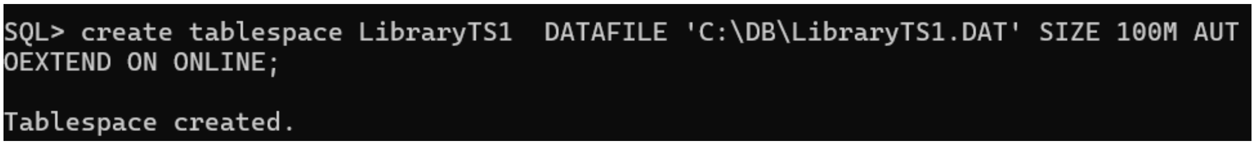
DATABASE LAB TEST

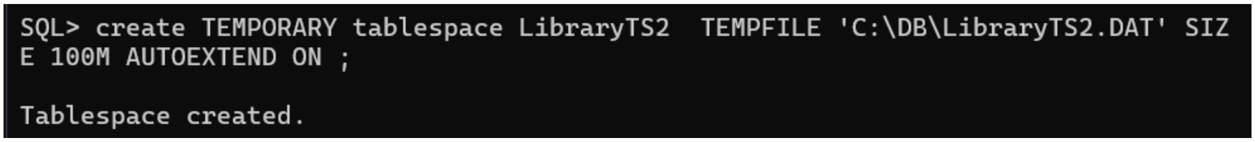
ZEGHILET AFRAH

GROUP 3

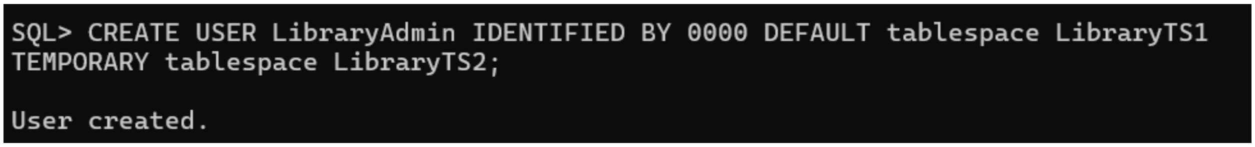
PART 1



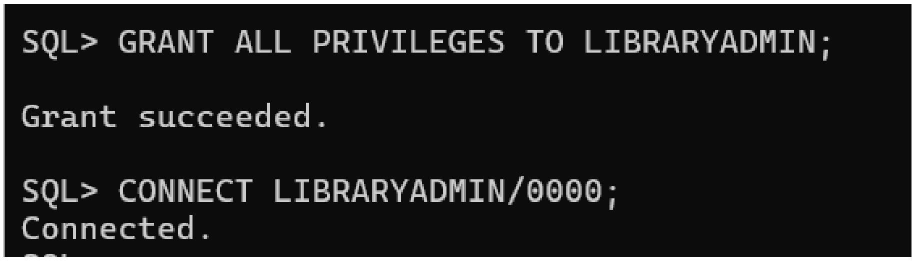
```
SQL> create tablespace LibraryTS1  DATAFILE 'C:\DB\LibraryTS1.DAT' SIZE 100M AUT
OEXTEND ON ONLINE;

Tablespace created.
```



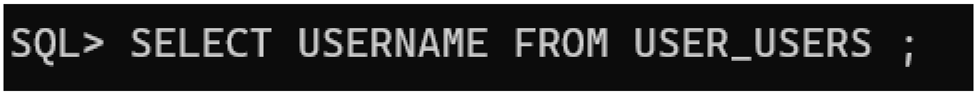
```
SQL> create TEMPORARY tablespace LibraryTS2  TEMPFILE 'C:\DB\LibraryTS2.DAT' SIZ
E 100M AUTOEXTEND ON ;

Tablespace created.
```



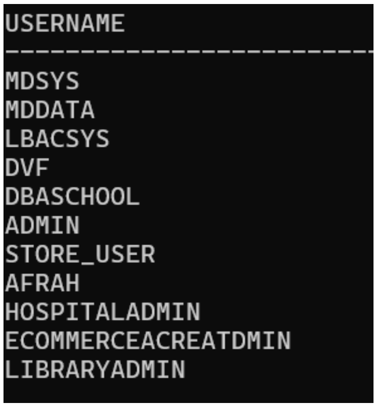
```
SQL> CREATE USER LibraryAdmin IDENTIFIED BY 0000 DEFAULT tablespace LibraryTS1
TEMPORARY tablespace LibraryTS2;

User created.
```



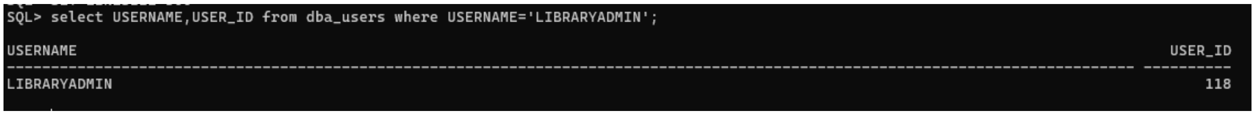
```
SQL> GRANT ALL PRIVILEGES TO LIBRARYADMIN;

Grant succeeded.

SQL> CONNECT LIBRARYADMIN/0000;
Connected.
```



```
SQL> SELECT USERNAME FROM USER_USERS ;
```

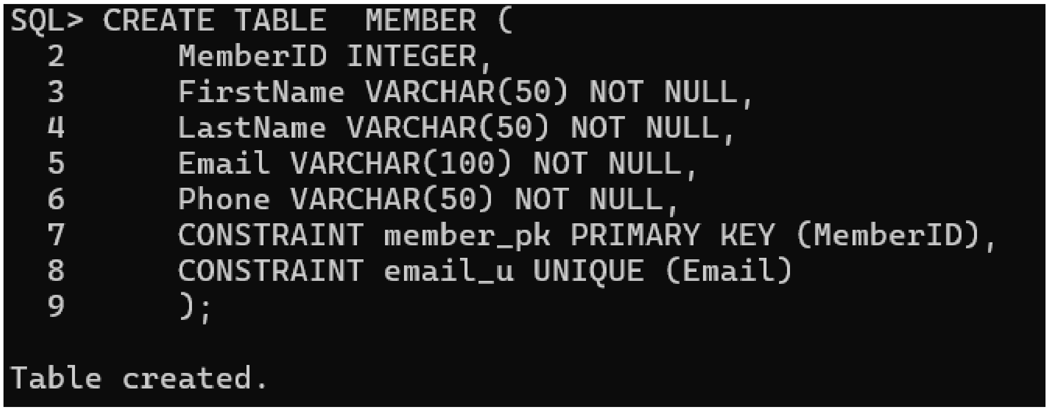


```
USERNAME
-------------------------
MDSYS
MDDATA
LBACSYS
DVF
DBASCHOOL
ADMIN
STORE_USER
AFRAH
HOSPITALADMIN
ECOMMERCEACREATDMIN
LIBRARYADMIN
```

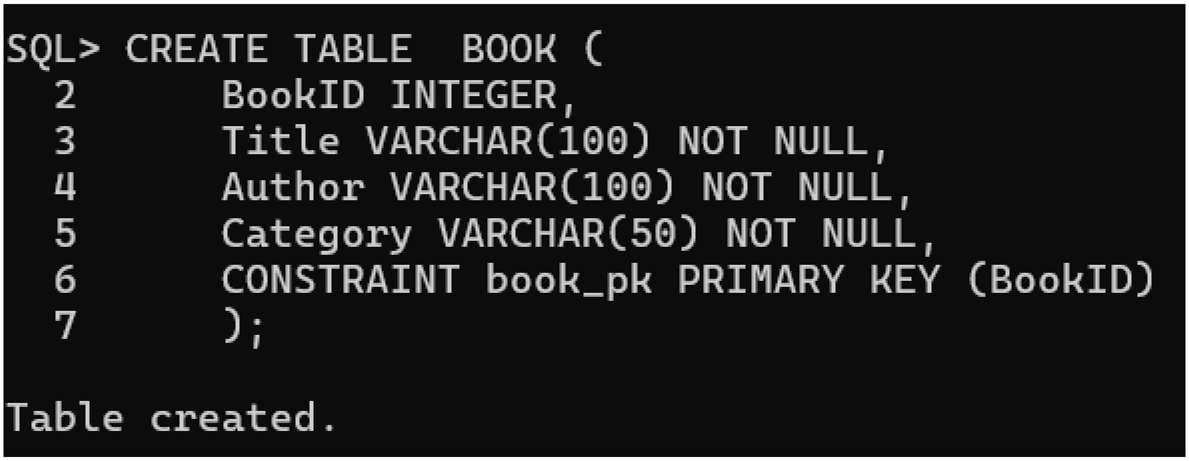


```
SQL> select USERNAME,USER_ID from dba_users where USERNAME='LIBRARYADMIN';

USERNAME                                                                                                    USER_ID
-------------------------------------------------------------------------------------------------------- ----------
LIBRARYADMIN                                                                                                    118
```

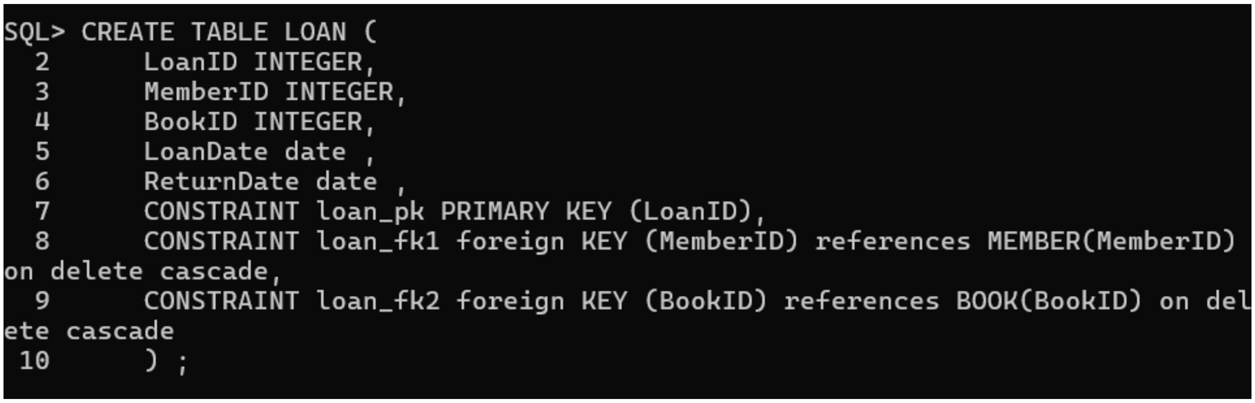
PART 2



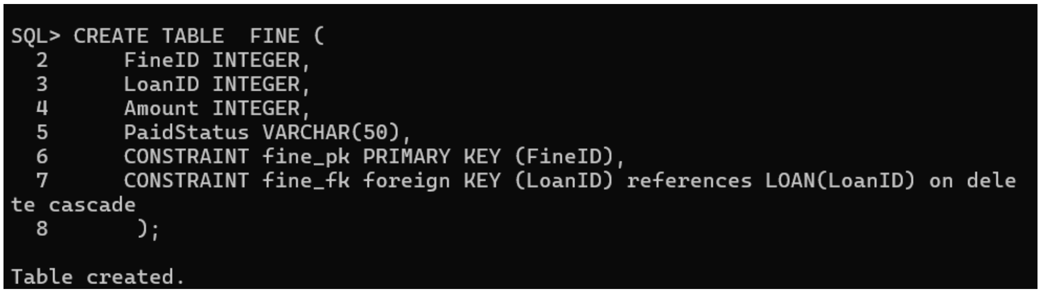
```
SQL> CREATE TABLE  MEMBER (
  2      MemberID INTEGER,
  3      FirstName VARCHAR(50) NOT NULL,
  4      LastName VARCHAR(50) NOT NULL,
  5      Email VARCHAR(100) NOT NULL,
  6      Phone VARCHAR(50) NOT NULL,
  7      CONSTRAINT member_pk PRIMARY KEY (MemberID),
  8      CONSTRAINT email_u UNIQUE (Email)
  9      );

Table created.
```



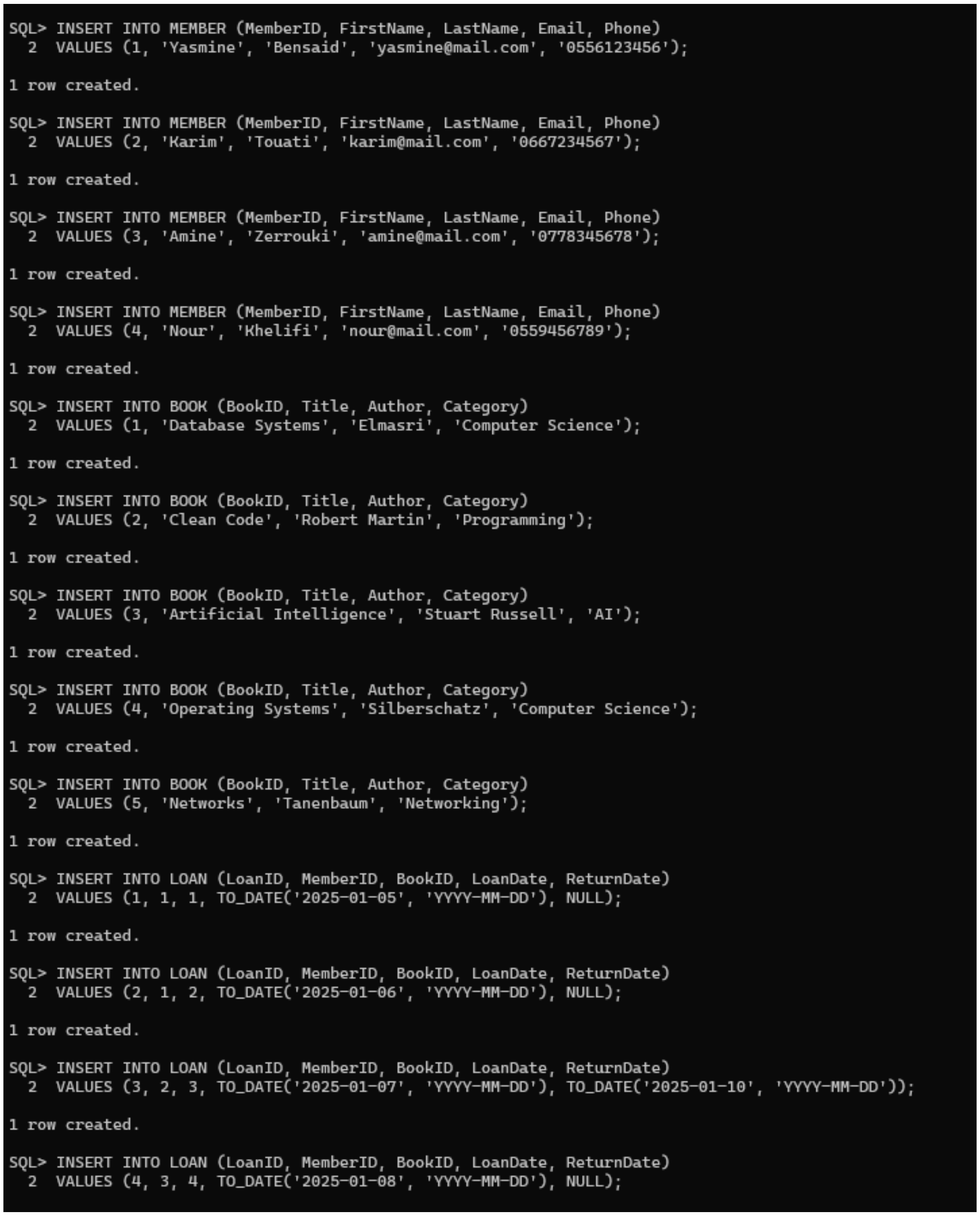
```
SQL> CREATE TABLE  BOOK (
  2      BookID INTEGER,
  3      Title VARCHAR(100) NOT NULL,
  4      Author VARCHAR(100) NOT NULL,
  5      Category VARCHAR(50) NOT NULL,
  6      CONSTRAINT book_pk PRIMARY KEY (BookID)
  7      );

Table created.
```



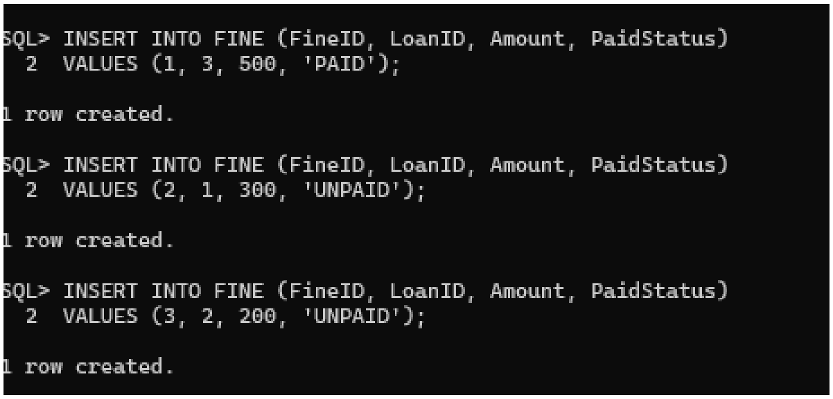
```
SQL> CREATE TABLE LOAN (
  2      LoanID INTEGER,
  3      MemberID INTEGER,
  4      BookID INTEGER,
  5      LoanDate date ,
  6      ReturnDate date ,
  7      CONSTRAINT loan_pk PRIMARY KEY (LoanID),
  8      CONSTRAINT loan_fk1 foreign KEY (MemberID) references MEMBER(MemberID)
on delete cascade,
  9      CONSTRAINT loan_fk2 foreign KEY (BookID) references BOOK(BookID) on del
ete cascade
 10      ) ;
```



```
SQL> CREATE TABLE  FINE (
  2      FineID INTEGER,
  3      LoanID INTEGER,
  4      Amount INTEGER,
  5      PaidStatus VARCHAR(50),
  6      CONSTRAINT fine_pk PRIMARY KEY (FineID),
  7      CONSTRAINT fine_fk foreign KEY (LoanID) references LOAN(LoanID) on dele
te cascade
  8       );

Table created.
```

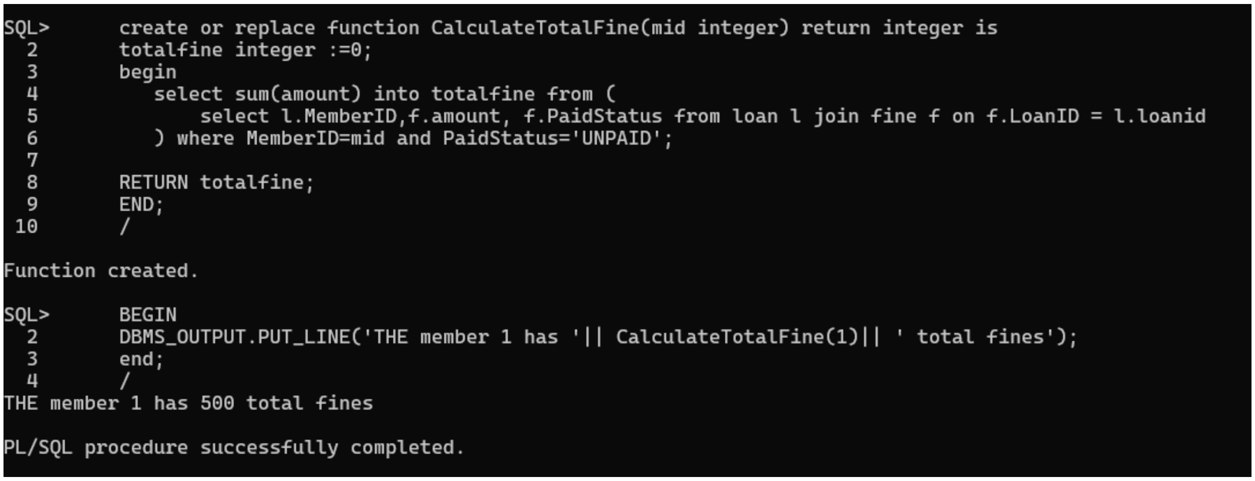


```
SQL> INSERT INTO MEMBER (MemberID, FirstName, LastName, Email, Phone)
  2  VALUES (1, 'Yasmine', 'Bensaid', 'yasmine@mail.com', '0556123456');

1 row created.

SQL> INSERT INTO MEMBER (MemberID, FirstName, LastName, Email, Phone)
  2  VALUES (2, 'Karim', 'Touati', 'karim@mail.com', '0667234567');

1 row created.

SQL> INSERT INTO MEMBER (MemberID, FirstName, LastName, Email, Phone)
  2  VALUES (3, 'Amine', 'Zerrouki', 'amine@mail.com', '0778345678');

1 row created.

SQL> INSERT INTO MEMBER (MemberID, FirstName, LastName, Email, Phone)
  2  VALUES (4, 'Nour', 'Khelifi', 'nour@mail.com', '0559456789');

1 row created.

SQL> INSERT INTO BOOK (BookID, Title, Author, Category)
  2  VALUES (1, 'Database Systems', 'Elmasri', 'Computer Science');

1 row created.

SQL> INSERT INTO BOOK (BookID, Title, Author, Category)
  2  VALUES (2, 'Clean Code', 'Robert Martin', 'Programming');

1 row created.

SQL> INSERT INTO BOOK (BookID, Title, Author, Category)
  2  VALUES (3, 'Artificial Intelligence', 'Stuart Russell', 'AI');

1 row created.

SQL> INSERT INTO BOOK (BookID, Title, Author, Category)
  2  VALUES (4, 'Operating Systems', 'Silberschatz', 'Computer Science');

1 row created.

SQL> INSERT INTO BOOK (BookID, Title, Author, Category)
  2  VALUES (5, 'Networks', 'Tanenbaum', 'Networking');

1 row created.

SQL> INSERT INTO LOAN (LoanID, MemberID, BookID, LoanDate, ReturnDate)
  2  VALUES (1, 1, 1, TO_DATE('2025-01-05', 'YYYY-MM-DD'), NULL);

1 row created.

SQL> INSERT INTO LOAN (LoanID, MemberID, BookID, LoanDate, ReturnDate)
  2  VALUES (2, 1, 2, TO_DATE('2025-01-06', 'YYYY-MM-DD'), NULL);

1 row created.

SQL> INSERT INTO LOAN (LoanID, MemberID, BookID, LoanDate, ReturnDate)
  2  VALUES (3, 2, 3, TO_DATE('2025-01-07', 'YYYY-MM-DD'), TO_DATE('2025-01-10', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO LOAN (LoanID, MemberID, BookID, LoanDate, ReturnDate)
  2  VALUES (4, 3, 4, TO_DATE('2025-01-08', 'YYYY-MM-DD'), NULL);
```

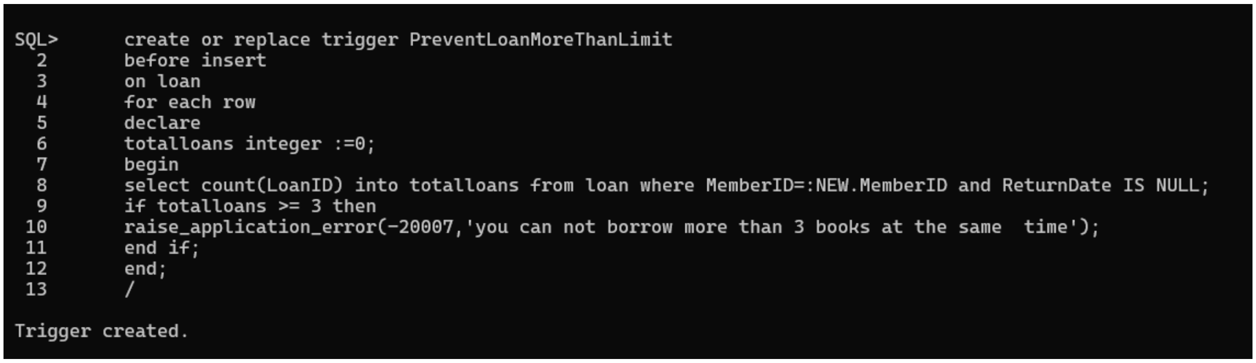


```
SQL> INSERT INTO FINE (FineID, LoanID, Amount, PaidStatus)
     VALUES (1, 3, 500, 'PAID');

1 row created.

SQL> INSERT INTO FINE (FineID, LoanID, Amount, PaidStatus)
     VALUES (2, 1, 300, 'UNPAID');

1 row created.

SQL> INSERT INTO FINE (FineID, LoanID, Amount, PaidStatus)
     VALUES (3, 2, 200, 'UNPAID');

1 row created.
```

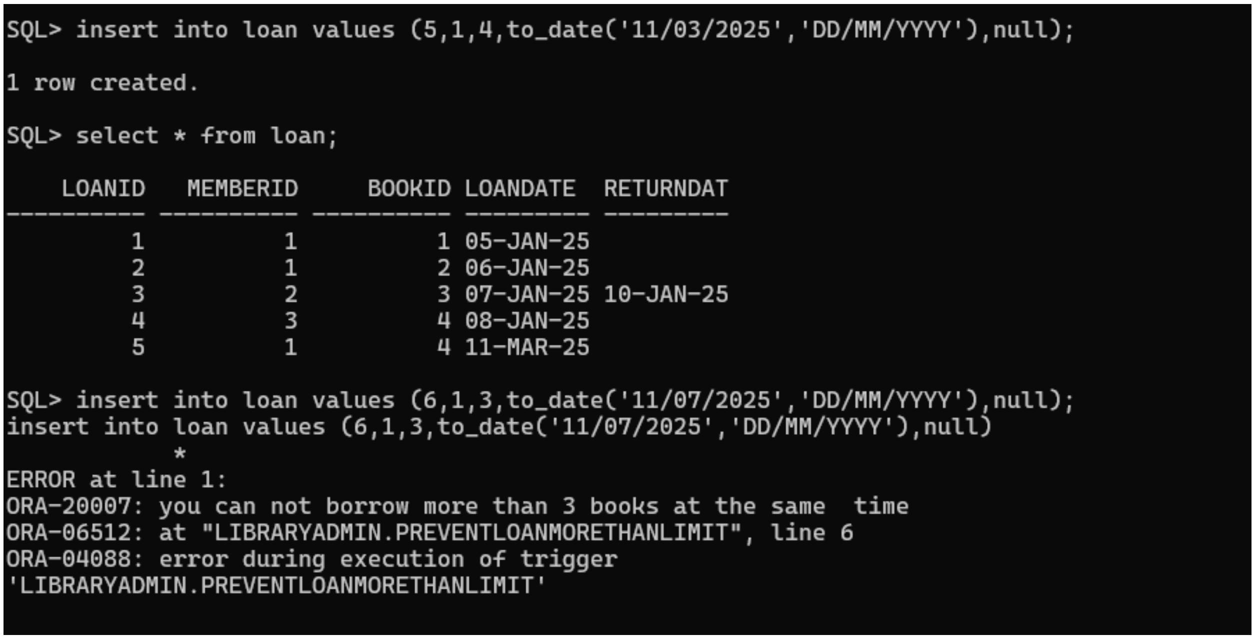
PART 3



```
SQL>      create or replace function CalculateTotalFine(mid integer) return integer is
          totalfine integer :=0;
          begin
             select sum(amount) into totalfine from (
                 select l.MemberID,f.amount, f.PaidStatus from loan l join fine f on f.LoanID = l.loanid
             ) where MemberID=mid and PaidStatus='UNPAID';

          RETURN totalfine;
          END;
          /

Function created.

SQL>      BEGIN
          DBMS_OUTPUT.PUT_LINE('THE member 1 has '|| CalculateTotalFine(1)|| ' total fines');
          end;
          /
THE member 1 has 500 total fines

PL/SQL procedure successfully completed.
```



```
SQL>      create or replace trigger PreventLoanMoreThanLimit
          before insert
          on loan
          for each row
          declare
          totalloans integer :=0;
          begin
          select count(LoanID) into totalloans from loan where MemberID=:NEW.MemberID and ReturnDate IS NULL;
          if totalloans >= 3 then
          raise_application_error(-20007,'you can not borrow more than 3 books at the same  time');
          end if;
          end;
          /

Trigger created.
```



```
SQL> insert into loan values (5,1,4,to_date('11/03/2025','DD/MM/YYYY'),null);

1 row created.

SQL> select * from loan;

    LOANID   MEMBERID     BOOKID LOANDATE  RETURNDAT
---------- ---------- ---------- --------- ---------
         1          1          1 05-JAN-25
         2          1          2 06-JAN-25
         3          2          3 07-JAN-25 10-JAN-25
         4          3          4 08-JAN-25
         5          1          4 11-MAR-25

SQL> insert into loan values (6,1,3,to_date('11/07/2025','DD/MM/YYYY'),null);
insert into loan values (6,1,3,to_date('11/07/2025','DD/MM/YYYY'),null)
            *
ERROR at line 1:
ORA-20007: you can not borrow more than 3 books at the same  time
ORA-06512: at "LIBRARYADMIN.PREVENTLOANMORETHANLIMIT", line 6
ORA-04088: error during execution of trigger
'LIBRARYADMIN.PREVENTLOANMORETHANLIMIT'
```