Trigger:  
  fire  
  Database:  
     a trigger is a special type of stored procedure  
     that is automatically executed or fired   
     in response to certain events or actions   
      occurring in the database.  
    What are those events ?  
    from which context ?----------------entity-----table  
      CRUD (data manipulation)Operations:  
        INSERT----insert event  
            before insert----trigger--SP  
            after insert-----trigger--SP

        UPDATE----update event  
            before update ------trigger--SP  
            after update -----trigger--SP

        DELETE       
            before update ------trigger--SP  
            after update -----trigger--SP  
   Triggers  are used  to enforce business rules,   
          maintaining data integrity,  
          automating tasks within database

   Triggers are used  to perform validation checks,  
               audit checks  
               propogating another execution cycles in respective related tables

  Key Points about Triggers:  
      Event-Based Execution:  
      Timing: BEfore or After  (DATA Manipulation) ( Update, Insert, Delete)  
      Scope: table level scope:  
      Atomicity:

CREATE database HRDB;  
use HRDB:

CREATE TABLE employees (  
    id INT AUTO\_INCREMENT PRIMARY KEY,  
    name VARCHAR(100),  
    department VARCHAR(100),  
    last\_updated TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
);

-- create a trigger named update\_last\_updated that updates the last\_updated column  
--  whenever a row is inserted or updated in the employees table

DELIMITER//

CREATE TRIGGER update\_last\_updated\_on\_update  
BEFORE UPDATE ON employees  
FOR EACH ROW  
BEGIN  
    SET NEW.last\_updated = CURRENT\_TIMESTAMP;  
END;  
//  
DELIMITER ;

-- after   defining table   try to write  DML queries to test triggers are working  
-- as per expectation of buisiness logic

-- Insert a new employee  
INSERT INTO employees (name, department) VALUES ('Ravi Tambade', 'training');

-- Update existing record

UPDATE  employees  SET department='BOD" WHERE id=1;

SELECT \* FROM employees;

 -- eCommerce  Solution  
-- orders :  details about customer orders  
-- inventory : information about product stock  
-- Order processing System

-- When an order is placed, the inventory is automatically  updated to  
-- reflect  new updated stock  
create database sample\_ecommerce;  
use sample\_ecommerce;

CREATE TABLE orders (  
    order\_id INT AUTO\_INCREMENT PRIMARY KEY,  
    product\_id INT,  
    quantity INT,  
    order\_date DATETIME,  
    status ENUM('pending', 'completed', 'canceled')  
);

CREATE TABLE inventory (  
    product\_id INT PRIMARY KEY,  
    stock\_quantity INT  
);

-- user story:  
-- When an order is placed, the inventory is automatically  updated to  
-- reflect  new updated stock

DELIMITER $$  
CREATE TRIGGER after\_order\_insert  
AFTER  INSERT ON  orders  
FOR EACH ROW  
BEGIN  
  DECLARE available\_stock INT;  
    SELECT stock\_quantity INTO available\_stock  
    FROM inventory  
    WHERE product\_id = NEW.product\_id;  
     IF available\_stock IS NOT NULL AND available\_stock >= NEW.quantity THEN  
    UPDATE  inventory  
    SET  stock\_quantity=stock\_quantity-NEW.quantity  
    WHERE  product\_id=NEW.product\_id;  
     ELSE  
    SIGNAL SQLSTATE '45000'   
        SET MESSAGE\_TEXT= 'Insufficient stock for the product';  
     END IF;  
END $$  
DELIMITER ;

-- Test trigger execution:

-- pre-requisite  
INSERT INTO inventory (product\_id,stock\_quantity)  
values(  (1,56),  
    (2,78),  
    (5,00));  
  
-- Insert a new order  
INSERT INTO orders (product\_id, quantity, order\_date, status)  
Values (1, 3, NOW(), 'pending');