

```
DBMS_OUTPUT.PUT_LINE('Doctor: ' || doctor_name);  
END;
```

Triggers

1. Write a trigger to check age validity of a customer using row level triggers.
(Age should not be less than 20)

```
CREATE OR REPLACE TRIGGER age_check_trigger  
BEFORE INSERT OR UPDATE ON customer  
FOR EACH ROW  
BEGIN  
    IF :NEW.age < 20 THEN  
        RAISE_APPLICATION_ERROR(-20001, 'Age must be at least 20.');
```

2. Create a Trigger for one instance of student table it will update another table while inserting values.

```
CREATE OR REPLACE TRIGGER student_update_trigger  
AFTER INSERT ON student  
FOR EACH ROW  
BEGIN  
    UPDATE other_table  
    SET related_column = :NEW.student_column  
    WHERE condition_column = :NEW.student_id;  
END;
```

3. Create a row level after trigger on customer table.

```
CREATE OR REPLACE TRIGGER log_customer_insert  
AFTER INSERT ON customer  
FOR EACH ROW  
BEGIN
```

```
INSERT INTO customer_log (cust_id, log_date)
VALUES (:NEW.cust_id, SYSDATE);
END;
```

4. Create a statement level trigger in employee table.

```
CREATE OR REPLACE TRIGGER log_employee_update
AFTER UPDATE ON employee
BEGIN
    INSERT INTO employee_log (log_message, log_date)
    VALUES ('An update occurred on employee table', SYSDATE);
END;
```

5. Create an after trigger to update rows in book relation

```
CREATE OR REPLACE TRIGGER update_book_rows
AFTER UPDATE ON book
FOR EACH ROW
BEGIN
    UPDATE book
    SET last_modified = SYSDATE
    WHERE book_id = :OLD.book_id;
END;
```