MySQL Exercise 7

Create the following 3 tables and insert sample data as shown:-

Ord_mst

Ord_no	Cust_cd	Status
1	C1	P

Ord_dtl

Ord_no	Prod_cd	Qty
1	P1	100
1	P2	200

Prod_mst

Prod_cd	Prod_name	Qty_in_stock	Booked_qty
P1	Floppies	10000	1000
P2	Printers	5000	600
P3	Modems	3000	200

1. Write a Before Insert trigger on Ord_dtl. Anytime a row is inserted in Ord_dtl, the Booked_qty in Prod_mst should be increased accordingly.

```
D2_92814_Krushna>CREATE TRIGGER before_insert_ord_dtl
    -> BEFORE INSERT ON Ord_dtl
    -> FOR EACH ROW
    -> BEGIN
    ->     UPDATE Prod_mst
    ->     SET booked_qty = booked_qty + NEW.qty
    ->     WHERE prod_cd = NEW.prod_cd;
    -> END //
Query OK, 0 rows affected (0.06 sec)
```

2. Write a Before Delete trigger on Ord_dtl. Anytime a row is deleted from Ord_dtl, the Booked_qty in Prod_mst should be decreased accordingly.

```
D2_92814_Krushna>DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE TRIGGER before_delete_ord_dtl
    -> BEFORE DELETE ON Ord_dtl
    -> FOR EACH ROW
    -> BEGIN
    ->     UPDATE Prod_mst
    ->     SET booked_qty = booked_qty - OLD.qty
    ->     WHERE prod_cd = OLD.prod_cd;
    -> END //
Query OK, 0 rows affected (0.01 sec)
```

3. Write a Before Update of Prod_cd, Qty trigger on Ord_dtl. Anytime the Prod_cd or Qty is updated, the Booked_qty in Prod_mst should be increased/decreased accordingly.

```
D2_92814_Krushna>CREATE TRIGGER trg_update_ord_dtl
    -> BEFORE UPDATE ON ord_dtl
    -> FOR EACH ROW
    -> BEGIN
    ->    -- If product changes
    ->    IF OLD.prod_cd <> NEW.prod_cd THEN
    ->       UPDATE prod_mst
    ->       SET booked_qty = booked_qty - OLD.qty
    ->       WHERE prod_cd = OLD.prod_cd;
    ->
    ->       UPDATE prod_mst
    ->       SET booked_qty = booked_qty + NEW.qty
    ->       WHERE prod_cd = NEW.prod_cd;
    ->
    ->    -- If only qty changes
    ->    ELSEIF OLD.qty <> NEW.qty THEN
    ->       UPDATE prod_mst
    ->       SET booked_qty = booked_qty - OLD.qty + NEW.qty
    ->       WHERE prod_cd = NEW.prod_cd;
    ->    END IF;
    -> END//
Query OK, 0 rows affected (0.05 sec)
```

4. Write a Before Update of Status trigger on Ord_mst. If the Status is updated from P (Pending) to D (Delivered), the Booked_qty and Qty_in_stock from Prod_mst should be decreased accordingly. If the Status is updated from P (Pending) to C (Cancelled), the details of the order should be deleted from Ord_dtl and corresponding Booked_qty from Prod_mst should be decreased accordingly. (The Before delete trigger on Ord_dtl would automatically decrease the Booked_qty from Prod_mst).

```
D2_92814_Krushna>CREATE TRIGGER trg_update_ordmst_status
    -> BEFORE UPDATE ON ord_mst
    -> FOR EACH ROW
    -> BEGIN
    ->     -- Case 1: Pending → Delivered
    ->     IF OLD.status = 'P' AND NEW.status = 'D' THEN
    ->        -- Decrease Qty_in_stock and Booked_qty
    ->        UPDATE prod_mst p
    ->        JOIN ord_dtl d ON p.prod_cd = d.prod_cd
    ->        SET p.qty_in_stock = p.qty_in_stock - d.qty,
    ->            p.booked_qty   = p.booked_qty - d.qty
    ->        WHERE d.ord_no = OLD.ord_no;
    ->     END IF;
    ->
    ->     -- Case 2: Pending → Cancelled
    ->     IF OLD.status = 'P' AND NEW.status = 'C' THEN
    ->        -- Delete details (Booked_qty auto-adjusted by delete trigger)
    ->        DELETE FROM ord_dtl
    ->        WHERE ord_no = OLD.ord_no;
    ->     END IF;
    -> END//
Query OK, 0 rows affected (0.05 sec)
```