON idproduto = idprodutoFK
```

-- Answer 4

```
CREATE VIEW produtos_com_promocoes
as
SELECT produto, concat(ifnull(novopreco, preco), '€') as 'preço já com desconto aplicado!',
stock FROM loja.produto
LEFT JOIN loja.promocoes ON idproduto = idprodutoFK
```

-- Answer 5

```
SELECT * FROM produtos_com_promocoes
```

-- Answer 6

```
CREATE PROCEDURE selecionar_produtos_com_preco_atualizado()
BEGIN
    SELECT produto,
           CASE
               WHEN ativo = 1 THEN concat(novopreco, '€')
               ELSE concat(preco, '€')
           END as 'preço já com desconto aplicado!',
           stock
    FROM loja.produto
    LEFT JOIN loja.promocoes ON idproduto = idprodutoFK;
END $$
delimiter;
```

-- Answer 7

```
CALL selecionar_produtos_com_preco_atualizado()
```

-- Answer 8

```
delimiter $$
CREATE PROCEDURE selecionar_produto_por_ID(id int unsigned)
BEGIN
    SELECT * FROM produto
    WHERE idproduto = id;
END $$
delimiter ;

CALL selecionar_produto_por_ID(3)
```

-- Answer 9

```
delimiter $$
CREATE PROCEDURE retornar_novo_preco_do_produto_porID(id int unsigned)
BEGIN
    SELECT
        CASE
            WHEN ativo = 1 THEN concat(novopreco, '€')
            ELSE concat(preco, '€')
        END as 'Preço'
    FROM loja.produto
    LEFT JOIN loja.promocoes ON idproduto = idprodutoFK
    WHERE idproduto = id;
END $$
delimiter ;
```

CALL retornar_novo_preco_do_produto_porID(3)

-- Answer 10

```
delimiter $$
CREATE PROCEDURE inserir_encomenda_e_obter_idencomenda(
    estado_enc VARCHAR(45),
    preco_enc DECIMAL(7, 2),
    id int unsigned
)
BEGIN
    INSERT INTO encomenda (estado, preco_total, idutilizadorFK)
    VALUES (estado_enc, preco_enc, id);
    SELECT last_insert_id() as idencomenda;
END $$
delimiter ;
```

CALL inserir_encomenda_e_obter_idencomenda('POR PAGAR',null,1)

-- Answer 11

```
INSERT INTO utilizador (nome, morada, email, password)
VALUES ('Carolina', 'Porto', 'carol@gmail.com', '321');
```

-- Answer 12

```
SELECT sum(quantidade*preco) as total
FROM itemcompra
WHERE idencomendaFK=2
```

-- Answer 13

```
delimiter $$
CREATE PROCEDURE calcular_custototal_encomenda(idencomenda int unsigned)
BEGIN
    SELECT sum(quantidade * preco) as total
    FROM itemcompra
    where idencomendaFK = idencomenda;
END$$
delimiter ;

CALL calcular_custototal_encomenda(2)
```

-- Answer 14

```
delimiter $$
CREATE PROCEDURE atualizar_custototal_encomenda(id_enc int unsigned)
BEGIN
    DECLARE total DECIMAL(7, 2)
    SELECT sum(quantidade * preco) into total FROM itemcompra
    WHERE idencomendaFK = id_enc;
    UPDATE encomenda
    SET preco_total = total
    WHERE idencomenda = id_enc;
END$$
delimiter ;

CALL atualizar_custototal_encomenda(1)
```

-- Answer 15

```
delimiter $$
CREATE TRIGGER atualizaStock1 AFTER INSERT ON itemcompra
FOR EACH ROW
BEGIN
    UPDATE produto SET stock = stock - NEW.quantidade
    WHERE idproduto = NEW.idprodutoFK;
END $$
delimiter ;
```

-- Answer 16

```
delimiter $$
CREATE TRIGGER atualizaStockOnDelete AFTER DELETE ON itemcompra
FOR EACH ROW
BEGIN
    UPDATE produto SET stock = stock + OLD.quantidade
    WHERE idproduto = OLD.idprodutoFK;
END $$
delimiter ;
```

-- Answer 17

```
delimiter $$
CREATE TRIGGER atualizaStockOnUpdate AFTER UPDATE ON itemcompra
FOR EACH ROW
BEGIN
    UPDATE produto SET stock = stock + OLD.quantidade - NEW.quantidade
    WHERE idproduto = OLD.idprodutoFK;
END $$
delimiter ;
```

-- Answer 18

```
delimiter $$
CREATE TRIGGER atualizaValorTotal2 after delete on itemcompra
FOR EACH ROW
BEGIN
    DECLARE total decimal(7,2);
    SELECT sum(quantidade * preco) into total
    FROM itemcompra
    WHERE idencomendaFK = OLD.idencomendaFK;

    UPDATE encomenda
    SET preco_total = total
    WHERE idencomenda = OLD.idencomendaFK;
END $$
delimiter ;
```