

❖ EXERCISE 7:

- 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.

⇒ -- Create Ord_mst table

```
CREATE TABLE Ord_mst (  
    Ord_no INT PRIMARY KEY,  
    Cust_cd VARCHAR(10),  
    Status VARCHAR(10)  
);
```

-- Insert sample data into Ord_mst

```
INSERT INTO Ord_mst (Ord_no, Cust_cd, Status) VALUES  
(1, 'C1', 'P');
```

-- Create Ord_dtl table

```
CREATE TABLE Ord_dtl (  
    Ord_no INT,  
    Prod_cd VARCHAR(10),  
    Qty INT,  
    PRIMARY KEY (Ord_no, Prod_cd),  
    FOREIGN KEY (Ord_no) REFERENCES Ord_mst(Ord_no)  
);
```

-- Insert sample data into Ord_dtl

```
INSERT INTO Ord_dtl (Ord_no, Prod_cd, Qty) VALUES  
(1, 'P1', 100);
```

```
-- Create Prod_mst table
```

```
CREATE TABLE Prod_mst (  
    Prod_cd VARCHAR(10) PRIMARY KEY,  
    Prod_name VARCHAR(50),  
    Qty_in_stock INT,  
    Booked_qty INT  
);
```

```
-- Insert sample data into Prod_mst
```

```
INSERT INTO Prod_mst (Prod_cd, Prod_name, Qty_in_stock,  
Booked_qty) VALUES  
( 'P1', 'Floppies', 10000, 1000),  
( 'P2', 'Printers', 5000, 600),  
( 'P3', 'Modems', 3000, 200);  
DELIMITER //
```

```
CREATE TRIGGER before_insert_ord_dtl  
BEFORE INSERT ON Ord_dtl  
FOR EACH ROW  
BEGIN
```

```
    -- Update the Booked_qty in Prod_mst
```

```
    UPDATE Prod_mst
```

```
    SET Booked_qty = Booked_qty + NEW.Qty
    WHERE Prod_cd = NEW.Prod_cd;
END //
DELIMITER ;
INSERT INTO Ord_dtl (Ord_no, Prod_cd, Qty) VALUES (1, 'P1',
50);
```

- 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.

```
⇒ DELIMITER //
CREATE TRIGGER before_delete_ord_dtl
BEFORE DELETE ON Ord_dtl
FOR EACH ROW
BEGIN
    -- Update the Booked_qty in Prod_mst
    UPDATE Prod_mst
    SET Booked_qty = Booked_qty - OLD.Qty
    WHERE Prod_cd = OLD.Prod_cd;
END //
DELIMITER ;
DELETE FROM Ord_dtl WHERE Ord_no = 1 AND Prod_cd = 'P1';
SELECT * FROM Prod_mst;
```

- 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.

```
⇒ DELIMITER //
CREATE TRIGGER before_update_ord_dtl
```

```

BEFORE UPDATE ON Ord_dt1
FOR EACH ROW
BEGIN
    -- Decrease Booked_qty for the old product
    UPDATE Prod_mst
    SET Booked_qty = Booked_qty - OLD.Qty
    WHERE Prod_cd = OLD.Prod_cd;

    -- Increase Booked_qty for the new product if Prod_cd is changing
    IF OLD.Prod_cd <> NEW.Prod_cd THEN
        UPDATE Prod_mst
        SET Booked_qty = Booked_qty + NEW.Qty
        WHERE Prod_cd = NEW.Prod_cd;
    ELSE
        -- If Prod_cd is not changing, just adjust the booked quantity
        UPDATE Prod_mst
        SET Booked_qty = Booked_qty + (NEW.Qty - OLD.Qty)
        WHERE Prod_cd = OLD.Prod_cd;
    END IF;
END //

DELIMITER ;

UPDATE Ord_dt1
SET Prod_cd = 'P2', Qty = 120
WHERE Ord_no = 1 AND Prod_cd = 'P1';

```

```
SELECT * FROM Prod_mst;
```

- 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).

⇒ DELIMITER //

```
CREATE TRIGGER before_update_ord_mst
```

```
BEFORE UPDATE ON Ord_mst
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Check if the status is changing from 'P' (Pending) to 'D' (Delivered)
```

```
IF OLD.Status = 'P' AND NEW.Status = 'D' THEN
```

```
-- Decrease Qty_in_stock and Booked_qty accordingly for each detail
```

```
UPDATE Prod_mst
```

```
SET Qty_in_stock = Qty_in_stock - (SELECT SUM(Qty)  
FROM Ord_dtl WHERE Ord_no = NEW.Ord_no),
```

```
Booked_qty = Booked_qty - (SELECT SUM(Qty) FROM  
Ord_dtl WHERE Ord_no = NEW.Ord_no)
```

```
WHERE Prod_cd IN (SELECT Prod_cd FROM Ord_dtl  
WHERE Ord_no = NEW.Ord_no);
```

```

-- Check if the status is changing from 'P' (Pending) to 'C'
(Cancelled)

ELSEIF OLD.Status = 'P' AND NEW.Status = 'C' THEN
    -- Decrease Booked_qty for each detail before deleting
    UPDATE Prod_mst
        SET Booked_qty = Booked_qty - (SELECT SUM(Qty) FROM
Ord_dt1 WHERE Ord_no = NEW.Ord_no)
        WHERE Prod_cd IN (SELECT Prod_cd FROM Ord_dt1
WHERE Ord_no = NEW.Ord_no);

    -- Delete details from Ord_dtl
    DELETE FROM Ord_dt1 WHERE Ord_no = NEW.Ord_no;

END IF;
END //

```

```

DELIMITER ;

UPDATE Ord_mst
SET Status = 'D'
WHERE Ord_no = 1;

UPDATE Ord_mst
SET Status = 'C'
WHERE Ord_no = 1;

SELECT * FROM Prod_mst;

```