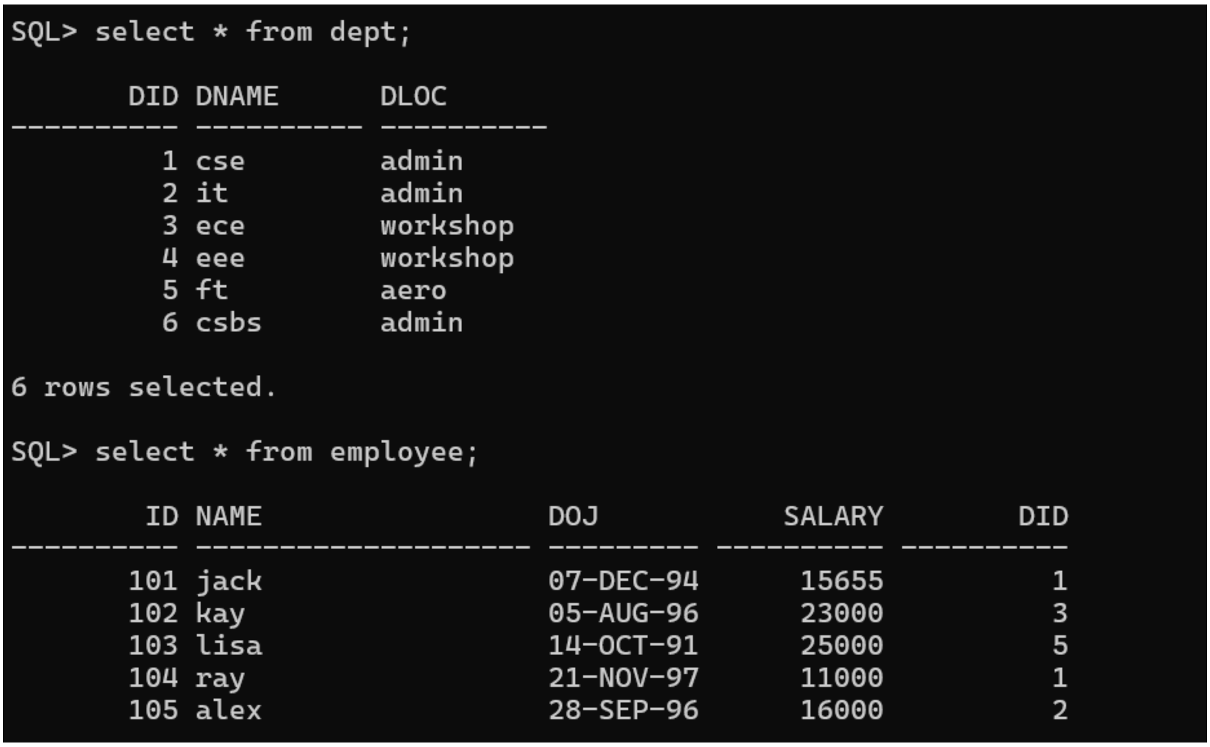
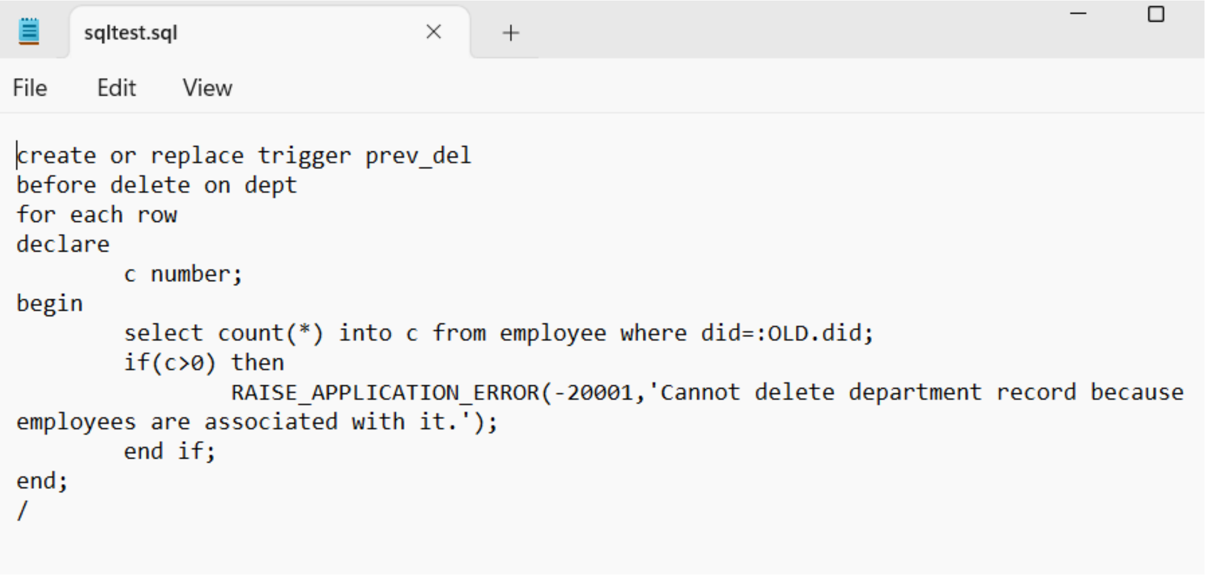
Exercise 18

Program 1

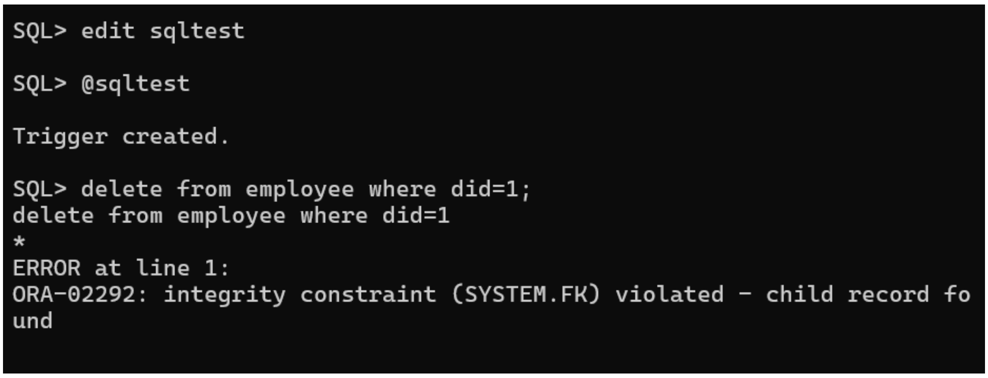
Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist.



```
SQL> select * from dept;

       DID DNAME      DLOC
---------- ---------- ----------
         1 cse        admin
         2 it         admin
         3 ece        workshop
         4 eee        workshop
         5 ft         aero
         6 csbs       admin

6 rows selected.

SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      15655          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2
```



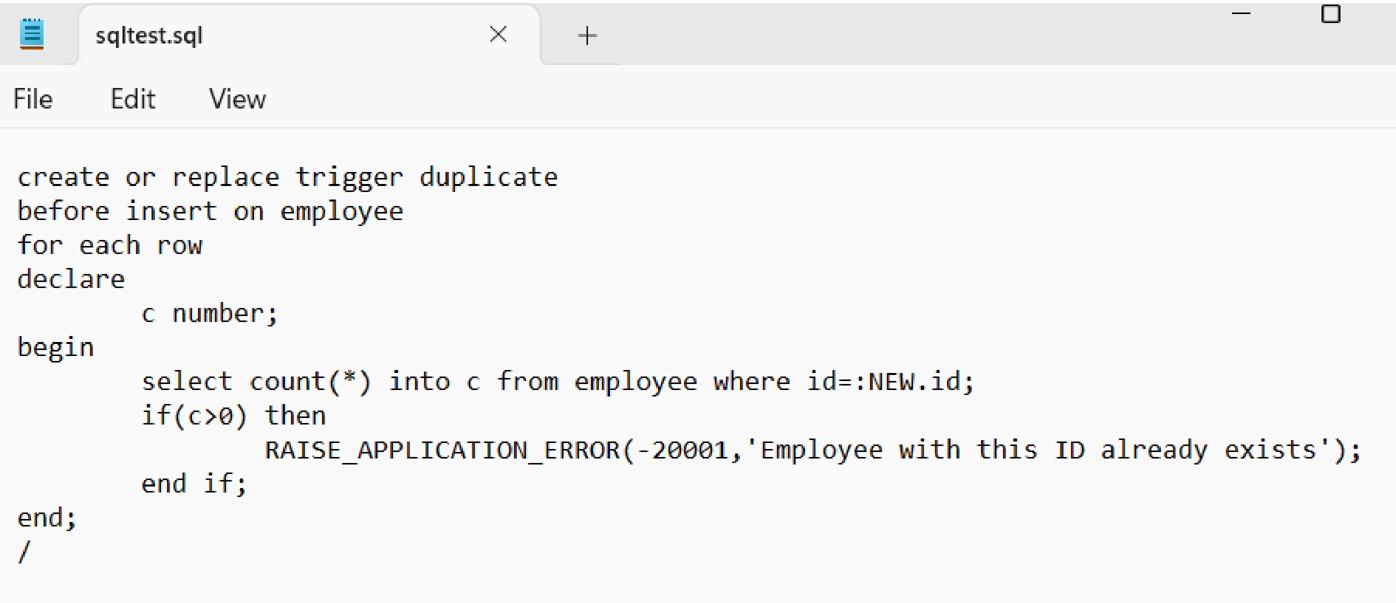
```
create or replace trigger prev_del
before delete on dept
for each row
declare
        c number;
begin
        select count(*) into c from employee where did=:OLD.did;
        if(c>0) then
                RAISE_APPLICATION_ERROR(-20001,'Cannot delete department record because
employees are associated with it.');
        end if;
end;
/
```



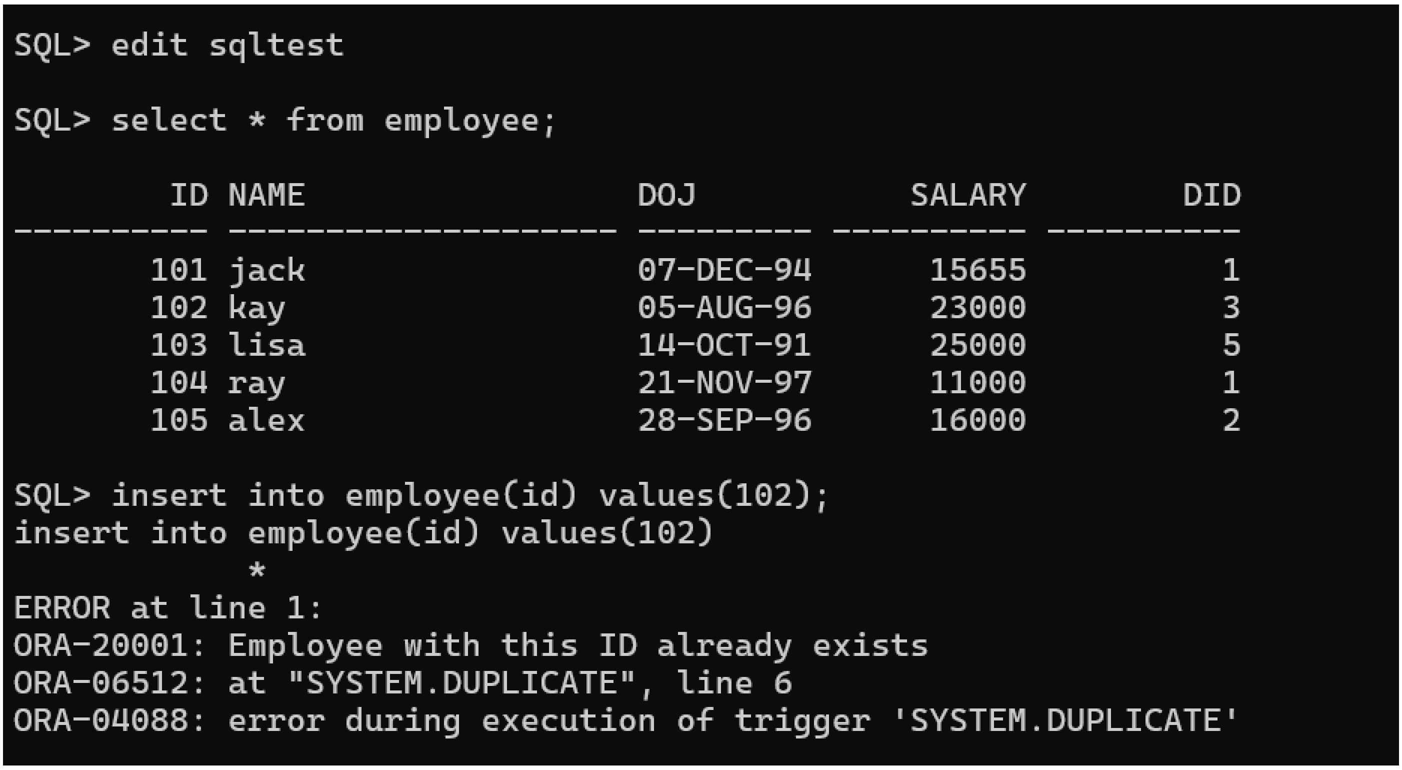
```
SQL> edit sqltest

SQL> @sqltest

Trigger created.

SQL> delete from employee where did=1;
delete from employee where did=1
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.FK) violated - child record fo
und
```

Program 2

Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found.



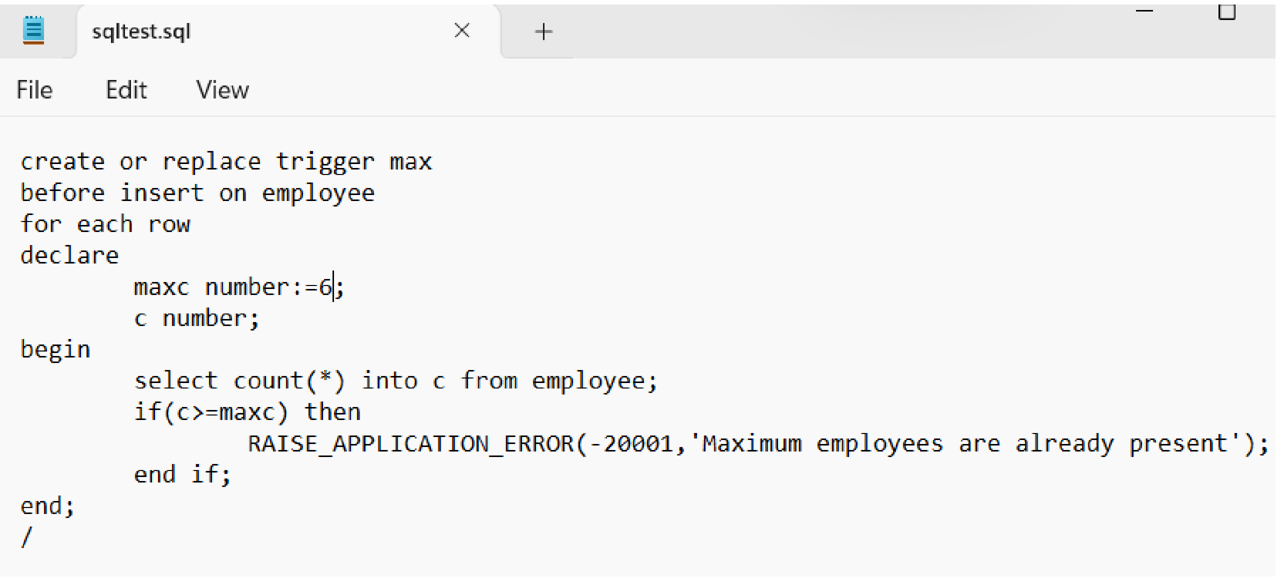
```
create or replace trigger duplicate
before insert on employee
for each row
declare
        c number;
begin
        select count(*) into c from employee where id=:NEW.id;
        if(c>0) then
                RAISE_APPLICATION_ERROR(-20001,'Employee with this ID already exists');
        end if;
end;
/
```



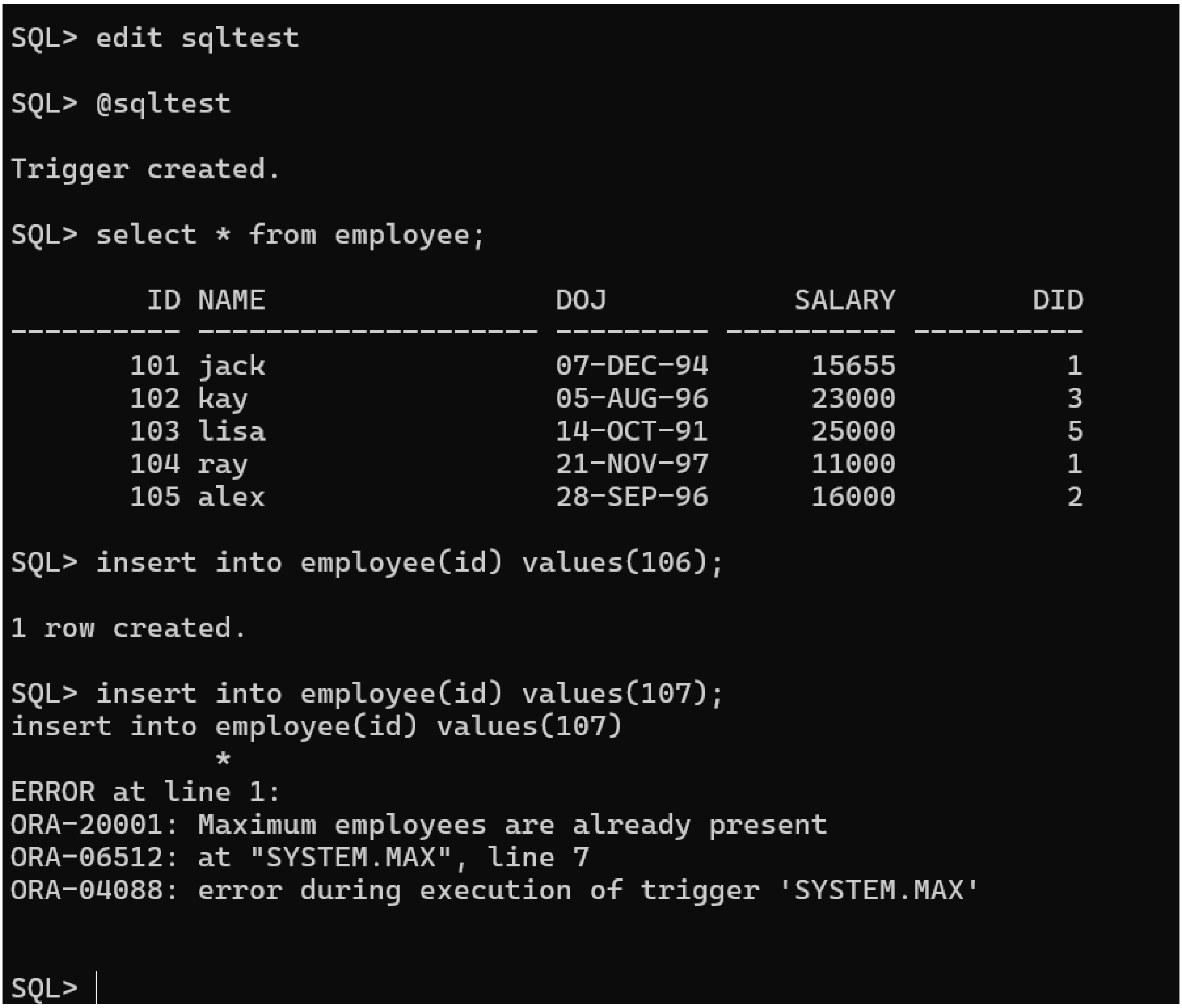
```
SQL> edit sqltest

SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      15655          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2

SQL> insert into employee(id) values(102);
insert into employee(id) values(102)
            *
ERROR at line 1:
ORA-20001: Employee with this ID already exists
ORA-06512: at "SYSTEM.DUPLICATE", line 6
ORA-04088: error during execution of trigger 'SYSTEM.DUPLICATE'
```

Program 3

Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold.



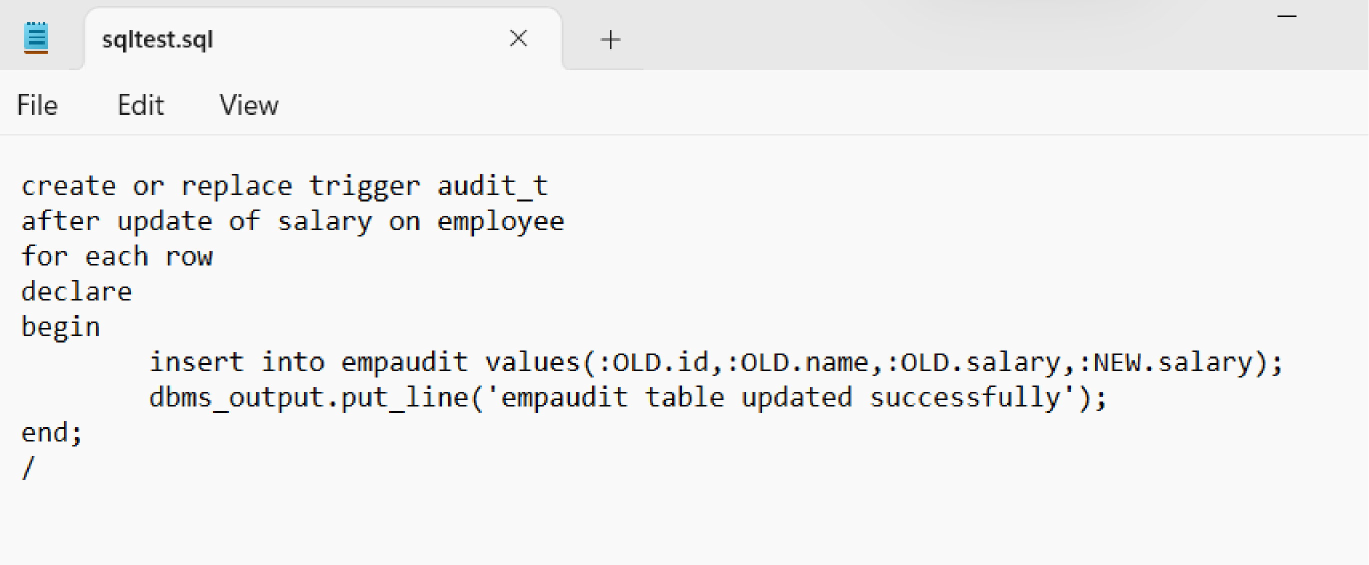
```
create or replace trigger max
before insert on employee
for each row
declare
        maxc number:=6;
        c number;
begin
        select count(*) into c from employee;
        if(c>=maxc) then
                RAISE_APPLICATION_ERROR(-20001,'Maximum employees are already present');
        end if;
end;
/
```



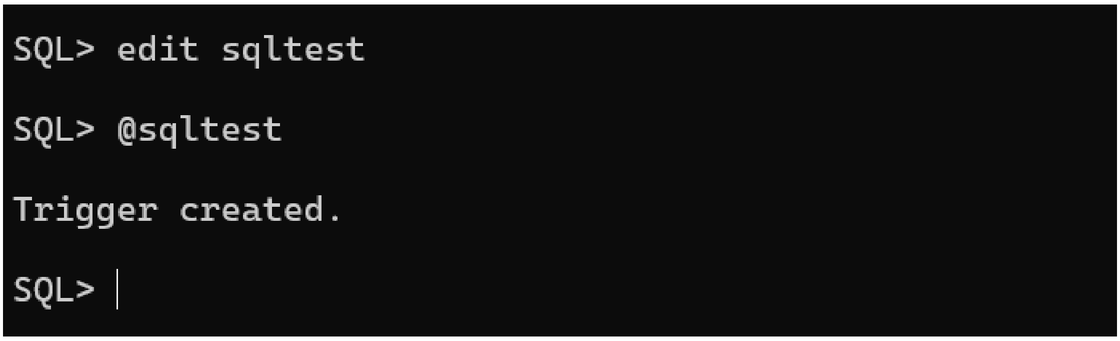
```
SQL> edit sqltest

SQL> @sqltest

Trigger created.

SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      15655          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2

SQL> insert into employee(id) values(106);

1 row created.

SQL> insert into employee(id) values(107);
insert into employee(id) values(107)
            *
ERROR at line 1:
ORA-20001: Maximum employees are already present
ORA-06512: at "SYSTEM.MAX", line 7
ORA-04088: error during execution of trigger 'SYSTEM.MAX'


SQL>
```

Program 4

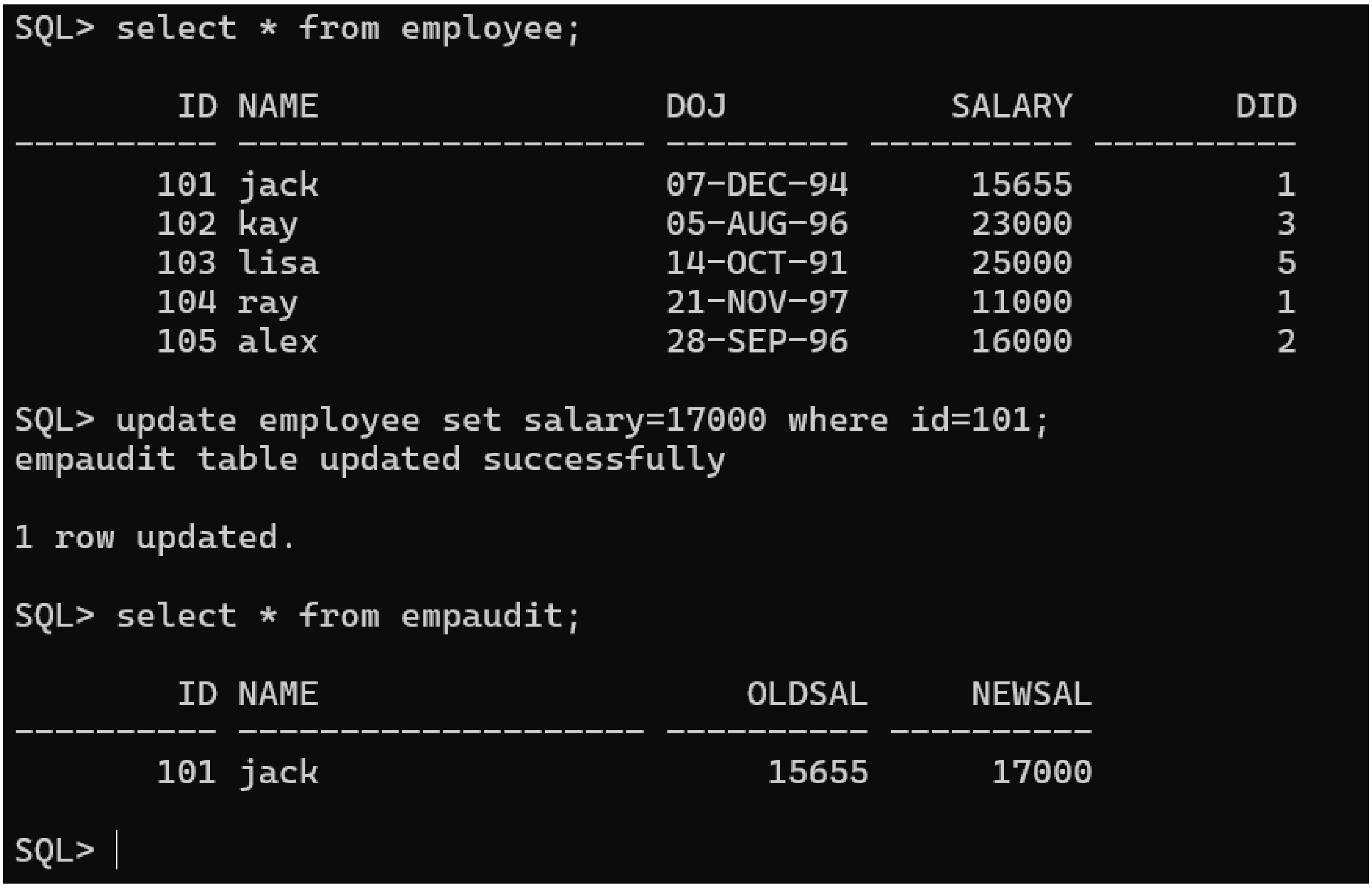
Write a code in PL/SQL to design a trigger that captures changes made to specific columns and logs them in an audit table.



```
create or replace trigger audit_t
after update of salary on employee
for each row
declare
begin
        insert into empaudit values(:OLD.id,:OLD.name,:OLD.salary,:NEW.salary);
        dbms_output.put_line('empaudit table updated successfully');
end;
/
```



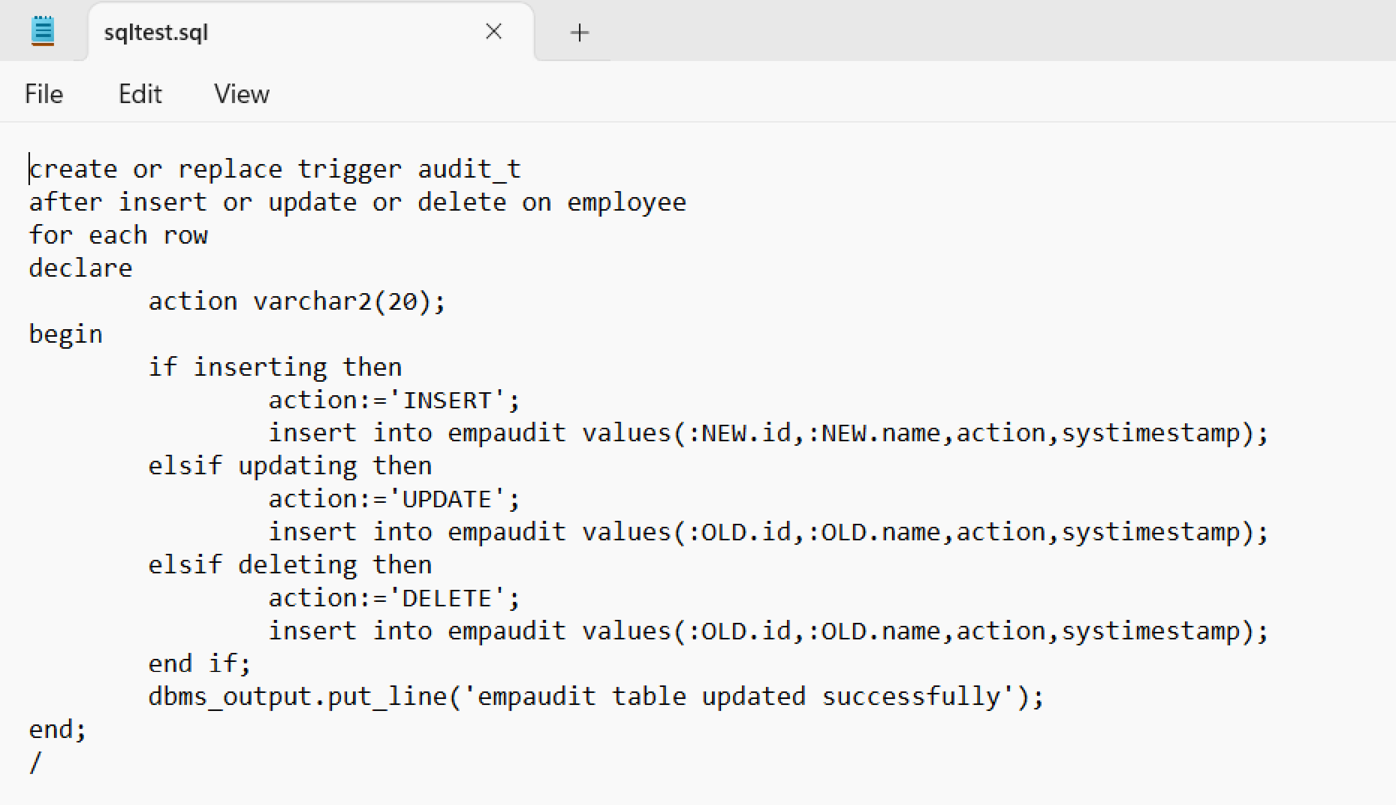
```
SQL> edit sqltest

SQL> @sqltest

Trigger created.

SQL>
```



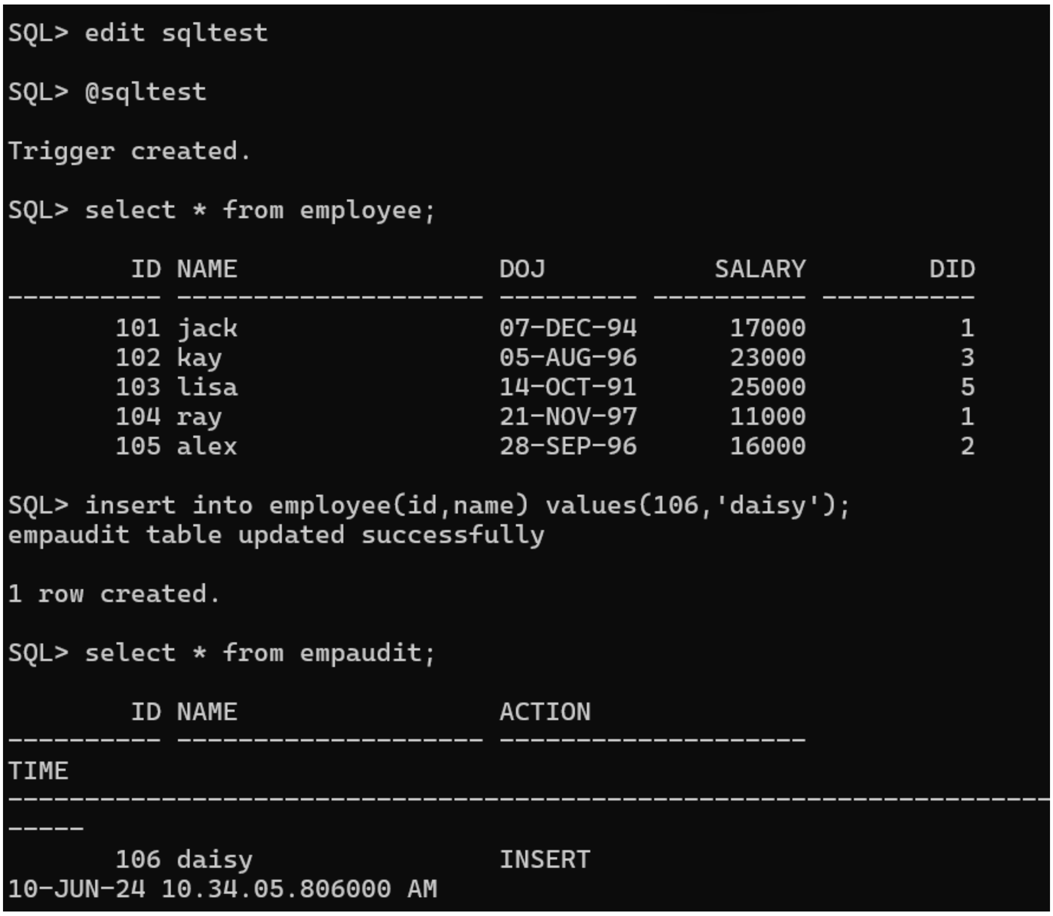
```
SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      15655          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2

SQL> update employee set salary=17000 where id=101;
empaudit table updated successfully

1 row updated.

SQL> select * from empaudit;

        ID NAME                     OLDSAL     NEWSAL
---------- -------------------- ---------- ----------
       101 jack                      15655      17000

SQL>
```

Program 5

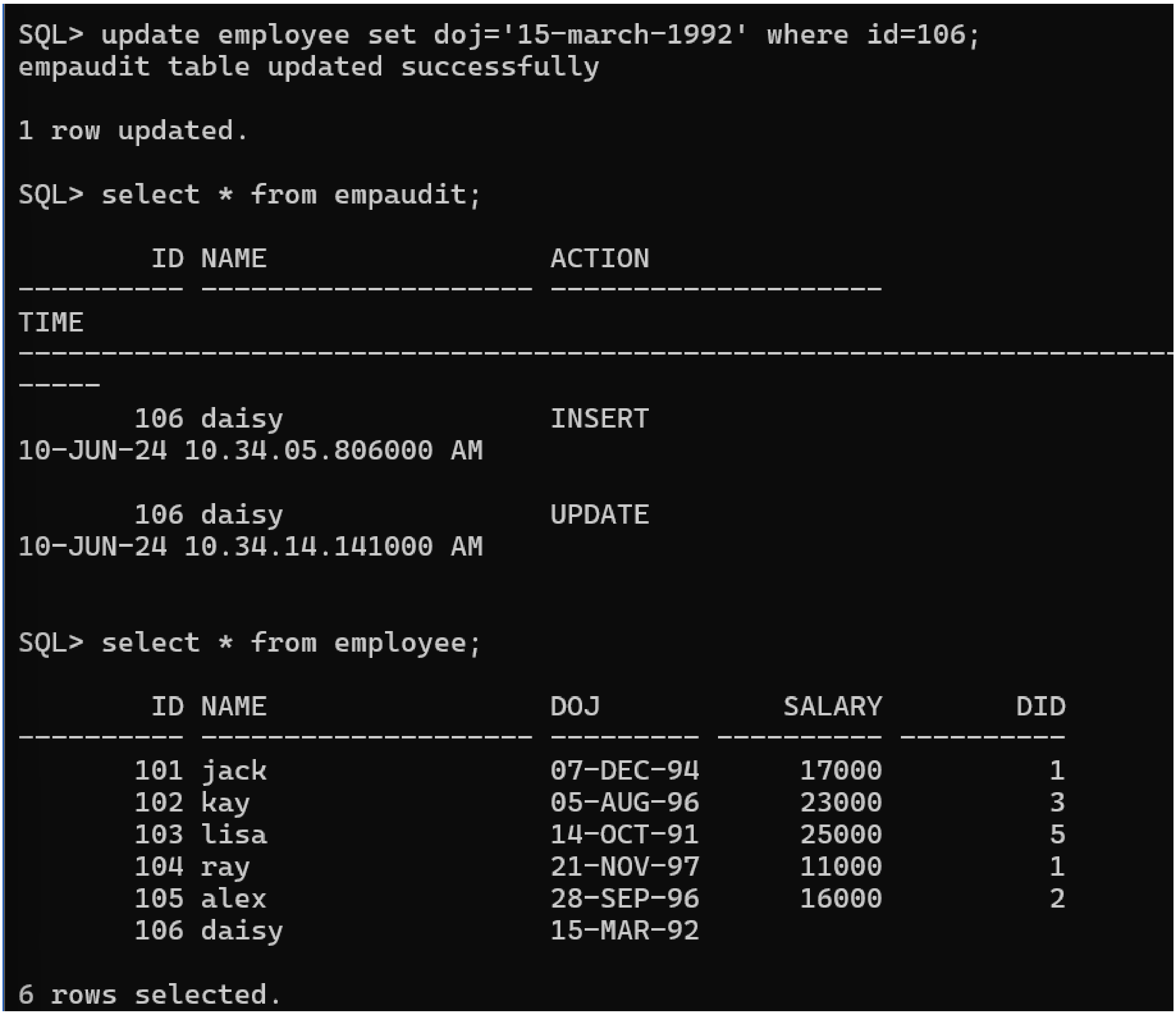
Write a code in PL/SQL to implement a trigger that records user activity (inserts, updates, deletes) in an audit log for a given set of tables.



```
create or replace trigger audit_t
after insert or update or delete on employee
for each row
declare
        action varchar2(20);
begin
        if inserting then
                action:='INSERT';
                insert into empaudit values(:NEW.id,:NEW.name,action,systimestamp);
        elsif updating then
                action:='UPDATE';
                insert into empaudit values(:OLD.id,:OLD.name,action,systimestamp);
        elsif deleting then
                action:='DELETE';
                insert into empaudit values(:OLD.id,:OLD.name,action,systimestamp);
        end if;
        dbms_output.put_line('empaudit table updated successfully');
end;
/
```



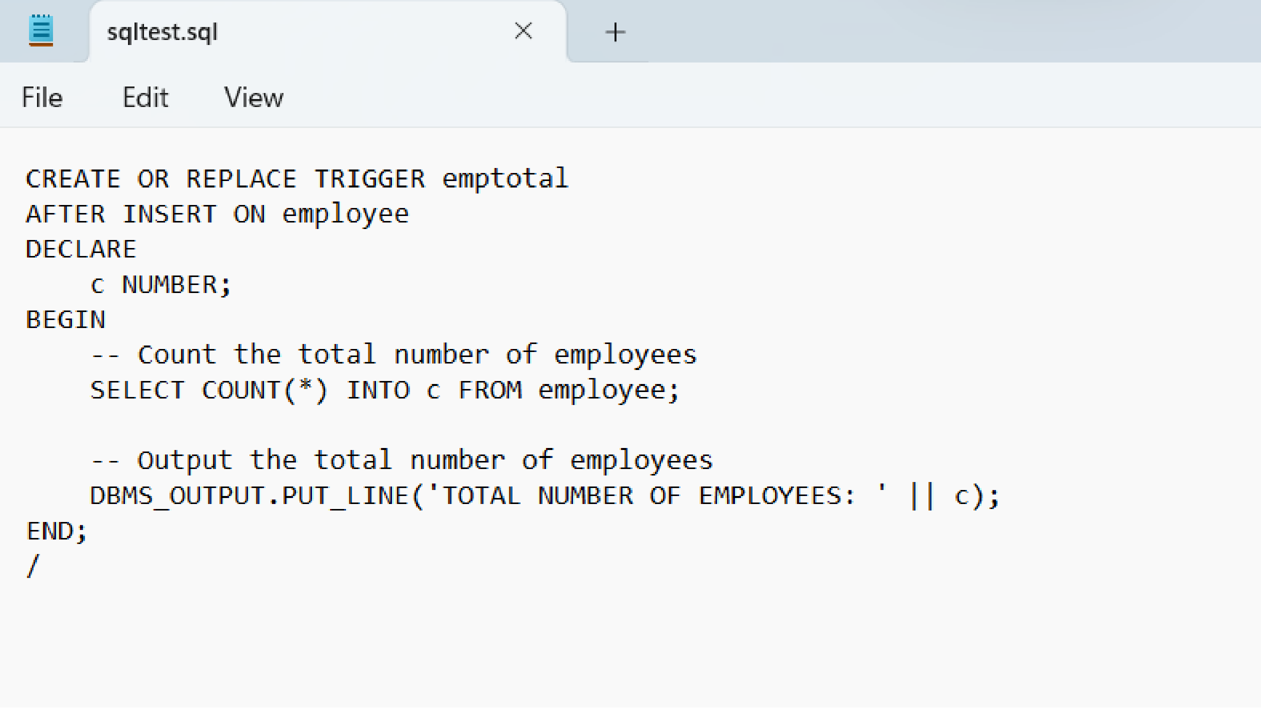
```
SQL> edit sqltest

SQL> @sqltest

Trigger created.

SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      17000          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2

SQL> insert into employee(id,name) values(106,'daisy');
empaudit table updated successfully

1 row created.

SQL> select * from empaudit;

        ID NAME                 ACTION
---------- -------------------- --------------------
TIME
---------------------------------------------------------------------------
       106 daisy                INSERT
10-JUN-24 10.34.05.806000 AM
```



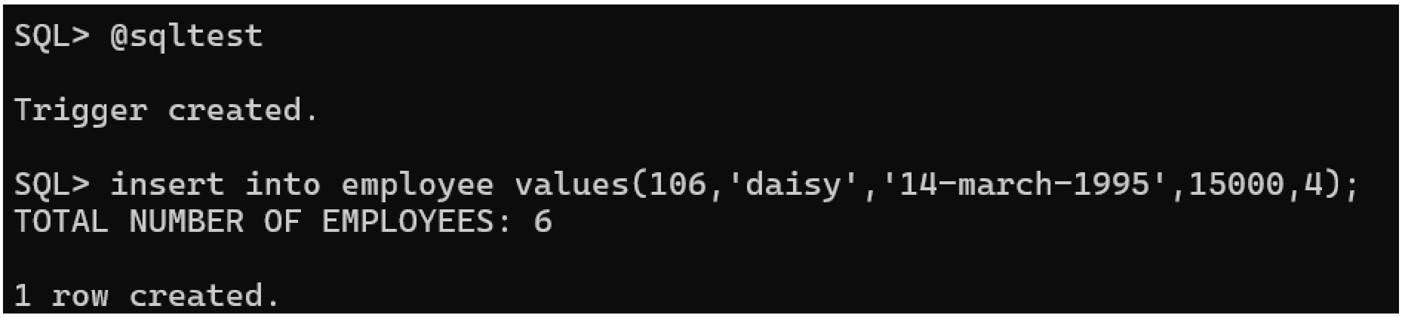
```
SQL> update employee set doj='15-march-1992' where id=106;
empaudit table updated successfully

1 row updated.

SQL> select * from empaudit;

        ID NAME                 ACTION
---------- -------------------- --------------------
TIME
----------------------------------------------------------------------
-----
       106 daisy                INSERT
10-JUN-24 10.34.05.806000 AM

       106 daisy                UPDATE
10-JUN-24 10.34.14.141000 AM


SQL> select * from employee;

        ID NAME                 DOJ           SALARY        DID
---------- -------------------- --------- ---------- ----------
       101 jack                 07-DEC-94      17000          1
       102 kay                  05-AUG-96      23000          3
       103 lisa                 14-OCT-91      25000          5
       104 ray                  21-NOV-97      11000          1
       105 alex                 28-SEP-96      16000          2
       106 daisy                15-MAR-92

6 rows selected.
```

Program 7

Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted.



```
CREATE OR REPLACE TRIGGER emptotal
AFTER INSERT ON employee
DECLARE
    c NUMBER;
BEGIN
    -- Count the total number of employees
    SELECT COUNT(*) INTO c FROM employee;

    -- Output the total number of employees
    DBMS_OUTPUT.PUT_LINE('TOTAL NUMBER OF EMPLOYEES: ' || c);
END;
/
```



```
SQL> @sqltest

Trigger created.

SQL> insert into employee values(106,'daisy','14-march-1995',15000,4);
TOTAL NUMBER OF EMPLOYEES: 6

1 row created.
```