Universidade Federal de Itajubá

Atividade 2

COM231 – Banco de Dados 2

Alunos: Fábio Piovani Viviani 2017006774

Ygor Salles Aniceto Carvalho 2017014382

Professora: Vanessa Cristina Oliveira de Souza

Agosto

2020

Questão 1

a)

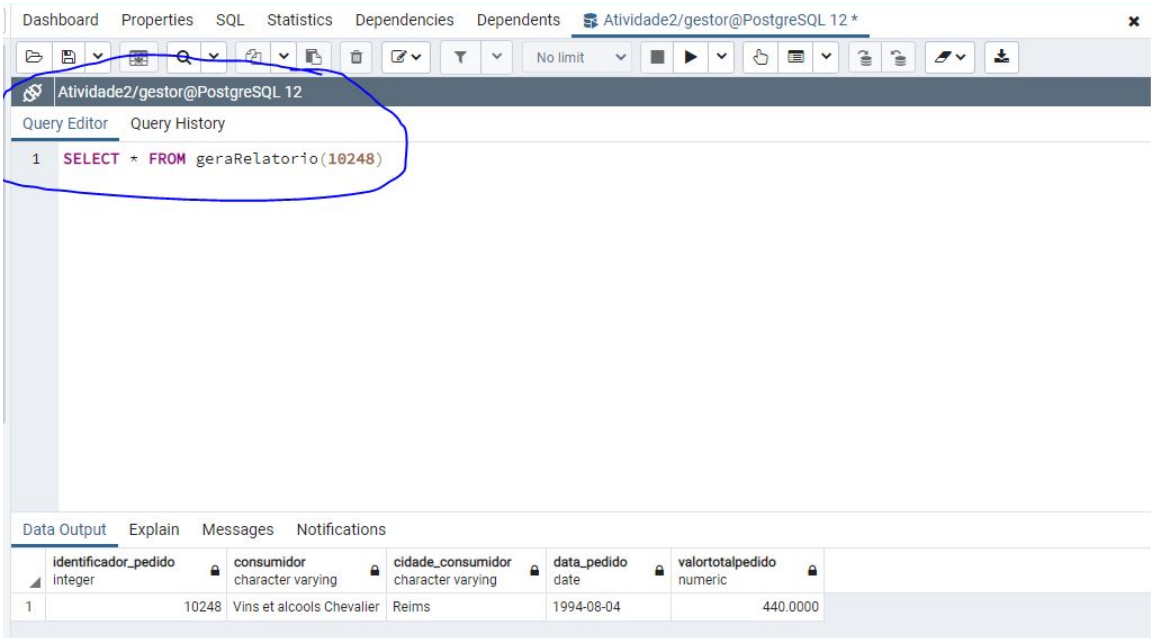
```
CREATE OR REPLACE FUNCTION geraRelatorio(pedido integer)
RETURNS TABLE (identificador_pedido int, consumidor varchar, cidade_consumidor varchar, data_pedido date, valortotalpedido numeric) AS $$
BEGIN
    ALTER TABLE northwind.orders
    ALTER COLUMN orderdate
    SET DATA TYPE date;
    RETURN QUERY SELECT o.orderid, c.companyname, c.city, o.orderdate, sum(od.unitprice * od.quantity)
    FROM northwind.orders o, northwind.customers c, northwind.order_details od
    WHERE o.customerid = c.customerid AND o.orderid = pedido AND od.orderid = pedido
    GROUP BY (o.orderid, c.customerid);
END;
$$ LANGUAGE 'plpgsql';

SELECT * FROM geraRelatorio(10248)
```

b)

```
GRANT EXECUTE ON FUNCTION geraRelatorio(pedido integer) TO gerente;
```

c) Testando a função com usuário gestor:



Questão 2

Pesquise sobre o chamado SQL Injection (o que é e como prevenir):

O SQL Injection é um ataque de vulnerabilidade. O mesmo se aproveita de formulários e inputs presentes em uma aplicação, para, através deles, injetar uma query SQL para tentar manipular, visualizar ou utilizar de informações consistentes no SGDB.

Uma maneira possível de se prevenir é o tratamento dos parâmetros antes de realizar a query, quando eles chegaram do formulário por um método POST. Outra prevenção é limitar quais usuários possuem acessos ao Banco e, desses usuários, quais suas devidas permissões.

Questão 3

Link para o vídeo:

https://www.youtube.com/watch?v=kMYevUUHWOU&feature=youtu.be

Link para o git com o código da aplicação:

https://github.com/fpviviani/pgsql-injection