```
SELECT * FROM sql_inventory.products;

set session transaction isolation level serializable;

start transaction;

update sql_inventory.products
set quantity_in_stock = 79
where product_id = 2;

select * from sql_inventory.products
where product_id = 2;

update sql_inventory.products
set quantity_in_stock = 10
where product_id = 6;

commit;
rollback;

-- // serialable:
-- // if someone else has updated the value, ypou will not be able to even read it
-- // if no one else has updated the vale, you will be able to read it
-- // unless you read it via select ... for update


-- Algo for BMS
-- select * from show_seats where show_id = {} -- // to show the seat chart

-- when you make booking
-- start transaction
-- select * from show_seats where show_seat_id in (1, 3, 5) for update;
-- update show_seats
-- set seat_status = LOCKED
-- where show_seat_id in (1, 3, 5)
-- commit

-- if payment successful
-- update show_seats
-- set seat_status = BOOKED
-- where show_seat_id in (1, 3, 5)

-- if no payment
-- update show_seats
-- set seat_status = AVAILABLE
-- where show_seat_id in (1, 3, 5)
```