```
SQL> CREATE OR REPLACE TRIGGER empl
  2  AFTER INSERT OR UPDATE OR DELETE ON employee7
  3  FOR EACH ROW
  4  DECLARE
  5  BEGIN
  6    IF INSERTING THEN
  7      INSERT INTO nemployee7 VALUES (:new.id, :new.name, :new.sal);
  8    ELSIF UPDATING THEN
  9      INSERT INTO cemployee7 VALUES (:new.id, :new.name, :new.sal);
 10    ELSIF DELETING THEN
 11      INSERT INTO xemployee7 VALUES (:old.id, :old.name, :old.sal);
 12    END IF;
 13  END;
 14  /

Trigger created.

SQL> INSERT INTO EMPLOYEE7 VALUES(102,'Jesto',300000);

1 row created.

SQL>  select * from xemployee7;

no rows selected

SQL>  select * from nemployee7;

        ID NAME                          SAL
---------- -------------------- ----------
       102 Jesto                    300000

SQL>
SQL> update employee7 set sal=100000 where id=102;

1 row updated.

SQL>  select * from cemployee7;

        ID NAME                          SAL
---------- -------------------- ----------
       102 Jesto                    100000

SQL> delete from  employee7 where id=102;

1 row deleted.

SQL>  select * from xemployee7;

        ID NAME                          SAL
---------- -------------------- ----------
       102 Jesto                    100000
```

```
SQL> create or replace trigger tr_restrict
  2  before
  3  delete or insert or update on employee711
  4  begin
  5  if(rtrim(to_char(sysdate,'DAY')) in ('saturday', 'sunday')) or
  6  (to_char(sysdate, 'hh24:mi') not between '08:30' and '18:30') then
  7   dbms_output.put_line('table unaccessible');
  8  end if;
  9  end;
 10  /

Trigger created.

SQL>  INSERT INTO EMPLOYEE711 VALUES(103,'Abey',300000);
table unaccessible
```

```
SQL> CREATE OR REPLACE PROCEDURE addBook( bname IN varchar2, aut IN varchar2,tot IN number)
  2  IS
  3  BEGIN
  4
  5    INSERT INTO Bookrec VALUES(bname,aut,tot,tot);
  6
  7  END;
  8  /

Procedure created.

SQL>
SQL> CREATE OR REPLACE TRIGGER BOOKIN
  2  AFTER INSERT ON Bookrec
  3  FOR EACH ROW
  4  DECLARE
  5    ID NUMBER;
  6  BEGIN
  7   SELECT BookID INTO ID FROM BookDetails1 WHERE ROWNUM=1 ORDER BY BookID DESC;
  8   IF ID is NULL THEN
  9    ID:=100;
 10   ELSE
 11    ID:=ID+1;
 12   END IF;
 13
 14   INSERT INTO BookDetails1 VALUES(ID,:new.BookName,0);
 15  END;
 16  /

Trigger created.

SQL>
SQL> select * from BookDetails1;

    BOOKID BOOKNAME              MEMBERID
---------- -------------------- ----------
       101 hello                         0
       102 Life                          0
       103 car                           0
       104 Rings                         0

SQL> select * from Bookrec;

BOOKNAME             AUTHOR               TOTALCOPIES AVAILABLECOPIES
-------------------- -------------------- ----------- ---------------
hello                Peter                        120             118
Life                 Denny                        100              99
car                  Will                          25              24
Rings                Sarah                        100              99

SQL>
SQL> begin
  2   addBook('Hollow','Robert',50);
  3  end;
  4  /

PL/SQL procedure successfully completed.

Commit complete.
SQL>
SQL> select * from BookDetails1;

    BOOKID BOOKNAME              MEMBERID
---------- -------------------- ----------
       101 hello                         0
       102 Life                          0
       103 car                           0
       104 Rings                         0
       105 Hollow                        0

SQL> select * from Bookrec;

BOOKNAME             AUTHOR               TOTALCOPIES AVAILABLECOPIES
-------------------- -------------------- ----------- ---------------
hello                Peter                        120             118
Life                 Denny                        100              99
Hollow               Robert                        50              50
car                  Will                          25              24
Rings                Sarah                        100              99

SQL> CREATE OR REPLACE FUNCTION issue_book (book_name IN VARCHAR2, member_id IN NUMBER)
  2  RETURN DATE
  3  IS
  4    l_issue_date DATE;
  5    l_return_date DATE;
  6    l_available_copies NUMBER;
  7    id number;
  8  BEGIN
  9    SELECT Availablecopies INTO l_available_copies FROM Bookrec WHERE BookName = book_name;
 10    SELECT BookID INTO id FROM BookDetails1 WHERE BookName = book_name;
 11    IF l_available_copies > 0 THEN
 12      l_issue_date := SYSDATE;
 13      l_return_date := l_issue_date + INTERVAL '30' DAY;
 14   INSERT INTO CirculationRec VALUES (id,member_id, l_issue_date, l_return_date);
 15
 16   UPDATE BookDetails1 SET MemberID=member_id WHERE bookname=book_name;
 17
 18
 19    RETURN l_return_date;
 20    ELSE
 21      RAISE_APPLICATION_ERROR(-20001, 'Book not available');
 22    END IF;
 23  END issue_book;
 24  /

Function created.

SQL> CREATE OR REPLACE TRIGGER updateavailable_copies
  2  AFTER update ON BookDetails1
  3  FOR EACH ROW
  4  BEGIN
  5    UPDATE bookrec
  6    SET availablecopies = availablecopies - 1
  7    WHERE bookname = :NEW.bookname;
  8  END update_available_copies;
  9  /

Trigger created.

SQL> DECLARE
  2    return_date DATE;
  3  BEGIN
  4    return_date := issue_book('Life',1);
  5    DBMS_OUTPUT.PUT_LINE('Return date: ' || return_date);
  6  END;
  7  /
Return date: 11-MAR-23

PL/SQL procedure successfully completed.

Commit complete.
SQL>
SQL> select * from BookDetails1;

    BOOKID BOOKNAME              MEMBERID
---------- -------------------- ----------
       101 hello                         0
       102 Life                          1
       103 car                           0
       104 Rings                         0
       105 Hollow                        0

SQL> select * from Bookrec;

BOOKNAME             AUTHOR               TOTALCOPIES AVAILABLECOPIES
-------------------- -------------------- ----------- ---------------
hello                Peter                        120             118
Life                 Denny                        100              96
Hollow               Robert                        50              50
car                  Will                          25              24
Rings                Sarah                        100              99

SQL> select * from CirculationRec;

    BOOKID   MEMBERID ISSUEDATE RETURNDAT
---------- ---------- --------- ---------
       102          1 09-FEB-23 11-MAR-23
```