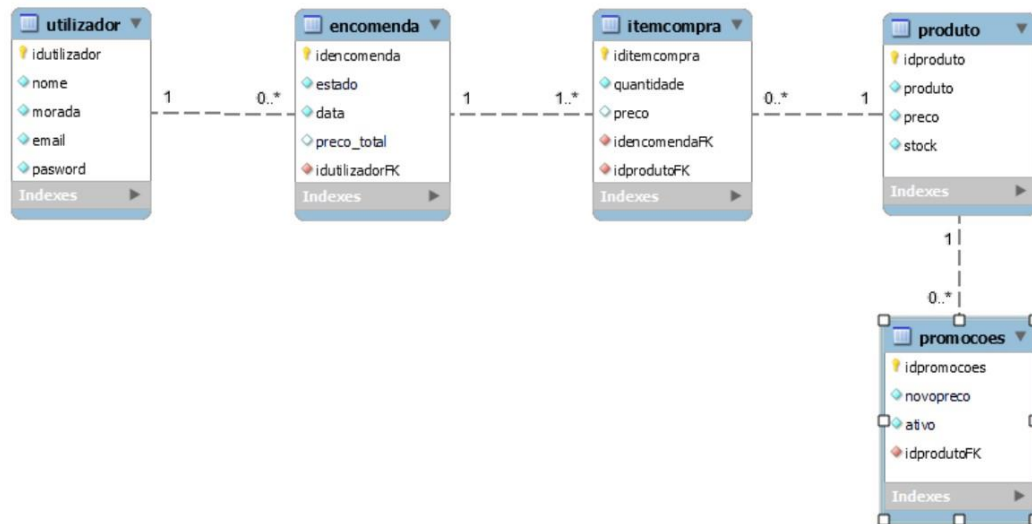


SQL QUERIES

Create this DataBase "LOJA":



-- 1. SQL code to see this query

	idproduto	produto	preco	stock	idpromocoies	novopreco	ativo	idprodutoFK
▶	1	Samsung A21	179.00	3	1	150.00	1	1
	2	Samsung S21	1000.00	4	NULL	NULL	NULL	NULL
	3	Iphone 12	829.00	5	NULL	NULL	NULL	NULL

```
SELECT * FROM loja.produto
```

```
LEFT JOIN loja.promocoies ON idproduto = idprodutoFK;
```

-- 2. SQL code to see this query

	idproduto	produto	preco	stock	promoção
▶	1	Samsung A21	179.00	3	150.00
	2	Samsung S21	1000.00	4	-
	3	Iphone 12	829.00	5	-

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

-- 3. SQL code to see this query

	produto	preço já com desconto aplicado!	stock
▶	Samsung A21	150.00€	3
	Samsung S21	1000.00€	4
	Iphone 12	829.00€	5

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

**-- 4. Create a view to save the last query (3) and name it
“produtos_com_promocoes”**

```
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;
```

-- 5.SQL query to select all the rows of the view “produtos_com_promocoes”

```
SELECT * FROM produtos_com_promocoes;
```

-- 6.SQL query to save SQL query of answer 3 as a stored procedure and name it "selecionar_produtos_com_preco_atualizado"

```
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;
```

-- 7.Who to call the last procedure

```
CALL selecionar_produtos_com_preco_atualizado();
```

-- 8.SQL query to save a stored procedure and name it "selecionar_produto_por_ID"

```
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);
```

-- 9.SQL query to save a stored procedure and name it "retornar_novo_preco_do_produto_porID()"

```
CREATE PROCEDURE retornar_novo_preco_do_produto_porID(id int unsigned)
BEGIN
    SELECT preco,
           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);
```

**-- 10.SQL query to save a stored procedure and name it
"inserir_encomenda_e_obter_idencomenda()**

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);

**-- 11.SQL query to insert a new "utilizador" with these values: "Carolina",
"Porto", carol@gmail.com, "321"**

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

-- 12.SQL query to see the total value of each sale as "total"

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

**13.SQL query to save a stored procedure and name it
"calcular_custototal_encomenda()"**

```
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)

-- 14.SQL query to save a stored procedure and name it

"atualizar_custototal_encomenda()"

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)

-- 15.SQL query to save a trigger to reduce the number of stock items when an item is sold and name it "atualizaStock1"

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;

-- 16.SQL query to save a trigger to delete the stock item when a sale is canceled and name it "atualizaStockonDelete"

CREATE TRIGGER atualizaStockonDelete AFTER DELETE ON itemcompra

FOR EACH ROW

BEGIN

UPDATE produto SET stock = stock + OLD.quantidade

WHERE idproduto = OLD.idprodutoFK;

END \$\$

delimiter ;

-- 17.SQL query to save a trigger to alter the stock item when a sale is done and name it "atualizaStockOnUpdate"

```
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;
```

-- 18.SQL query to save a trigger to alter the price sale when a item is deleted from the sale and name it "atualizaValorTotal2"

```
delimiter $$
CREATE TRIGGER atualizaValorTotal2 after deleting 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 ;
```