OUTPUT:

CREATING TRIGGER BEFORE INSERTING INTO TABLE

```
SQL> CREATE TABLE STUDENTS(
  2  ID NUMBER PRIMARY KEY,
  3  NAME VARCHAR(15),
  4  COURSE VARCHAR(10),
  5  SCORE NUMBER
  6  );

Table created.

SQL> CREATE OR REPLACE TRIGGER before_insert_students
  2  BEFORE INSERT ON students
  3  FOR EACH ROW
  4  BEGIN
  5      IF :NEW.score > 100 THEN
  6          :NEW.score := 100;
  7      END IF;
  8  END;
  9  /

Trigger created.

SQL> INSERT INTO students (id, name, course, score) VALUES (1, 'Alice', 'Python', 95);

1 row created.

SQL> INSERT INTO students (id, name, course, score) VALUES (2, 'Bob', 'Python', 105);

1 row created.

SQL> SELECT * FROM STUDENTS;

    ID       NAME       COURSE     SCORE
---------- ---------- ---------- ----------
     1       Alice       Python        95
     2       Bob         Python       100
```

CREATING TRIGGER BEFORE UPDATING DATA FROM TABLE

```
SQL> CREATE TABLE employees (
  2      employee_id NUMBER PRIMARY KEY,
  3      employee_name VARCHAR2(50),
  4      salary NUMBER(10, 2),
  5      department VARCHAR2(50)
  6  );

Table created.
```

```
SQL> CREATE OR REPLACE TRIGGER before_update_employee
  2  BEFORE UPDATE ON employees
  3  FOR EACH ROW
  4  BEGIN
  5      DBMS_OUTPUT.PUT_LINE('Updating employee ' || :OLD.employee_name || ' in department ' ||
             :OLD.department);
  6  END;
  7  /

Trigger created.

SQL> INSERT INTO employees (employee_id, employee_name, salary, department)
  2  VALUES (1, 'John Doe', 50000, 'Engineering');

1 row created.

SQL> UPDATE employees
  2  SET salary = 55000
  3  WHERE employee_id = 1;

Updating employee John Doe in department Engineering

1 row updated.
```

CREATING TRIGGER BEFORE DELETING DATA FROM TABLE

```
SQL> CREATE TABLE orders (
  2      order_id NUMBER PRIMARY KEY,
  3      customer_name VARCHAR2(100),
  4      order_date DATE,
  5      total_amount NUMBER(10, 2)
  6  );

Table created.

SQL> CREATE OR REPLACE TRIGGER before_delete_order
  2  BEFORE DELETE ON orders
  3  FOR EACH ROW
  4  BEGIN
  5      DBMS_OUTPUT.PUT_LINE('Deleting order ' || :OLD.order_id || ' for customer ' ||
             :OLD.customer_name);
  6  END;
  7  /

Trigger created.

SQL> INSERT INTO orders (order_id, customer_name, order_date, total_amount) VALUES (1, 'Jane
Smith', SYSDATE, 150.00);

1 row created.
```

SQL> DELETE FROM orders WHERE order_id = 1;

Deleting order 1 for customer Jane Smith

1 row deleted.