**TASK 1:** **Create a trigger on products table, check if the newly added MSRP is less than 100 update the value to 555.**

USE northwind;

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '123456';

CREATE TABLE products1 (

productCode varchar(15) NOT NULL,

productName varchar(70) NOT NULL,

productLine varchar(50) NOT NULL,

productScale varchar(10) NOT NULL,

productVendor varchar(50) NOT NULL,

productDescription text NOT NULL,

quantityInStock int(11) NOT NULL,

buyPrice decimal(10, 2) NOT NULL,

MSRP decimal(10, 2) NOT NULL,

PRIMARY KEY (productCode)

);

DELIMITER //

CREATE TRIGGER check\_MSRP

BEFORE INSERT ON products1

FOR EACH ROW

BEGIN

DECLARE new\_MSRP DECIMAL(10, 2);

SET new\_MSRP = NEW.MSRP;

IF new\_MSRP < 100 THEN

SET NEW.MSRP = 555;

END IF;

END //

DELIMITER ;

INSERT INTO products1 (productCode, productName, productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP)

VALUES ('PAK001', 'Pakistani', 'Food', '1:10', 'Pakistani Food Inc.', 'Traditional Pakistani cuisine', 100, 10.99, 99.99);

SELECT \* FROM PRODUCTS1

**Creating a trigger that logs changes to an audit table whenever a record is inserted, updated, or deleted from the employees table.**

CREATE TRIGGER audit\_trigger

AFTER INSERT OR UPDATE OR DELETE ON employees

FOR EACH ROW

BEGIN

DECLARE action VARCHAR(50);

IF INSERTING THEN

SET action = 'INSERT';

ELSIF UPDATING THEN

SET action = 'UPDATE';

ELSE

SET action = 'DELETE';

END IF;

INSERT INTO audit\_table (user, action, timestamp)

VALUES (CURRENT\_USER(), action, NOW());

END;

// to remove trigger

DROP TRIGGER audit\_trigger;

**Creating an Event:**

CREATE EVENT daily\_backup

ON SCHEDULE

EVERY 1 DAY

STARTS CURRENT\_TIMESTAMP

DO

BEGIN

-- SQL statements to perform daily backup

INSERT INTO backup\_table (backup\_time, backup\_files)

SELECT NOW(), file\_name

FROM important\_data\_table;

END;

CREATE TRIGGER after\_insert\_order

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

-- Update total\_sales in products table when a new order is inserted

UPDATE products

SET total\_sales = total\_sales + NEW.quantity \* NEW.unit\_price

WHERE product\_id = NEW.product\_id;

END;