

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.

```
CREATE OR REPLACE FUNCTION remove_product()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.quantity = 0 THEN
        DELETE FROM product WHERE id = NEW.id;
        DELETE FROM stock WHERE id = NEW.id;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_remove_product
AFTER UPDATE ON stock
FOR EACH ROW
EXECUTE FUNCTION remove_product();
```

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.

```
CREATE OR REPLACE FUNCTION record_medication_purchase()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO medication_purchase_log (user_id, product_id, purchase_time)
    VALUES (NEW.user_id, NEW.product_id, NOW());
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_record_medication_purchase
AFTER INSERT ON purchase_table -- Substitua "purchase_table" pelo nome da tabela de compras
FOR EACH ROW
EXECUTE FUNCTION record_medication_purchase();
```

