CREATE DATABASE triggerpractice_3

-- BEFORE INSERT Trigger

```
CREATE TABLE Student_Trigger
(
Student_RollNo INT NOT NULL PRIMARY KEY,
Student_FirstName Varchar (100),
Student_EnglishMarks INT,
Student_PhysicsMarks INT,
Student_ChemistryMarks INT,
Student_MathsMarks INT,
Student_TotalMarks INT,
Student_Percentage INT
);

CREATE TRIGGER Student_Table_Marks
BEFORE INSERT  ON Student_Trigger
FOR EACH ROW
SET new.Student_TotalMarks = new.Student_EnglishMarks + new.Student_PhysicsMarks +
new.Student_ChemistryMarks + new.Student_MathsMarks,
new.Student_Percentage = ( new.Student_TotalMarks / 400) * 100;

INSERT INTO Student_Trigger (Student_RollNo, Student_FirstName, Student_EnglishMarks,
Student_PhysicsMarks, Student_ChemistryMarks, Student_MathsMarks, Student_TotalMarks,
Student_Percentage)
VALUES ( 201, 'Sorya', 88, 75, 69, 92, 0, 0);
```

Student_RollNo	Student_FirstName	Student_EnglishMarks	Student_PhysicsMarks	Student_ChemistryMarks	Student_MathsMarks	Student_TotalMarks	Student_Percentage
201	Sorya	88	75	69	92	324	81

-- AFTER INSERT Trigger

```
CREATE TABLE Student_Trigger_New
(
   Student_RollNo INT NOT NULL PRIMARY KEY,
   Student_FirstName VARCHAR(100),
   Student_EnglishMarks INT,
   Student_PhysicsMarks INT,
   Student_ChemistryMarks INT,
   Student_MathsMarks INT,
   Student_TotalMarks INT DEFAULT 0,
   Student_Percentage INT DEFAULT 0
);
```

```
CREATE TABLE Student_Trigger_Calculation (
   Student_RollNo INT PRIMARY KEY,
   Student_TotalMarks INT,
   Student_Percentage INT
);

DELIMITER //

CREATE OR REPLACE TRIGGER Student_Table_Marks_After_Insert
AFTER INSERT ON Student_Trigger_New
FOR EACH ROW
BEGIN
   -- Insert into the calculation table with computed total and percentage
   INSERT INTO Student_Trigger_Calculation (Student_RollNo, Student_TotalMarks,
Student_Percentage)
   VALUES (
      NEW.Student_RollNo,
      NEW.Student_EnglishMarks + NEW.Student_PhysicsMarks + NEW.Student_ChemistryMarks +
NEW.Student_MathsMarks,
      ((NEW.Student_EnglishMarks + NEW.Student_PhysicsMarks + NEW.Student_ChemistryMarks +
NEW.Student_MathsMarks) / 400) * 100
   );
END;
//

DELIMITER ;

-- To update the actual student table (in your application or as a separate query), use:
UPDATE Student_Trigger_New AS st
JOIN Student_Trigger_Calculation AS calc ON st.Student_RollNo = calc.Student_RollNo
SET
   st.Student_TotalMarks = calc.Student_TotalMarks,
   st.Student_Percentage = calc.Student_Percentage
WHERE st.Student_RollNo = calc.Student_RollNo;

-- Insert test data
INSERT INTO Student_Trigger_New (Student_RollNo, Student_FirstName, Student_EnglishMarks,
Student_PhysicsMarks, Student_ChemistryMarks, Student_MathsMarks)
VALUES (202, 'Sorya', 88, 75, 69, 92);

-- Example to verify if the calculation works correctly after trigger firing
SELECT * FROM Student_Trigger_New;
SELECT * FROM Student_Trigger_Calculation;
```

```
-- CODE ONLY FOR PREVIOUS ONE
/*
CREATE TABLE Student_Trigger_New
(
   Student_RollNo INT NOT NULL PRIMARY KEY,
   Student_FirstName VARCHAR(100),
   Student_EnglishMarks INT,
   Student_PhysicsMarks INT,
   Student_ChemistryMarks INT,
   Student_MathsMarks INT,
   Student_TotalMarks INT DEFAULT 0,
   Student_Percentage INT DEFAULT 0
);

CREATE TABLE Student_Trigger_Calculation (
   Student_RollNo INT PRIMARY KEY,
   Student_TotalMarks INT,
   Student_Percentage INT
);

DELIMITER //

CREATE OR REPLACE TRIGGER Student_Table_Marks_After_Insert
AFTER INSERT ON Student_Trigger_New
FOR EACH ROW
BEGIN
   -- Insert into the calculation table with computed total and percentage
   INSERT INTO Student_Trigger_Calculation (Student_RollNo, Student_TotalMarks,
Student_Percentage)
   VALUES (
      NEW.Student_RollNo,
      NEW.Student_EnglishMarks + NEW.Student_PhysicsMarks + NEW.Student_ChemistryMarks +
NEW.Student_MathsMarks,
      ((NEW.Student_EnglishMarks + NEW.Student_PhysicsMarks + NEW.Student_ChemistryMarks +
NEW.Student_MathsMarks) / 400) * 100
   );
END;
//

DELIMITER ;
```

```
-- To update the actual student table (in your application or as a separate query), use:
UPDATE Student_Trigger_New AS st
JOIN Student_Trigger_Calculation AS calc ON st.Student_RollNo = calc.Student_RollNo
SET
   st.Student_TotalMarks = calc.Student_TotalMarks,
   st.Student_Percentage = calc.Student_Percentage
WHERE st.Student_RollNo = calc.Student_RollNo;

-- Insert test data
INSERT INTO Student_Trigger_New (Student_RollNo, Student_FirstName, Student_EnglishMarks,
Student_PhysicsMarks, Student_ChemistryMarks, Student_MathsMarks)
VALUES (203, 'Sorya', 88, 75, 69, 92);

-- Example to verify if the calculation works correctly after trigger firing
SELECT * FROM Student_Trigger_New;
SELECT * FROM Student_Trigger_Calculation;

*/
```

-- Before INSERT Trigger

```
DELIMITER //

CREATE OR REPLACE TRIGGER Student_Table_Marks_Before_Insert
BEFORE INSERT ON Student_Trigger_New
FOR EACH ROW
BEGIN
   -- Calculate total marks and percentage before inserting into the main table
   SET NEW.Student_TotalMarks = NEW.Student_EnglishMarks + NEW.Student_PhysicsMarks +
NEW.Student_ChemistryMarks + NEW.Student_MathsMarks;
   SET NEW.Student_Percentage = (NEW.Student_TotalMarks / 400) * 100;

   -- Insert the calculated total and percentage into the calculation table
   INSERT INTO Student_Trigger_Calculation (Student_RollNo, Student_TotalMarks,
Student_Percentage)
   VALUES (
      NEW.Student_RollNo,
      NEW.Student_TotalMarks,
      NEW.Student_Percentage
   );
END;
//

DELIMITER ;
```

```
-- Insert test data
INSERT INTO Student_Trigger_New (Student_RollNo, Student_FirstName, Student_EnglishMarks,
Student_PhysicsMarks, Student_ChemistryMarks, Student_MathsMarks)
VALUES (204, 'Sorya', 88, 75, 69, 92);


-- Check the results
SELECT * FROM Student_Trigger_New;
SELECT * FROM Student_Trigger_Calculation;

-- After insert trigger in one single table
CREATE TABLE students (
   student_id INT PRIMARY KEY AUTO_INCREMENT,
   first_name VARCHAR(100),
   last_name VARCHAR(100),
   enrollment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

/* -- We will create a trigger that runs after a new student is inserted, and if
the enrollment_date is NULL, it will set it to the current timestamp. */

DELIMITER $$

CREATE TRIGGER after_student_insert
AFTER INSERT ON students
FOR EACH ROW
BEGIN
   -- Check if the enrollment_date is NULL
   IF NEW.enrollment_date IS NULL THEN
      UPDATE students
      SET enrollment_date = CURRENT_TIMESTAMP
      WHERE student_id = NEW.student_id;
   END IF;
END$$

DELIMITER ;

-- Insert Data
INSERT INTO students (first_name, last_name)
VALUES ('John', 'Doe');

--
Verify Output
SELECT * FROM students;
```

```
INSERT INTO students (first_name, last_name, enrollment_date )
VALUES ('John', 'Doe', '2024-11-12 10:57:11');
```

Verify Output

```
SELECT * FROM students;
```