To fulfill the requirements for the \*\*before trigger\*\* for both the `INSERT` and `UPDATE` events on the `Emp` table, we will create a trigger that:

1. Rejects the insertion of a new employee if the salary is less than Rs. 50,000.

2. Rejects the update of an employee's salary if the new salary is less than Rs. 50,000.

3. Stores the attempted values (employee number and salary) in a new `Tracking` table if the salary is less than Rs. 50,000.

### 1. \*\*Create the `Emp` Table and `Tracking` Table\*\*

Before writing the triggers, we will create the necessary tables: `Emp` (for employee data) and `Tracking` (to store rejected attempts).

```sql

-- Create Emp table

CREATE TABLE Emp (

e\_no INT PRIMARY KEY,

e\_name VARCHAR(100),

salary DECIMAL(10, 2)

);

-- Create Tracking table to store attempted invalid salary insertions/updates

CREATE TABLE Tracking (

e\_no INT,

salary DECIMAL(10, 2),

action\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

```

### 2. \*\*Before Insert Trigger\*\*

This trigger will fire \*\*before\*\* an insert operation on the `Emp` table. If the salary value being inserted is less than Rs. 50,000, the trigger will reject the insert and log the attempted insert into the `Tracking` table.

```sql

DELIMITER $$

CREATE TRIGGER before\_insert\_emp

BEFORE INSERT ON Emp

FOR EACH ROW

BEGIN

-- Check if salary is less than 50000

IF NEW.salary < 50000 THEN

-- Insert the attempted insert data into the Tracking table

INSERT INTO Tracking (e\_no, salary)

VALUES (NEW.e\_no, NEW.salary);

-- Signal that the insert operation should be rejected

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Salary must be at least 50,000';

END IF;

END $$

DELIMITER ;

```

### Explanation:

- `BEFORE INSERT`: This ensures that the trigger is fired before the insertion of a new row in the `Emp` table.

- `NEW.salary`: The `NEW` keyword is used to access the values of the row that is about to be inserted. We check if the `salary` is less than 50,000.

- `SIGNAL SQLSTATE '45000'`: This statement generates an error message, effectively rejecting the `INSERT` operation if the salary condition is met.

- If the salary is less than Rs. 50,000, the trigger inserts the attempted `e\_no` and `salary` into the `Tracking` table and raises an error to reject the insert operation.

### 3. \*\*Before Update Trigger\*\*

This trigger will fire \*\*before\*\* an update operation on the `Emp` table. If the salary being updated is less than Rs. 50,000, the trigger will reject the update and log the attempted update into the `Tracking` table.

```sql

DELIMITER $$

CREATE TRIGGER before\_update\_emp

BEFORE UPDATE ON Emp

FOR EACH ROW

BEGIN

-- Check if the new salary is less than 50000

IF NEW.salary < 50000 THEN

-- Insert the attempted update data into the Tracking table

INSERT INTO Tracking (e\_no, salary)

VALUES (NEW.e\_no, NEW.salary);

-- Signal that the update operation should be rejected

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Salary must be at least 50,000';

END IF;

END $$

DELIMITER ;

```

### Explanation:

- `BEFORE UPDATE`: This trigger is fired before an update operation on the `Emp` table.

- `NEW.salary`: We access the new value of the `salary` column after the update operation but before the data is actually updated in the table.

- If the salary is less than Rs. 50,000, the trigger logs the attempted update into the `Tracking` table and raises an error to reject the update operation using the `SIGNAL` statement.

### 4. \*\*Test the Triggers\*\*

Now that we have defined the triggers, we can test them by trying to insert and update records with salaries below Rs. 50,000.

#### Test 1: \*\*Insert with Salary Below 50,000\*\*

```sql

-- Attempt to insert a record with salary less than 50,000

INSERT INTO Emp (e\_no, e\_name, salary)

VALUES (1, 'John Doe', 45000);

```

- This insert will be rejected, and an entry will be made in the `Tracking` table with `e\_no = 1` and `salary = 45000`.

#### Test 2: \*\*Update with Salary Below 50,000\*\*

```sql

-- Attempt to update salary to a value less than 50,000

UPDATE Emp

SET salary = 40000

WHERE e\_no = 1;

```

- This update will also be rejected, and an entry will be made in the `Tracking` table with the `e\_no = 1` and `salary = 40000`.

#### Check the `Tracking` Table:

After performing the above insert and update operations, you can check the `Tracking` table to see the entries that were rejected:

```sql

SELECT \* FROM Tracking;

```

You should see the following output (assuming the tests were executed as described):

| e\_no | salary | action\_time |

|------|---------|----------------------|

| 1 | 45000 | 2024-11-11 12:34:56 |

| 1 | 40000 | 2024-11-11 12:35:10 |

### Conclusion:

- The \*\*before insert\*\* and \*\*before update\*\* triggers work to prevent salary values less than Rs. 50,000 from being inserted or updated in the `Emp` table.

- In case the salary is invalid (i.e., less than Rs. 50,000), the action is rejected, and the attempt is logged into the `Tracking` table.

- The `SIGNAL SQLSTATE '45000'` statement is used to reject the operation with a custom error message.

This solution ensures that only employees with a salary greater than or equal to Rs. 50,000 are inserted or updated in the `Emp` table.