

1- se o produto em estoque chegar ao valor 0 (não tem mais o produto) então o mesmo deve ser removido das tabelas Product e Stock. A aplicação faz o controle para o caso de quantity=0.

# Trigger 1

```
CREATE OR REPLACE FUNCTION quantidadeAlcancouZero() RETURNS  
trigger AS $$  
    BEGIN  
        IF NEW.quantity = 0 THEN  
            DELETE FROM product WHERE eid = NEW.eid;  
        END IF;  
        RETURN NEW;  
    END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER quantidadeAlcancouZero AFTER UPDATE ON stock FOR ROW  
EXECUTE PROCEDURE quantidadeAlcancouZero();
```

2 - Para evitar problemas com uso indevido de medicamentos controlados, crie uma tabela auxiliar de controle para armazenar quem está comprando cada medicamento com a data e hora. Ou seja, usuário que solicitou (logou no sistema), hora, eid (tabela product). Note que a aplicação irá fazer o update. O trigger deve apenas monitorar e armazenar as informações requisitadas.

## Trigger 2

```
CREATE OR REPLACE FUNCTION controleAcessoMedicamento() RETURNS
```

```
trigger AS $$
```

```
    BEGIN
```

```
        INSERT INTO controleAcessoMedicamento VALUES (current_user,  
current_timestamp, NEW.eid);
```

```
        RETURN NEW;
```

```
    END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER controleAcessoMedicamento AFTER UPDATE ON stock FOR ROW EXECUTE  
PROCEDURE controleAcessoMedicamento();
```

# Resolução

```
CREATE OR REPLACE FUNCTION process_booking_update()
RETURNS TRIGGER
AS $$
DECLARE
    lugares INTEGER;
    preco FLOAT;
BEGIN
    SELECT INTO lugares af.numberOfFreeSeats FROM AvailableFlights af WHERE af.Num_flight=NEW.Num_flight;
    SELECT INTO preco af.price FROM AvailableFlights af WHERE af.Num_flight=NEW.Num_flight;
    IF lugares=0 THEN
        RAISE EXCEPTION 'Nao existem mais lugares';
    END IF;
    UPDATE AvailableFlights SET numberOfFreeSeats=numberOfFreeSeats-1, price=price+50;
    NEW.price := preco;
    NEW.date := now();
RETURN NEW;
END $$ LANGUAGE 'plpgsql';
```

```
explain UPDATE available_flights AS a_flight SET free_seats = a_flight.free_seats - 1 WHERE a_flight.id = NEW.flight_id &&  
a_flight.free_seats > 0;
```

```
IF FOUND = false THEN
```

```
    RAISE EXCEPTION 'não tem voo';
```

```
END IF;
```

```
UPDATE available_flights AS a_flight SET price = a_flight.price + 50 WHERE a_flight.id = NEW.flight_id;
```

## Resolução

```
DROP TRIGGER IF EXISTS booking_update ON Bookings;
```

```
CREATE TRIGGER booking_update
```

```
BEFORE INSERT ON Bookings
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE process_booking_update();
```