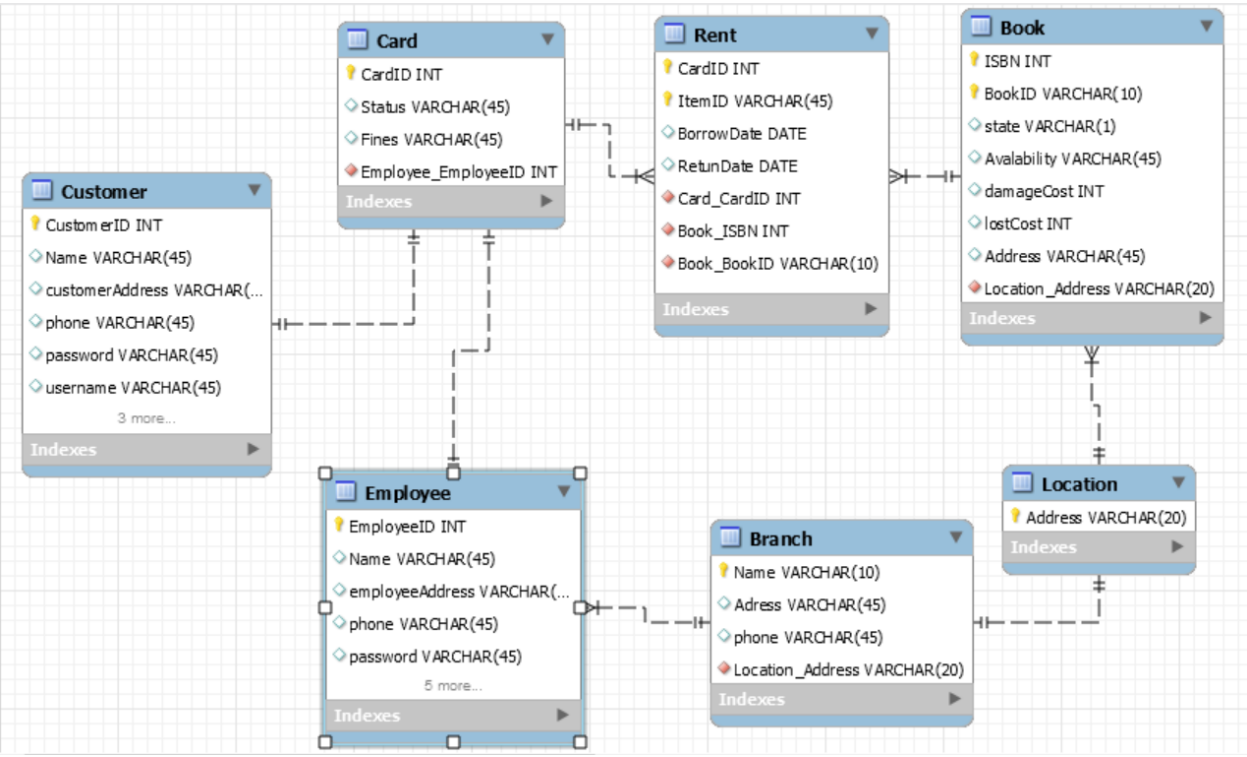
LIBRARY DATABASE MANAGEMENT SYSTEM

Sahana Jangali NUID:001491367

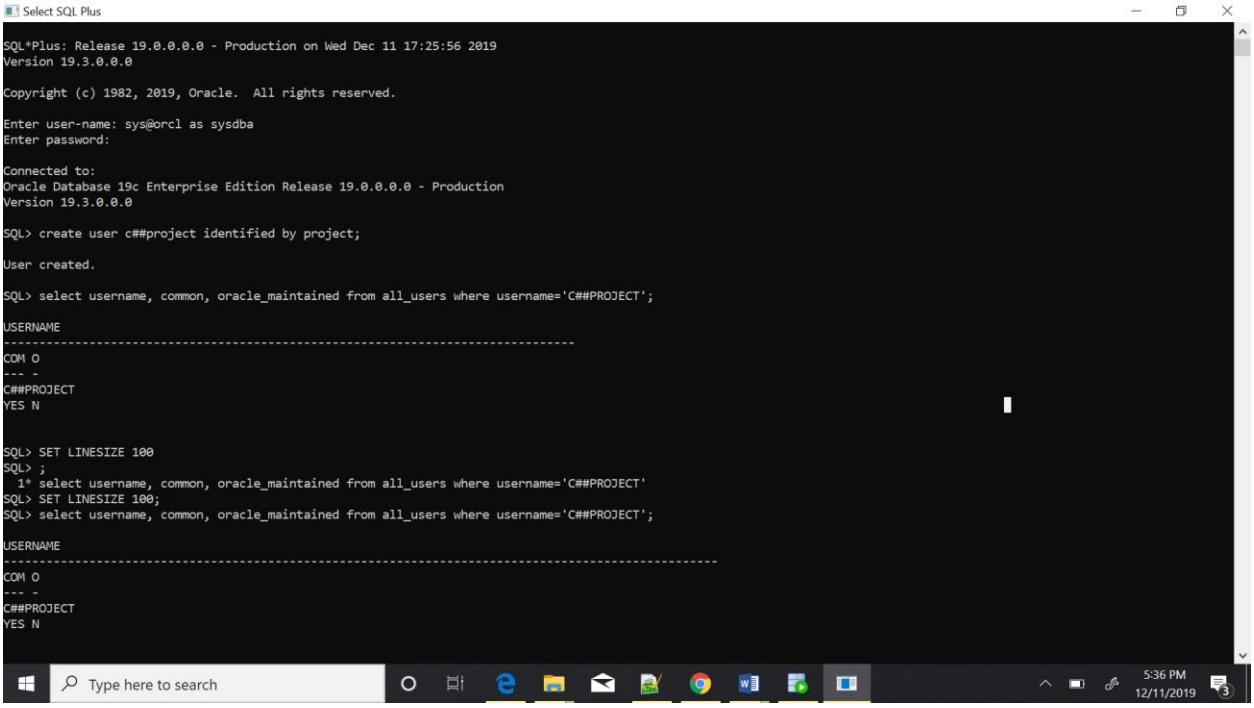
Objective:

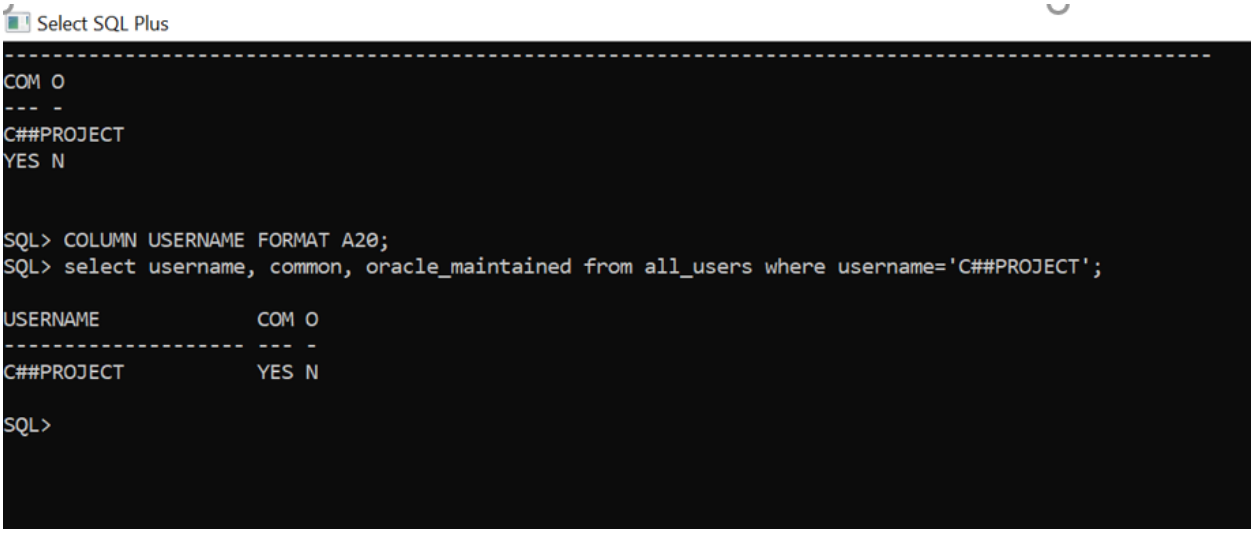
Library is collection of all information and resources that is made accessible to university so that they can access or borrow them for their reference. This database system will keep track of the books, videos that are available in the library. It will also track customers who are borrowing books/online resources and their statuses. The database system will also track employees working at different location and branches.

ER Diagram:

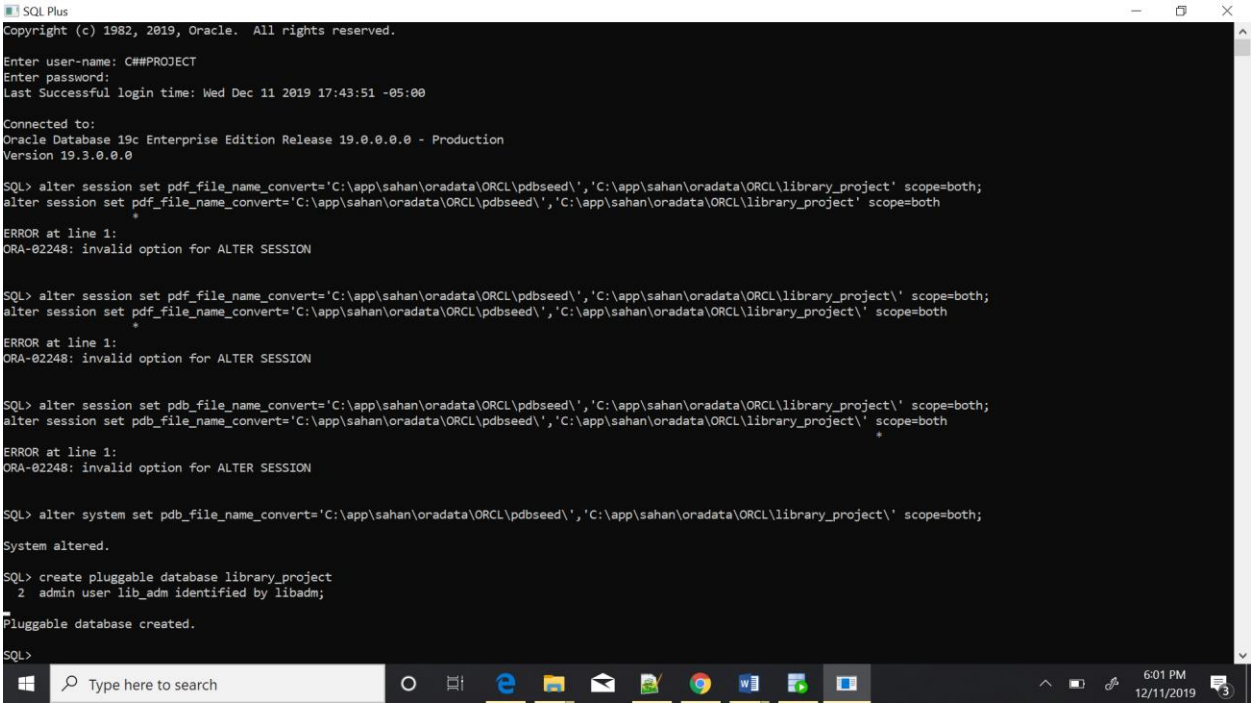


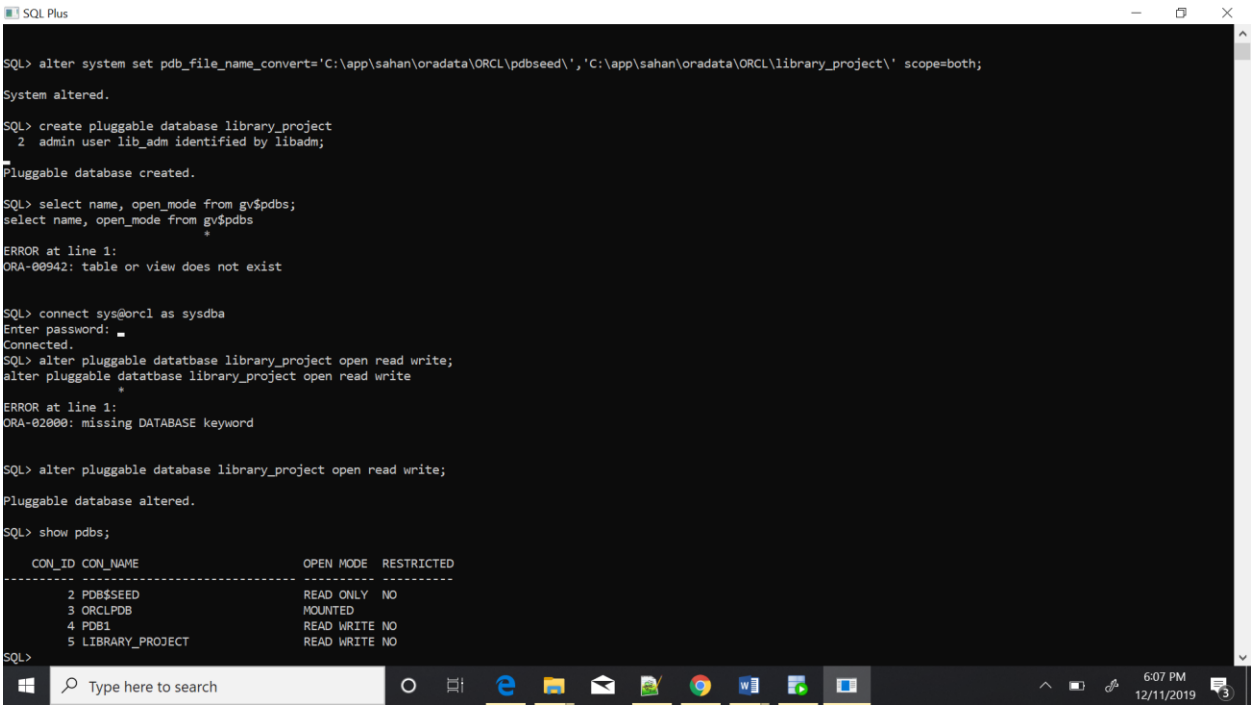
1. Logging into Oracle as sysdba; creation of CDB user;

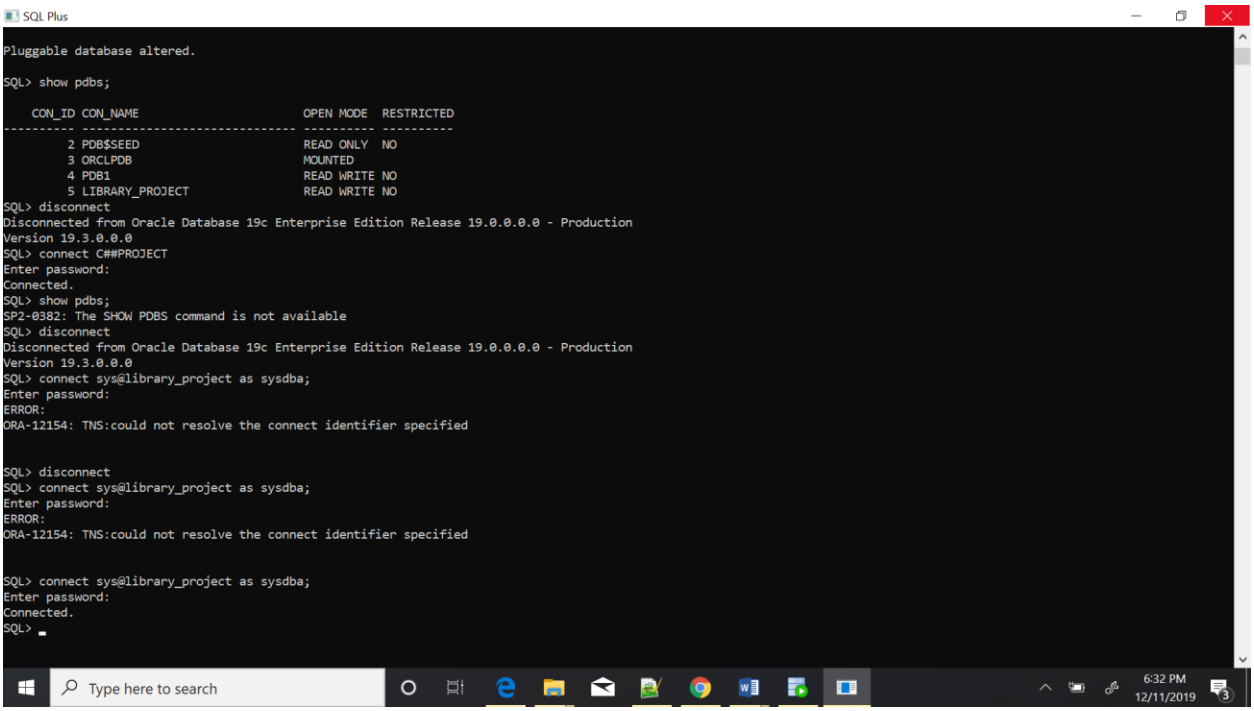




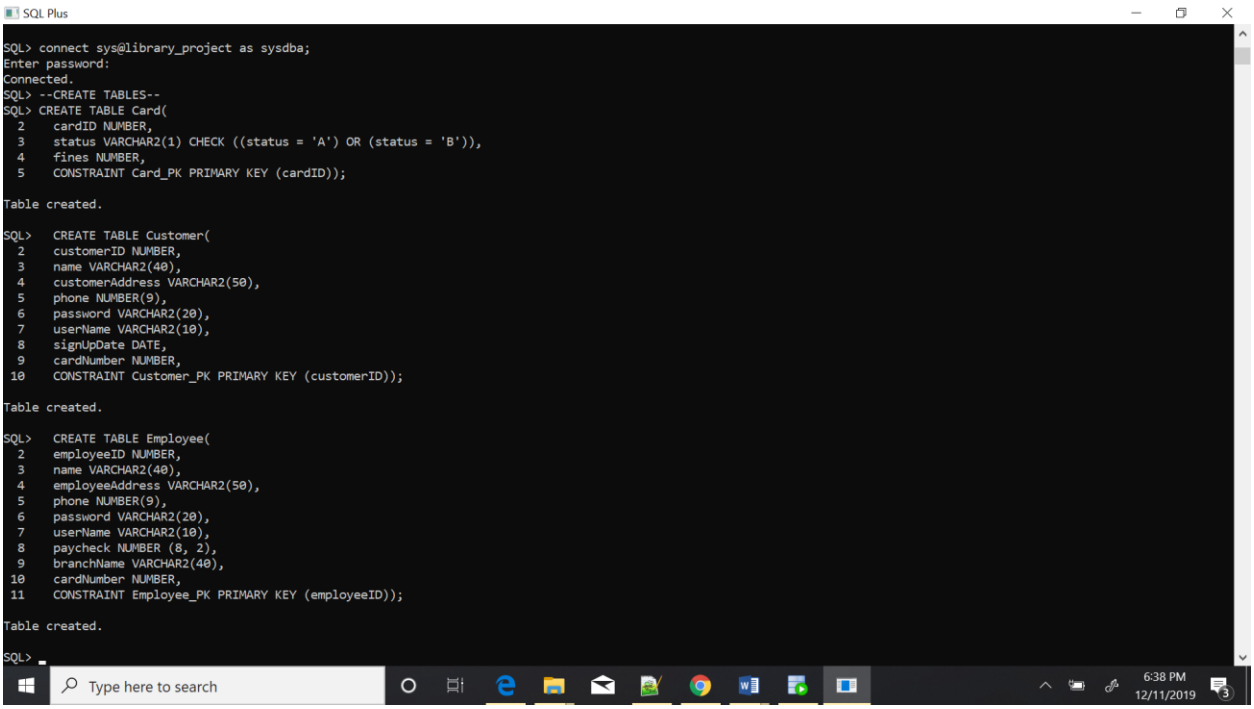
2. Creation of Pluggable database and PDB user:

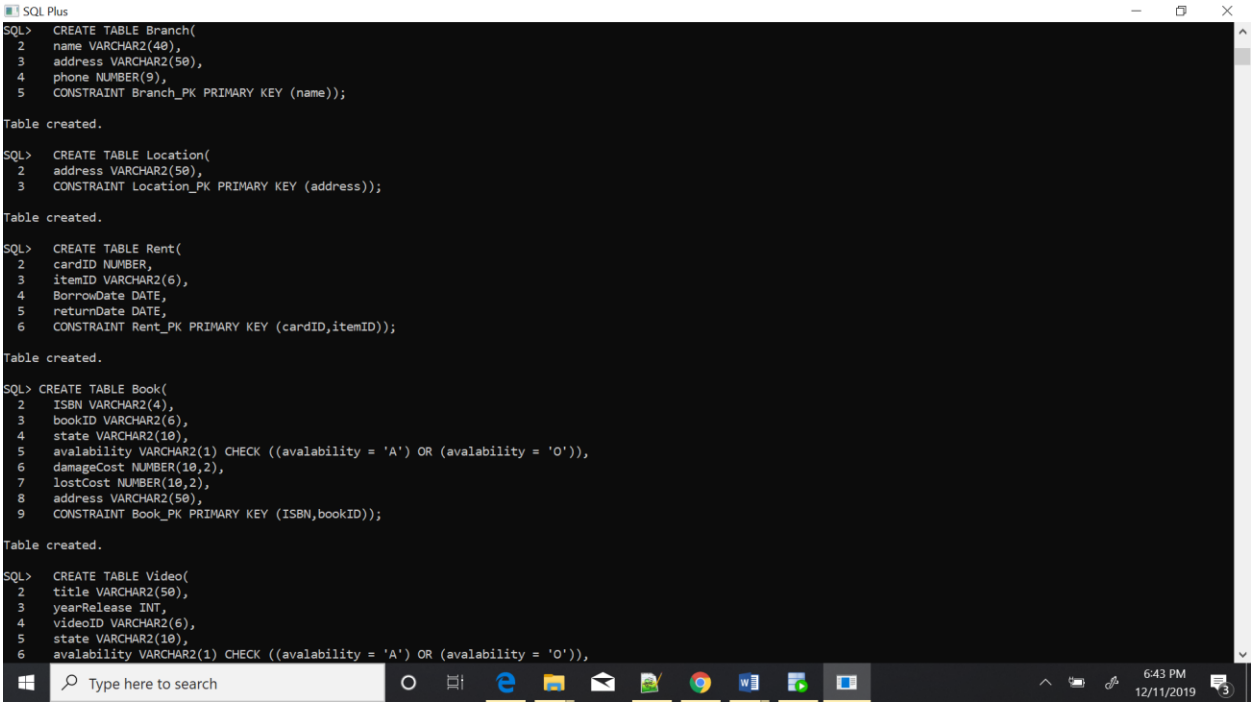


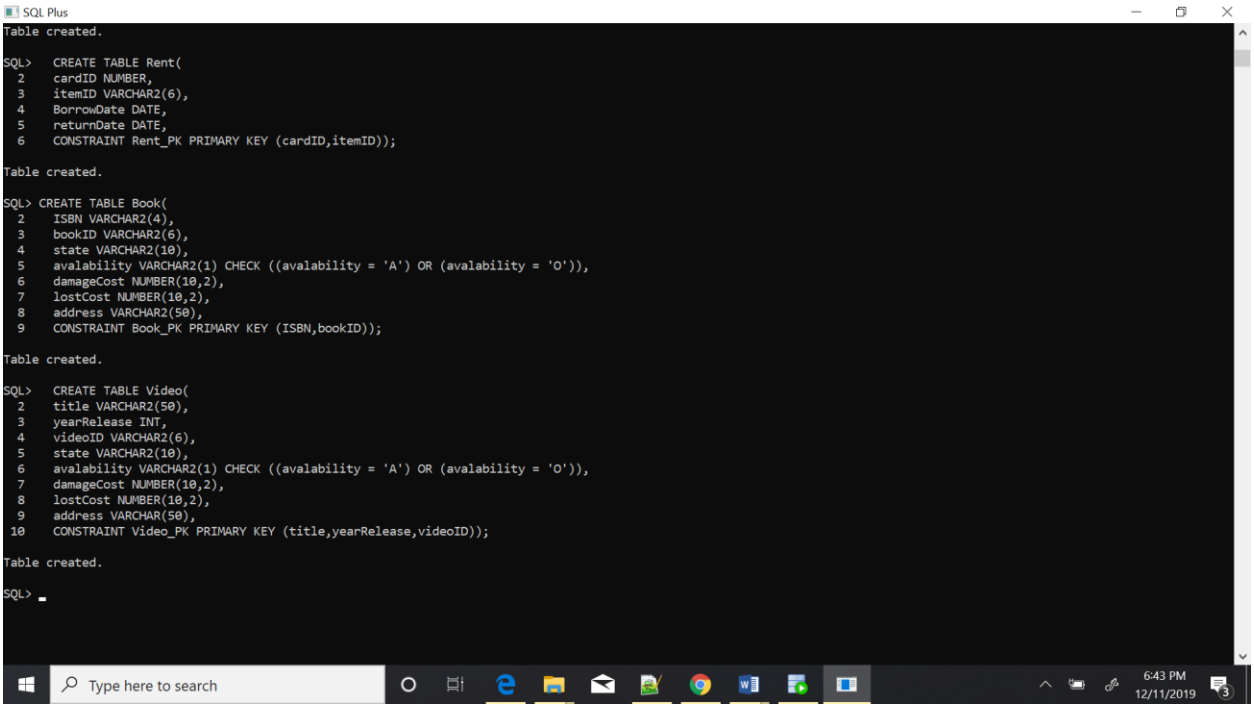




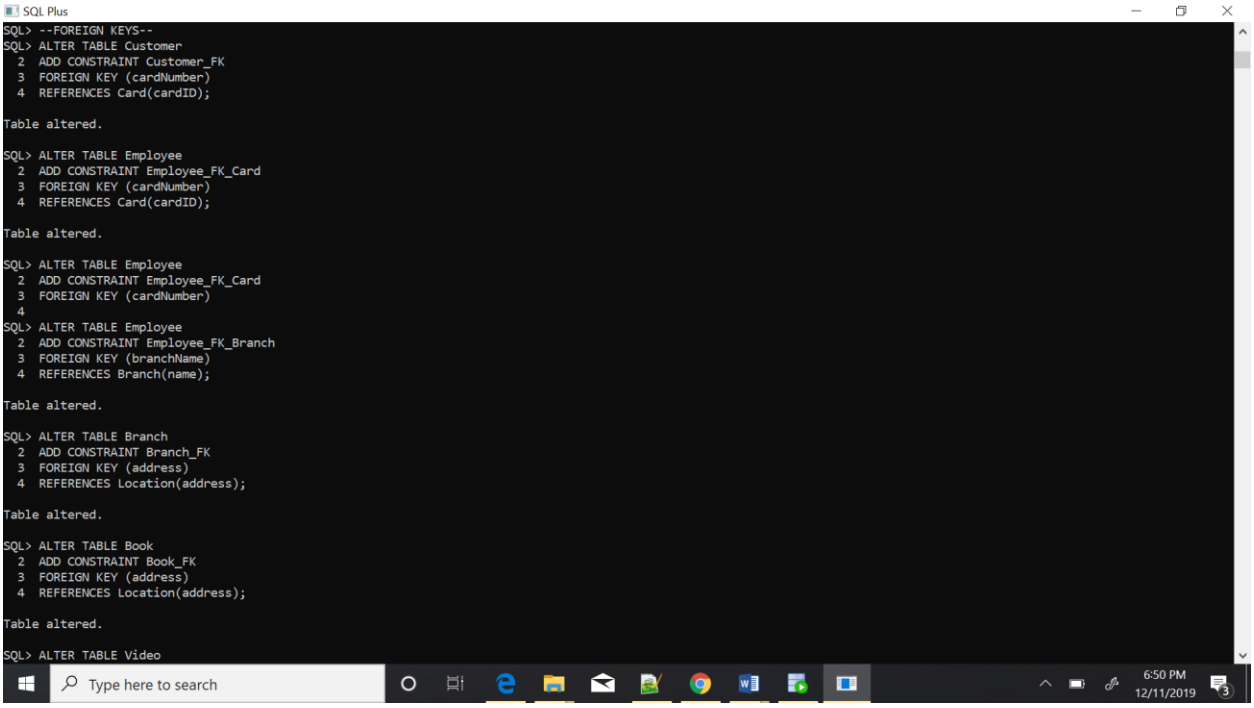
3. Creating tables:

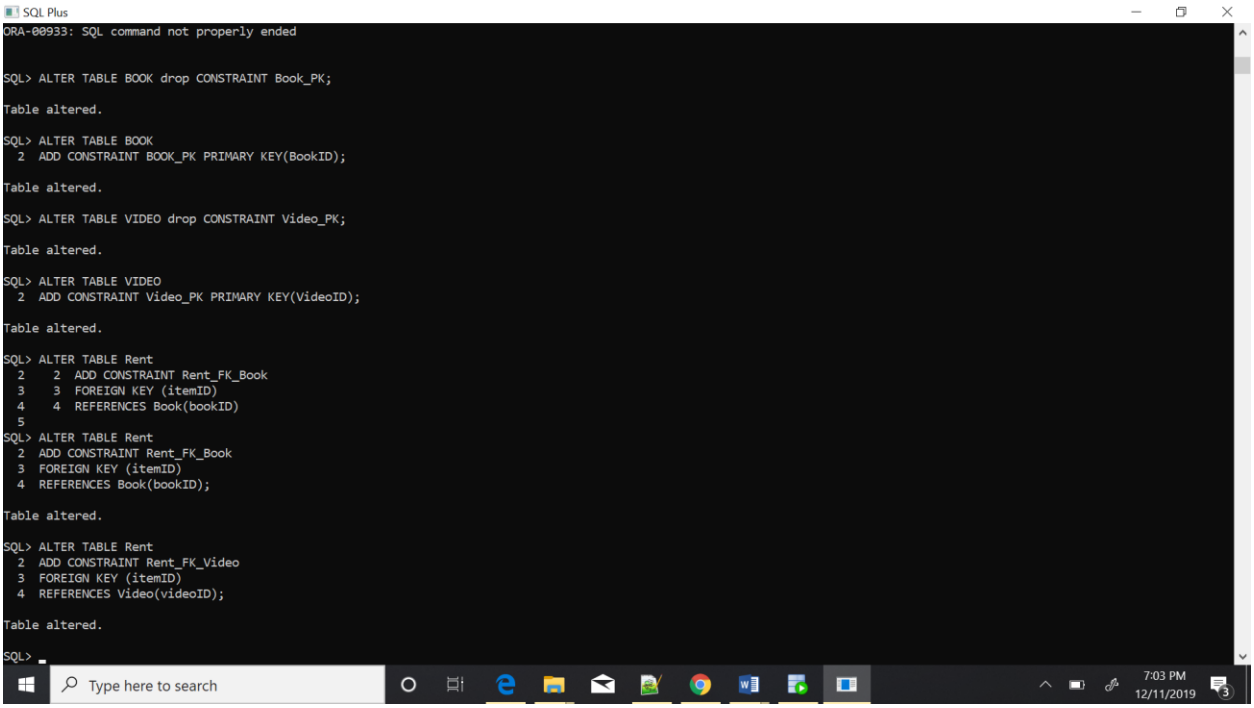



SQL Plus
SQL> CREATE TABLE Branch(
2 name VARCHAR2(40),
3 address VARCHAR2(50),
4 phone NUMBER(9),
5 CONSTRAINT Branch_PK PRIMARY KEY (name));
Table created.
SQL> CREATE TABLE Location(
2 address VARCHAR2(50),
3 CONSTRAINT Location_PK PRIMARY KEY (address));
Table created.
SQL> CREATE TABLE Rent(
2 cardID NUMBER,
3 itemID VARCHAR2(6),
4 BorrowDate DATE,
5 returnDate DATE,
6 CONSTRAINT Rent_PK PRIMARY KEY (cardID,itemID));
Table created.
SQL> CREATE TABLE Book(
2 ISBN VARCHAR2(4),
3 bookID VARCHAR2(6),
4 state VARCHAR2(10),
5 avalability VARCHAR2(1) CHECK ((avalability = 'A') OR (avalability = 'O')),
6 damageCost NUMBER(10,2),
7 lostCost NUMBER(10,2),
8 address VARCHAR2(50),
9 CONSTRAINT Book_PK PRIMARY KEY (ISBN,bookID));
Table created.
SQL> CREATE TABLE Video(
2 title VARCHAR2(50),
3 yearRelease INT,
4 videoID VARCHAR2(6),
5 state VARCHAR2(10),
6 avalability VARCHAR2(1) CHECK ((avalability = 'A') OR (avalability = 'O')),
Type here to search
6:43 PM
12/11/2019


SQL Plus
Table created.
SQL> CREATE TABLE Rent(
2 cardID NUMBER,
3 itemID VARCHAR2(6),
4 BorrowDate DATE,
5 returnDate DATE,
6 CONSTRAINT Rent_PK PRIMARY KEY (cardID,itemID));
Table created.
SQL> CREATE TABLE Book(
2 ISBN VARCHAR2(4),
3 bookID VARCHAR2(6),
4 state VARCHAR2(10),
5 avalability VARCHAR2(1) CHECK ((avalability = 'A') OR (avalability = 'O')),
6 damageCost NUMBER(10,2),
7 lostCost NUMBER(10,2),
8 address VARCHAR2(50),
9 CONSTRAINT Book_PK PRIMARY KEY (ISBN,bookID));
Table created.
SQL> CREATE TABLE Video(
2 title VARCHAR2(50),
3 yearRelease INT,
4 videoID VARCHAR2(6),
5 state VARCHAR2(10),
6 avalability VARCHAR2(1) CHECK ((avalability = 'A') OR (avalability = 'O')),
7 damageCost NUMBER(10,2),
8 lostCost NUMBER(10,2),
9 address VARCHAR(50),
10 CONSTRAINT Video_PK PRIMARY KEY (title,yearRelease,videoID));
Table created.
SQL>
Type here to search
6:43 PM
12/11/2019

4. Adding foreign constraints to implement relationships between tables:





Oracle Concepts

1. External tables:

```
SQL> CREATE DIRECTORY ext_table_dir AS 'C:\Users\sahan\Desktop\branch_data_load.csv';

Directory created.

SQL> grant read on directory ext_table_dir to lib_adm
  2  ;

Grant succeeded.
```

```
SQL>  create table branch_load(
  2   name varchar2(25),
  3   adress varchar2(40),
  4   phoneNumber number
  5   )
  6   organization external (
  7    type oracle_loader
  8    default directory ext_table_dir
  9    access parameters (
 10    nobadfile
 11    fields terminated by ',')
 12    location('branch_data_load.csv')
 13    )
 14    reject limit unlimited
 15    /

Table created.
```

```
SQL> set linesize 100
SQL> desc branch_load;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 NAME                                                           VARCHAR2(25)
 ADRESS                                                         VARCHAR2(40)
 PHONENUMBER                                                    NUMBER
```

2. Relational view:

```
SQL>   create or replace view rent_customer as
  2    select cardID,name
  3    from customer inner join rent
  4    on rent.cardID=customer.cardNumber;

View created.

SQL> select * from rent_customer;

    CARDID NAME
---------- ------------------------------
       101 ALEX WILLIAMS
       105 JENNY KEATING
       102 CHRISTINA LUDDINGTON
```

3. Inline views:

```
SQL>   select branch.name, count(*),
  2       to_char((count(*)/emp.cnt)*100,'99.99')||'%' Employee_Percentage
  3     from branch,
  4     employee,
  5      (select count(*) cnt
  6   from employee) emp
  7   where branch.name=employee.branchname
  8   group by branch.name, emp.cnt
  9   /

NAME                           COUNT(*) EMPLOYEE_PERCENTAGE
------------------------------ ---------- --------------------
COMPUTER SCIENCE                        1  20.00%
MATHEMATICS                             1  20.00%
PHYSICS                                 1  20.00%
CHEMISTRY                               1  20.00%
BIOLOGY                                 1  20.00%
```

4. Materialized view:

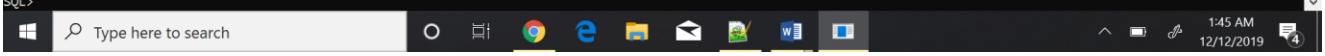
```
SQL>  create materialized view renters_mv
  2   build immediate
  3   refresh on commit
  4   as
  5   select cardID, count(*) Rentals
  6   from rent
  7   group by cardID;

Materialized view created.

SQL> select * from renters_mv;

    CARDID    RENTALS
---------- ----------
       101          1
       102          1
       105          1
       110          1
       112          1
       114          1

6 rows selected.
```

5. Procedure:

```
SQL Plus
SQL> INSERT INTO Rent VALUES (103, 'B1C321', to_date('19-NOV-2019','dd-MON-yyyy'), to_date('19-DEC-2019','dd-MON-yyyy'));

1 row created.

SQL> CREATE OR REPLACE PROCEDURE handle_Returns(l_ItemID IN VARCHAR2)
  2  IS
  3    l_rented NUMBER;
  4    l_book NUMBER;
  5
  6  BEGIN
  7    SELECT COUNT(*) INTO l_rented
  8    FROM rent
  9    WHERE itemid LIKE l_ItemID;
 10
 11    SELECT COUNT(*) INTO l_book
 12    FROM book
 13    WHERE bookid LIKE l_ItemID;
 14
 15    IF l_rented > 0 THEN
 16      DELETE FROM rent
 17      WHERE itemid = l_ItemID;
 18      IF l_Book > 0 THEN
 19        UPDATE book
 20        SET avalability = 'A'
 21        WHERE bookid LIKE l_ItemID;
 22        DBMS_OUTPUT.PUT_LINE('The book ' || l_ItemID || ' is now available.');
 23      END IF;
 24    ELSE
 25      DBMS_OUTPUT.PUT_LINE('This item is not rented at the moment');
 26    END IF;
 27    EXCEPTION WHEN no_data_found THEN
 28    DBMS_OUTPUT.PUT_LINE('Item ID incorrect');
 29  END;
 30  /

Procedure created.

SQL> execute handle_Returns('B1C321');
The book B1C321 is now available.

PL/SQL procedure successfully completed.
```

6. Trigger:

```
SQL Plus
SQL> CREATE OR REPLACE TRIGGER modify_Fines
  2  AFTER DELETE
  3  ON rent
  4  FOR EACH ROW
  5  DECLARE
  6    l_CardID NUMBER;
  7    l_ItemID VARCHAR2(6);
  8    l_Book NUMBER;
  9    l_damage NUMBER;
 10  BEGIN
 11    SELECT cardid, itemid INTO l_CardID, l_ItemID
 12    FROM rent
 13    WHERE cardid LIKE :old.cardid;
 14
 15    SELECT COUNT(*) INTO l_Book
 16    FROM book
 17    WHERE bookid LIKE l_ItemID;
 18
 19      IF sysdate > :old.returndate THEN
 20      ELSIF l_Book > 0 THEN
 21        SELECT damageCost INTO l_damage
 22        FROM book
 23        WHERE bookid LIKE l_ItemID;
 24      END IF;
 25
 26      UPDATE card
 27      SET status = 'B', fines = (fines + l_damage)
 28      WHERE cardid LIKE l_CardID;
 29    ELSE
 30      DBMS_OUTPUT.PUT_LINE('The item has been return before deadline');
 31    END IF;
 32  END;
 33  /
CREATE OR REPLACE TRIGGER modify_Fines
                          *
ERROR at line 1:
ORA-04089: cannot create triggers on objects owned by SYS


SQL>
```



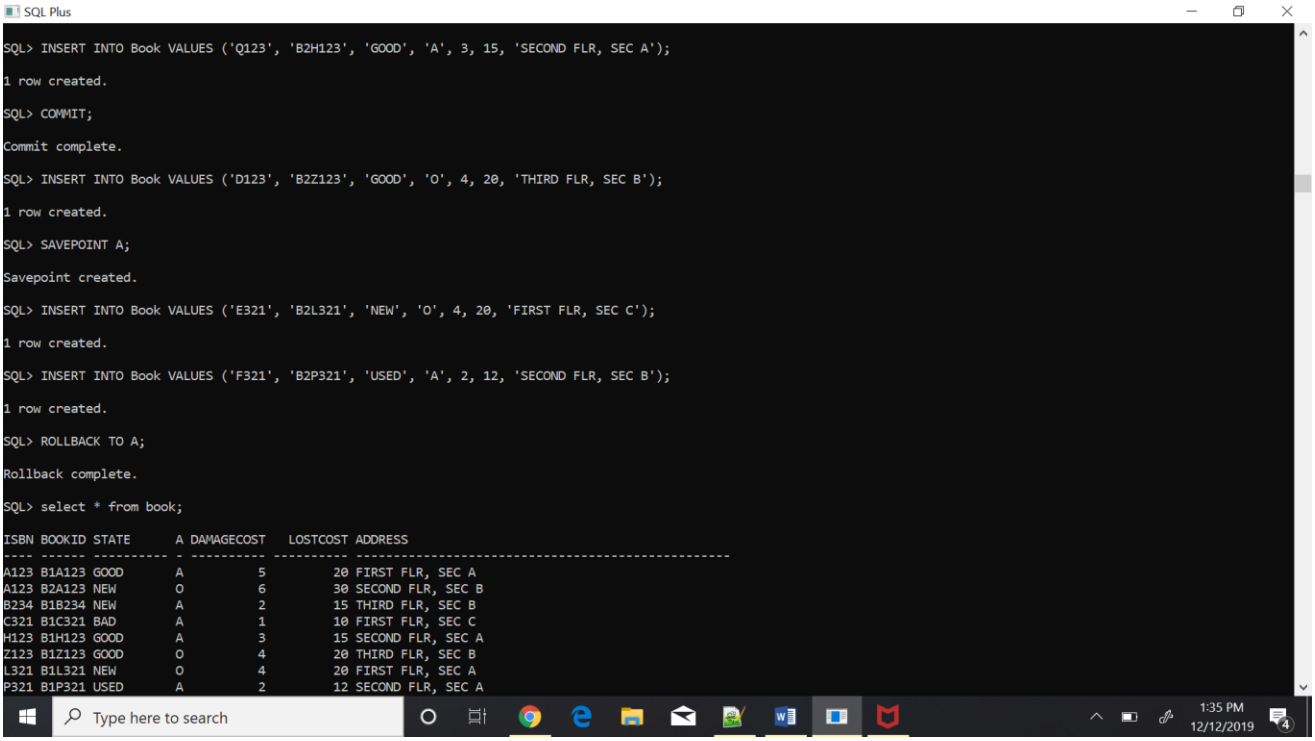
7. Index:

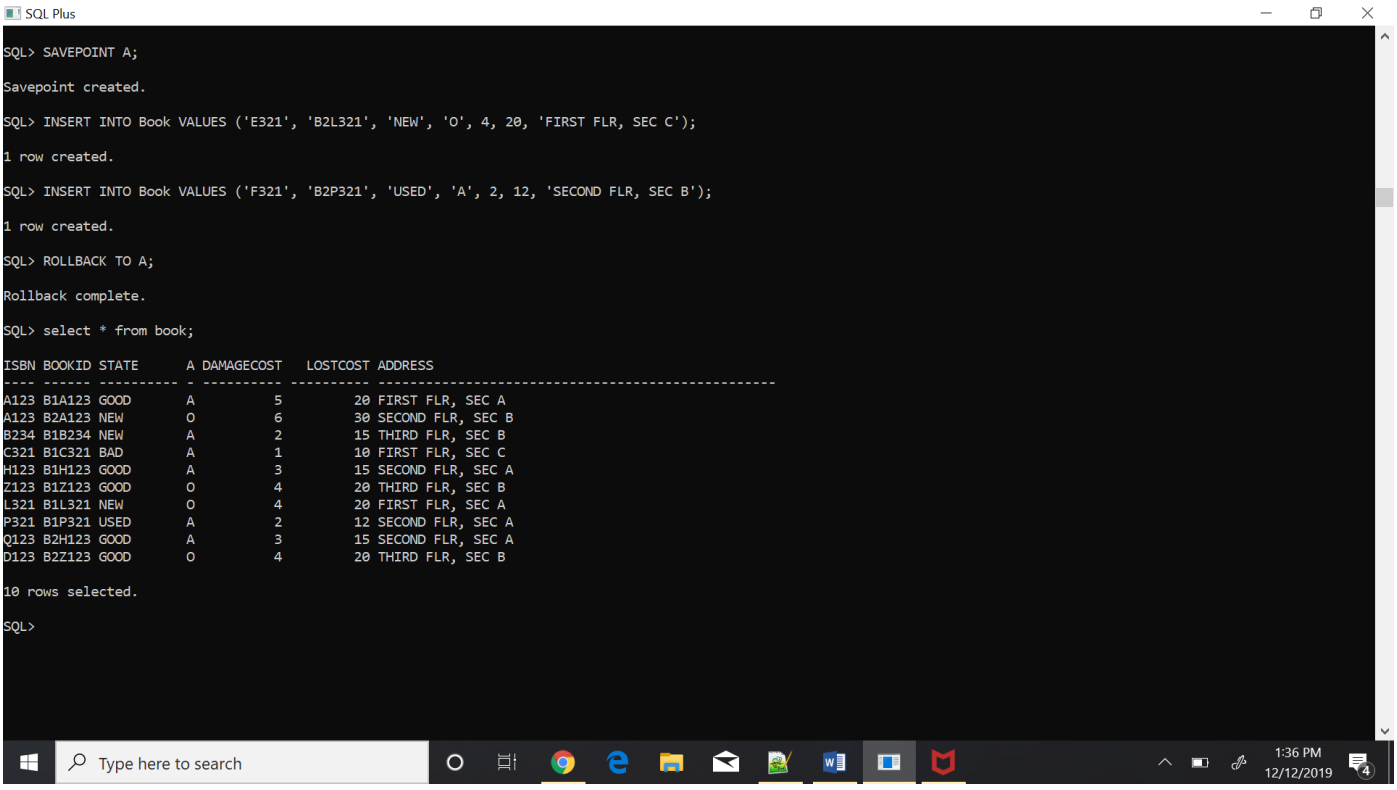
```
SQL>  create index cust_idx
  2   on customer(name);

Index created.
```

8. Explicit cursors:

```
SQL> set serveroutput on
SQL>  declare
  2    cursor card_cur(p_status in varchar2)
  3    is select *
  4    from card
  5    where card.status=p_status;
  6
  7    l_card card%rowtype;
  8    begin
  9     dbms_output.put_line('Getting Blocked card owners');
 10     open card_cur('B');
 11     loop
 12      fetch card_cur into l_card;
 13  exit when card_cur%notfound;
 14  dbms_output.put('Card ID ' || l_card.cardID || ' is ');
 15          dbms_output.put_line(l_card.status);
 16  end loop;
 17  close card_cur;
 18         end;
 19        /
Getting Blocked card owners
Card ID 106 is B
Card ID 107 is B
Card ID 108 is B
Card ID 109 is B

PL/SQL procedure successfully completed.
```

9. Transactions:



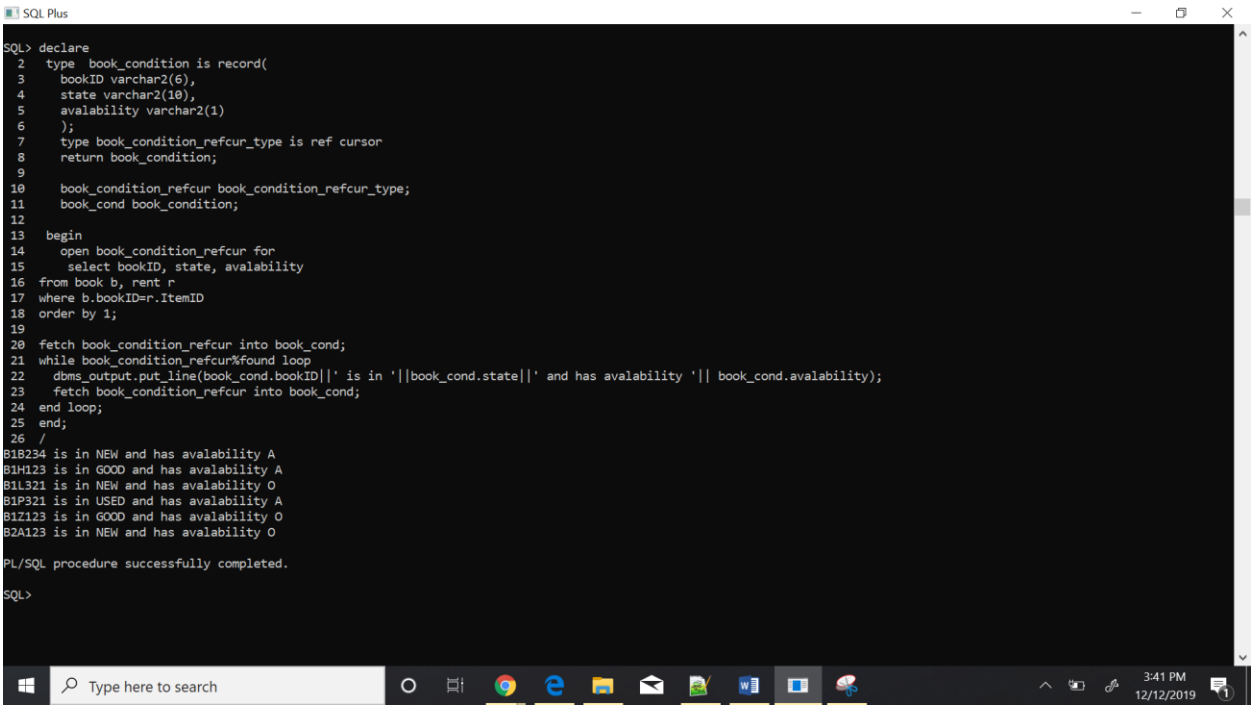


10. Functions:

SQL Plus

```
SQL> CREATE OR REPLACE FUNCTION Emp_details (empid in number)
  2  RETURN VARCHAR2
  3  IS emp VARCHAR2(200);
  4  BEGIN
  5  SELECT '1)Name-' ||Employee.name|| ' 2)Address -' || Employee.employeeAddress || ' 3)Work Location -' ||Employee.branchName into emp
  6  from Employee, Branch where
  7  Employee.branchname=Branch.name
  8  and Employee.EmployeeID=empid;
  9  RETURN(emp);
 10  END Emp_details;
 11  /

Function created.

SQL> select emp_details(211) as "Employee Address" FROM DUAL;

Employee Address
-----------------------------------------------------------------------------------------------
1)Name-SAM 2)Address -150 PARKMAN ST HOUSE 3)Work Location -COMPUTER SCIENCE

SQL>
```

11. Ref cursors: Strongly typed



12. Pre defined exceptions

```
SQL> declare
  2  l_bookid varchar2(10);
  3  l_dc number;
  4  begin
  5  l_bookid := 'B1A123';
  6  l_dc := 'A';
  7
  8  exception
  9  when VALUE_ERROR then
 10  dbms_output.put_line('We encountered the VALUE_ERROR exception');
 11  end;
 12  /
We encountered the VALUE_ERROR exception

PL/SQL procedure successfully completed.

SQL>
SQL>
```

APPENDIX

```
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 11 17:58:07 2019
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Enter user-name: C##PROJECT
Enter password:
Last Successful login time: Wed Dec 11 2019 17:43:51 -05:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> alter system set
pdb_file_name_convert='C:\app\sahan\oradata\ORCL\pdbseed\','C:\app\sahan\oradata\ORCL\lib
rary_project\' scope=both;

System altered.

SQL> create pluggable database library_project
  2  admin user lib_adm identified by libadm;

Pluggable database created.

SQL> connect sys@orcl as sysdba
Enter password:
Connected.

SQL> alter pluggable database library_project open read write;

Pluggable database altered.

SQL> show pdbs;

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB                        MOUNTED
         4 PDB1                         READ WRITE NO
         5 LIBRARY_PROJECT                READ WRITE NO
SQL> disconnect
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

```
SQL> connect C##PROJECT
Enter password:
Connected.
SQL> show pdbs;
SP2-0382: The SHOW PDBS command is not available
SQL> disconnect
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> connect sys@library_project as sysdba;
Enter password:
Connected.
SQL> --CREATE TABLES--
SQL> CREATE TABLE Card(
  2    cardID NUMBER,
  3    status VARCHAR2(1) CHECK ((status = 'A') OR (status = 'B')),
  4    fines NUMBER,
  5    CONSTRAINT Card_PK PRIMARY KEY (cardID));

Table created.

SQL>   CREATE TABLE Customer(
  2    customerID NUMBER,
  3    name VARCHAR2(40),
  4    customerAddress VARCHAR2(50),
  5    phone NUMBER(9),
  6    password VARCHAR2(20),
  7    userName VARCHAR2(10),
  8    signUpDate DATE,
  9    cardNumber NUMBER,
 10    CONSTRAINT Customer_PK PRIMARY KEY (customerID));

Table created.

SQL>   CREATE TABLE Employee(
  2    employeeID NUMBER,
  3    name VARCHAR2(40),
  4    employeeAddress VARCHAR2(50),
  5    phone NUMBER(9),
  6    password VARCHAR2(20),
  7    userName VARCHAR2(10),
  8    paycheck NUMBER (8, 2),
  9    branchName VARCHAR2(40),
 10    cardNumber NUMBER,
 11    CONSTRAINT Employee_PK PRIMARY KEY (employeeID));
```

```
Table created.

SQL>   CREATE TABLE Branch(
  2    name VARCHAR2(40),
  3    address VARCHAR2(50),
  4    phone NUMBER(9),
  5    CONSTRAINT Branch_PK PRIMARY KEY (name));

Table created.

SQL>   CREATE TABLE Location(
  2    address VARCHAR2(50),
  3    CONSTRAINT Location_PK PRIMARY KEY (address));

Table created.

SQL>   CREATE TABLE Rent(
  2    cardID NUMBER,
  3    itemID VARCHAR2(6),
  4    BorrowDate DATE,
  5    returnDate DATE,
  6    CONSTRAINT Rent_PK PRIMARY KEY (cardID,itemID));

Table created.

SQL> CREATE TABLE Book(
  2    ISBN VARCHAR2(4),
  3    bookID VARCHAR2(6),
  4    state VARCHAR2(10),
  5    avalability VARCHAR2(1) CHECK ((avalability = 'A') OR (avalability = 'O')),
  6    damageCost NUMBER(10,2),
  7    lostCost NUMBER(10,2),
  8    address VARCHAR2(50),
  9    CONSTRAINT Book_PK PRIMARY KEY (ISBN,bookID));

Table created.

SQL> --FOREIGN KEYS--
SQL> ALTER TABLE Customer
  2  ADD CONSTRAINT Customer_FK
  3  FOREIGN KEY (cardNumber)
  4  REFERENCES Card(cardID);

Table altered.

SQL> ALTER TABLE Employee
```

```
  2  ADD CONSTRAINT Employee_FK_Card
  3  FOREIGN KEY (cardNumber)
  4  REFERENCES Card(cardID);

Table altered.

SQL> ALTER TABLE Employee
  2  ADD CONSTRAINT Employee_FK_Card
  3  FOREIGN KEY (cardNumber)
  4
SQL> ALTER TABLE Employee
  2  ADD CONSTRAINT Employee_FK_Branch
  3  FOREIGN KEY (branchName)
  4  REFERENCES Branch(name);

Table altered.

SQL> ALTER TABLE Branch
  2  ADD CONSTRAINT Branch_FK
  3  FOREIGN KEY (address)
  4  REFERENCES Location(address);

Table altered.

SQL> ALTER TABLE Book
  2  ADD CONSTRAINT Book_FK
  3  FOREIGN KEY (address)
  4  REFERENCES Location(address);

Table altered.


SQL> ALTER TABLE Rent
  2  ADD CONSTRAINT Rent_FK_Card
  3  FOREIGN KEY (cardID)
  4  REFERENCES Card(cardID);

Table altered.

SQL> ALTER TABLE Rent
  2  ADD CONSTRAINT Rent_FK_Book
  3  FOREIGN KEY (itemID)
  4  REFERENCES Book(bookID);
REFERENCES Book(bookID)
           *
ERROR at line 4:
```

```
ORA-02270: no matching unique or primary key for this column-list


SQL> ALTER TABLE BOOK MODIFY CONSTRAINT Book_PK PRIMARY KEY(BookID);
ALTER TABLE BOOK MODIFY CONSTRAINT Book_PK PRIMARY KEY(BookID)
                                  *
ERROR at line 1:
ORA-00933: SQL command not properly ended


SQL> ALTER TABLE BOOK drop CONSTRAINT Book_PK;

Table altered.

SQL> ALTER TABLE BOOK
  2  ADD CONSTRAINT BOOK_PK PRIMARY KEY(BookID);

Table altered.

SQL> ALTER TABLE VIDEO drop CONSTRAINT Video_PK;

Table altered.

SQL> ALTER TABLE VIDEO
  2  ADD CONSTRAINT Video_PK PRIMARY KEY(VideoID);

Table altered.

SQL> ALTER TABLE Rent
  2    2  ADD CONSTRAINT Rent_FK_Book
  3    3  FOREIGN KEY (itemID)
  4    4  REFERENCES Book(bookID)
  5
SQL> ALTER TABLE Rent
  2  ADD CONSTRAINT Rent_FK_Book
  3  FOREIGN KEY (itemID)
  4  REFERENCES Book(bookID);

Table altered.

SQL> INSERT INTO Card VALUES (100,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (101,'A',0);
```

1 row created.

SQL> INSERT INTO Card VALUES (102,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (103,'A',13);

1 row created.

SQL> INSERT INTO Card VALUES (104,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (105,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (106,'B',0);

1 row created.

SQL> INSERT INTO Card VALUES (107,'B',10);

1 row created.

SQL> INSERT INTO Card VALUES (108,'B',20);

1 row created.

SQL> INSERT INTO Card VALUES (109,'B',0);

1 row created.

SQL> INSERT INTO Card VALUES (110,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (111,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (112,'A',6);

1 row created.

```
SQL> INSERT INTO Card VALUES (113,'A',0);

1 row created.

SQL> INSERT INTO Card VALUES (114,'A',0);

1 row created.


SQL> INSERT INTO Location VALUES ('FIRST FLR, SEC A');

1 row created.

SQL> INSERT INTO Location VALUES ('SECOND FLR, SEC B');

1 row created.

SQL> INSERT INTO Location VALUES ('THIRD FLR, SEC B');

1 row created.

SQL> INSERT INTO Location VALUES ('FIRST FLR, SEC C');

1 row created.

SQL> INSERT INTO Location VALUES ('SECOND FLR, SEC A');

1 row created.

SQL> INSERT INTO Branch VALUES ('COMPUTER SCIENCE', 'FIRST FLR, SEC A',
857303838);

1 row created.

SQL> INSERT INTO Branch VALUES ('CHEMISTRY', 'SECOND FLR, SEC B', 622863281);

1 row created.

SQL> INSERT INTO Branch VALUES ('BIOLOGY', 'THIRD FLR, SEC B', 642908944);

1 row created.

SQL> INSERT INTO Branch VALUES ('PHYSICS', 'FIRST FLR, SEC C', 768903666);

1 row created.
```

```
SQL> INSERT INTO Branch VALUES ('MATHEMATICS', 'SECOND FLR, SEC A',
657890561);

1 row created.

SQL> INSERT INTO Book VALUES ('A123', 'B1A123', 'GOOD', 'A', 5, 20, 'FIRST FLR, SEC
A');

1 row created.

SQL> INSERT INTO Book VALUES ('A123', 'B2A123', 'NEW', 'O', 6, 30, 'SECOND FLR,
SEC B');

1 row created.

SQL> INSERT INTO Book VALUES ('B234', 'B1B234', 'NEW', 'A', 2, 15, 'THIRD FLR, SEC
B');

1 row created.

SQL> INSERT INTO Book VALUES ('C321', 'B1C321', 'BAD', 'A', 1, 10, 'FIRST FLR, SEC
C');

1 row created.

SQL> INSERT INTO Book VALUES ('H123', 'B1H123', 'GOOD', 'A', 3, 15, 'SECOND FLR,
SEC A');

1 row created.

SQL> INSERT INTO Book VALUES ('Z123', 'B1Z123', 'GOOD', 'O', 4, 20, 'THIRD FLR, SEC
B');

1 row created.

SQL> INSERT INTO Book VALUES ('L321', 'B1L321', 'NEW', 'O', 4, 20, 'FIRST FLR, SEC
A');

1 row created.

SQL> INSERT INTO Book VALUES ('P321', 'B1P321', 'USED', 'A', 2, 12, 'SECOND FLR,
SEC A');

1 row created.
```

```
SQL> INSERT INTO Employee VALUES (211, 'SAM', '150 PARKMAN ST HOUSE',
671671671, 'SAM123', 'SAM1', 5000, 'COMPUTER SCIENCE', 110);

1 row created.

SQL> INSERT INTO Employee VALUES (212, 'WES', '13 HEARTH ST', 688688688,
'WES123', 'WES12', 3500.50, 'CHEMISTRY', 111);

1 row created.

SQL> INSERT INTO Employee VALUES (213, 'ASHER', '76 PERTH AVE', 628628628,
'ASHER123', 'ASHER13', 4570.75, 'BIOLOGY', 112);

1 row created.

SQL> INSERT INTO Employee VALUES (214, 'VICTOR', '89 INDIA ST', 654321987,
'VICTOR123', 'VICTOR14', 5575, 'PHYSICS', 113);

1 row created.

SQL> INSERT INTO Employee VALUES (215, 'SABRINA', '100 ITALY PKWY', 698754321,
'SABRINA123', 'SABRINA5', 5050.50, 'MATHEMATICS', 114);

1 row created.


SQL> INSERT INTO Customer VALUES (1, 'NIDHI REDDY', '170 PARKER HILL',
651098656, 'NIDHI123', 'RNIDHI1', to_date('14-APR-16', 'dd-MON-yyyy'), 100);

1 row created.


SQL> INSERT INTO Customer VALUES (2, 'ALEX WILLIAMS', '45 PARK DRIVE
ABBEY', 615516890, 'ALEX123', 'WALEX22',to_date('10-JUN-2018', 'dd-MON-yyyy'), 101);

1 row created.

SQL> INSERT INTO Customer VALUES (4, 'TOM PETERS', '45 TREMONT ST', 658530958,
'tom123', 'PTOM4', to_date('05-DEC-2016', 'dd-MON-yyyy'), 103);

1 row created.

SQL> INSERT INTO Customer VALUES (5, 'JEREMY WYATT', '23 BEACON ST',
652659082, 'JEREMY123', 'WJEREMYe55', to_date('09-AUG-2019', 'dd-MON-yyyy'), 104);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (6, 'JENNY KEATING', '40 IROQUOIS ST',
651651678, 'JENNY123', 'KJENNY6', to_date('30-APR-2017', 'dd-MON-yyyy'), 105);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (7, 'OLIVIA SANDERS', '75 ST. ALPHONSUS ST',
879061237, 'OLIVIA123', 'SOLIVIA7', to_date('28-FEB-2018', 'dd-MON-yyyy'), 106);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (8, 'MONICA MARK', '890 PETERBOROUGH ST',
879025497, 'MONICA123', 'MMONICA8', to_date('15-JAN-2019','dd-MON-yyyy'), 107);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (10, 'RACHEL KAREN', '12 SUMMER ST',
879065497, 'RACHEL123', 'KRACHEL0', to_date('01-SEP-2019', 'dd-MON-yyyy'), 109);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (3, 'CHRISTINA LUDDINGTON', '1186
BOYLSTON ST', 879054670, 'CHRISTINA123', 'LCHRISTNA3',to_date('21-MAY-2017','dd-
MON-yyyy'), 102);
```

1 row created.

```
SQL> INSERT INTO Customer VALUES (9, 'STEPHANIE NIELSON', '165 RIVERWAY
PKWY', 879089097, 'STEPHANIE123', 'NSTEPHNI9', to_date('25-MAR-2018', 'dd-MON-
yyyy'), 108);
```

1 row created.

```
SQL> commit;
```

Commit complete.

```
SQL> INSERT INTO Rent VALUES (101, 'B2A123', to_date('10-DEC-2019','dd-MON-yyyy'),
to_date('20-DEC-2019','dd-MON-yyyy'));
```

1 row created.

```
SQL> INSERT INTO Rent VALUES (102, 'B1Z123', to_date('10-NOV-2019','dd-MON-yyyy'),
to_date('25-NOV-2019','dd-MON-yyyy'));
```

1 row created.

SQL> INSERT INTO Rent VALUES (114, 'B1B234', to_date('01-OCT-2019','dd-MON-yyyy'), to_date('21-OCT-2019','dd-MON-yyyy'));

1 row created.

SQL> INSERT INTO Rent VALUES (105, 'B1H123', to_date('02-DEC-2019','dd-MON-yyyy'), to_date('25-DEC-2019','dd-MON-yyyy'));

1 row created.

SQL> INSERT INTO Rent VALUES (110, 'B1L321', to_date('04-DEC-2019','dd-MON-yyyy'), to_date('26-DEC-2019','dd-MON-yyyy'));

1 row created.

SQL> INSERT INTO Rent VALUES (112, 'B1P321', to_date('11-DEC-2019','dd-MON-yyyy'), to_date('29-DEC-2019','dd-MON-yyyy'));

1 row created.

SQL> SELECT * FROM CARD;

CARDID	S	FINES
100	A	0
101	A	0
102	A	0
103	A	13
104	A	0
105	A	0
106	B	0
107	B	10
108	B	20
109	B	0
110	A	0

CARDID	S	FINES
111	A	0
112	A	6
113	A	0
114	A	0

15 rows selected.

SQL> SELECT * FROM BRANCH;

NAME

ADDRESS PHONE
---------- ----------
COMPUTER SCIENCE
FIRST FLR, SEC A 857303838

CHEMISTRY
SECOND FLR, SEC B 622863281

BIOLOGY
THIRD FLR, SEC B 642908944

NAME

ADDRESS PHONE
---------- ----------
PHYSICS
FIRST FLR, SEC C 768903666

MATHEMATICS
SECOND FLR, SEC A 657890561

SQL> SELECT * FROM CUSTOMER;

CUSTOMERID NAME
---------- ----------
CUSTOMERADDRESS PHONE
---------- ----------
PASSWORD USERNAME SIGNUPDAT CARDNUMBER
---------- ---------- --------- ----------
1 NIDHI REDDY
170 PARKER HILL 651098656
NIDHI123 RNIDHI1 14-APR-16 100

2 ALEX WILLIAMS
45 PARK DRIVE ABBEY 615516890
ALEX123 WALEX22 10-JUN-18 101

CUSTOMERID NAME

---------- ----------
CUSTOMERADDRESS PHONE
---------- ----------
PASSWORD USERNAME SIGNUPDAT CARDNUMBER
---------- ---------- --------- ----------

4 TOM PETERS
45 TREMONT ST 658530958
tom123 PTOM4 05-DEC-16 103

5 JEREMY WYATT
23 BEACON ST 652659082

CUSTOMERID NAME
---------- ----------
CUSTOMERADDRESS PHONE
---------- ----------
PASSWORD USERNAME SIGNUPDAT CARDNUMBER
---------- ---------- --------- ----------
JEREMY123 WJEREMYe55 09-AUG-19 104

6 JENNY KEATING
40 IROQUOIS ST 651651678
JENNY123 KJENNY6 30-APR-17 105

7 OLIVIA SANDERS

CUSTOMERID NAME
---------- ----------
CUSTOMERADDRESS PHONE
---------- ----------
PASSWORD USERNAME SIGNUPDAT CARDNUMBER
---------- ---------- --------- ----------
75 ST. ALPHONSUS ST 879061237
OLIVIA123 SOLIVIA7 28-FEB-18 106

8 MONICA MARK
890 PETERBOROUGH ST 879025497
MONICA123 MMONICA8 15-JAN-19 107

CUSTOMERID NAME
---------- ----------
CUSTOMERADDRESS PHONE
---------- ----------
PASSWORD USERNAME SIGNUPDAT CARDNUMBER

```
-------------------- ---------- --------- ----------
       10 RACHEL KAREN
12 SUMMER ST                                879065497
RACHEL123          KRACHEL0   01-SEP-19       109

        3 CHRISTINA LUDDINGTON
1186 BOYLSTON ST                            879054670
CHRISTINA123       LCHRISTNA3 21-MAY-17       102

CUSTOMERID NAME
---------- ----------------------------------------
CUSTOMERADDRESS                                 PHONE
---------------------------------------------------- ----------
PASSWORD           USERNAME   SIGNUPDAT CARDNUMBER
-------------------- ---------- --------- ----------

        9 STEPHANIE NIELSON
165 RIVERWAY PKWY                           879089097
STEPHANIE123       NSTEPHNI9  25-MAR-18       108


10 rows selected.

SQL> SELECT * FROM EMPLOYEE;

EMPLOYEEID NAME
---------- ----------------------------------------
EMPLOYEEADDRESS                                 PHONE
---------------------------------------------------- ----------
PASSWORD           USERNAME     PAYCHECK
-------------------- ---------- ----------
BRANCHNAME                     CARDNUMBER
---------------------------------------- ----------
      211 SAM
150 PARKMAN ST HOUSE                        671671671
SAM123             SAM1           5000
COMPUTER SCIENCE                    110


EMPLOYEEID NAME
---------- ----------------------------------------
EMPLOYEEADDRESS                                 PHONE
---------------------------------------------------- ----------
PASSWORD           USERNAME     PAYCHECK
-------------------- ---------- ----------
BRANCHNAME                     CARDNUMBER
```

---------- ----------
212 WES
13 HEARTH ST 688688688
WES123 WES12 3500.5
CHEMISTRY 111

EMPLOYEEID NAME
---------- ----------
EMPLOYEEADDRESS PHONE
---------- ----------
PASSWORD USERNAME PAYCHECK
---------- ---------- ----------
BRANCHNAME CARDNUMBER
---------- ----------
213 ASHER
76 PERTH AVE 628628628
ASHER123 ASHER13 4570.75
BIOLOGY 112

EMPLOYEEID NAME
---------- ----------
EMPLOYEEADDRESS PHONE
---------- ----------
PASSWORD USERNAME PAYCHECK
---------- ---------- ----------
BRANCHNAME CARDNUMBER
---------- ----------
214 VICTOR
89 INDIA ST 654321987
VICTOR123 VICTOR14 5575
PHYSICS 113

EMPLOYEEID NAME
---------- ----------
EMPLOYEEADDRESS PHONE
---------- ----------
PASSWORD USERNAME PAYCHECK
---------- ---------- ----------
BRANCHNAME CARDNUMBER
---------- ----------
215 SABRINA
100 ITALY PKWY 698754321
SABRINA123 SABRINA5 5050.5

MATHEMATICS 114

SQL> SELECT * FROM LOCATION;

ADDRESS

FIRST FLR, SEC A
SECOND FLR, SEC B
THIRD FLR, SEC B
FIRST FLR, SEC C
SECOND FLR, SEC A

SQL> SELECT * FROM BOOK;

ISBN BOOKID STATE A DAMAGECOST LOSTCOST
---- ------ ---------- - ---------- ----------
ADDRESS

A123 B1A123 GOOD A 5 20
FIRST FLR, SEC A

A123 B2A123 NEW O 6 30
SECOND FLR, SEC B

B234 B1B234 NEW A 2 15
THIRD FLR, SEC B

ISBN BOOKID STATE A DAMAGECOST LOSTCOST
---- ------ ---------- - ---------- ----------
ADDRESS

C321 B1C321 BAD A 1 10
FIRST FLR, SEC C

H123 B1H123 GOOD A 3 15
SECOND FLR, SEC A

Z123 B1Z123 GOOD O 4 20
THIRD FLR, SEC B

ISBN BOOKID STATE A DAMAGECOST LOSTCOST
---- ------ ---------- - ---------- ----------
ADDRESS

```
--------------------------------------------------
L321 B1L321 NEW       O         4         20
FIRST FLR, SEC A

P321 B1P321 USED      A         2         12
SECOND FLR, SEC A


8 rows selected.

SQL> SELECT * FROM RENT;

   CARDID ITEMID BORROWDAT RETURNDAT
---------- ------ --------- ---------
       101 B2A123 10-DEC-19 20-DEC-19
       102 B1Z123 10-NOV-19 25-NOV-19
       114 B1B234 01-OCT-19 21-OCT-19
       105 B1H123 02-DEC-19 25-DEC-19
       110 B1L321 04-DEC-19 26-DEC-19
       112 B1P321 11-DEC-19 29-DEC-19

6 rows selected.

SQL> commit;

Commit complete.

SQL> set linesize 200;
SQL> SELECT * FROM EMPLOYEE;

EMPLOYEEID NAME                           EMPLOYEEADDRESS
PHONE PASSWORD           USERNAME     PAYCHECK BRANCHNAME
---------- ------------------------------------ ---------------------------------------------- ---------- ----
---------------- ---------- ---------- ---------------------------------------
CARDNUMBER
----------
       211 SAM                            150 PARKMAN ST HOUSE
671671671 SAM123             SAM1            5000 COMPUTER SCIENCE
       110

       212 WES                            13 HEARTH ST                                    688688688
WES123             WES12          3500.5 CHEMISTRY
       111

       213 ASHER                          76 PERTH AVE                                    628628628
ASHER123           ASHER13        4570.75 BIOLOGY
```

```
     112


EMPLOYEEID NAME                           EMPLOYEEADDRESS
PHONE PASSWORD          USERNAME    PAYCHECK BRANCHNAME
---------- ------------------------------------ ------------------------------------------------- ---------- ----
---------- ---------- ---------- ------------------------------------
CARDNUMBER
----------
     214 VICTOR                        89 INDIA ST                                654321987
VICTOR123         VICTOR14       5575 PHYSICS
     113

     215 SABRINA                       100 ITALY PKWY                             698754321
SABRINA123        SABRINA5      5050.5 MATHEMATICS
     114


SQL> column name format a30
SQL> SELECT * FROM EMPLOYEE;

EMPLOYEEID NAME                           EMPLOYEEADDRESS
PHONE PASSWORD          USERNAME    PAYCHECK BRANCHNAME
               CARDNUMBER
---------- ------------------------------ ------------------------------------------------- ---------- ----------
------ ---------- ---------- ------------------------------------ ----------
     211 SAM                       150 PARKMAN ST HOUSE                       671671671
SAM123            SAM1          5000 COMPUTER SCIENCE
                    110
     212 WES                       13 HEARTH ST                              688688688 WES123
WES12        3500.5 CHEMISTRY
                    111
     213 ASHER                     76 PERTH AVE                               628628628
ASHER123          ASHER13      4570.75 BIOLOGY
                    112
     214 VICTOR                    89 INDIA ST                               654321987
VICTOR123         VICTOR14       5575 PHYSICS
                    113
     215 SABRINA                   100 ITALY PKWY                             698754321
SABRINA123        SABRINA5      5050.5 MATHEMATICS
                    114

SQL> column employeeaddress format a25;
SQL> SELECT * FROM EMPLOYEE;
```

```
EMPLOYEEID NAME                 EMPLOYEEADDRESS          PHONE
PASSWORD          USERNAME    PAYCHECK BRANCHNAME
CARDNUMBER
---------- ---------------------------- ---------------------- ---------- ------------------- ---------- -------
--- ---------------------------------------- ----------
     211 SAM                    150 PARKMAN ST HOUSE      671671671 SAM123
SAM1          5000 COMPUTER SCIENCE                        110
     212 WES                    13 HEARTH ST           688688688 WES123          WES12
3500.5 CHEMISTRY                                111
     213 ASHER                  76 PERTH AVE           628628628 ASHER123
ASHER13      4570.75 BIOLOGY                              112
     214 VICTOR                 89 INDIA ST            654321987 VICTOR123
VICTOR14       5575 PHYSICS                              113
     215 SABRINA                100 ITALY PKWY         698754321 SABRINA123
SABRINA5      5050.5 MATHEMATICS                           114

SQL> column branchname format a20;
SQL> SELECT * FROM EMPLOYEE;

EMPLOYEEID NAME                 EMPLOYEEADDRESS          PHONE
PASSWORD          USERNAME    PAYCHECK BRANCHNAME         CARDNUMBER
---------- ---------------------------- ---------------------- ---------- ------------------- ---------- -------
--- -------------------- ----------
     211 SAM                    150 PARKMAN ST HOUSE      671671671 SAM123
SAM1          5000 COMPUTER SCIENCE          110
     212 WES                    13 HEARTH ST           688688688 WES123          WES12
3500.5 CHEMISTRY               111
     213 ASHER                  76 PERTH AVE           628628628 ASHER123
ASHER13      4570.75 BIOLOGY                112
     214 VICTOR                 89 INDIA ST            654321987 VICTOR123
VICTOR14       5575 PHYSICS                  113
     215 SABRINA                100 ITALY PKWY         698754321 SABRINA123
SABRINA5      5050.5 MATHEMATICS              114

SQL> SELECT * FROM CUSTOMER;

CUSTOMERID NAME                 CUSTOMERADDRESS
PHONE PASSWORD          USERNAME   SIGNUPDAT CARDNUMBER
---------- ---------------------------- -------------------------------------------------- ---------- --------------
------ ---------- --------- ----------
       1 NIDHI REDDY                170 PARKER HILL                                  651098656
NIDHI123          RNIDHI1    14-APR-16       100
       2 ALEX WILLIAMS              45 PARK DRIVE ABBEY
615516890 ALEX123          WALEX22    10-JUN-18       101
       4 TOM PETERS                 45 TREMONT ST                                    658530958
tom123          PTOM4      05-DEC-16       103
```

```
      5 JEREMY WYATT                23 BEACON ST                                652659082
JEREMY123          WJEREMYe55 09-AUG-19       104
      6 JENNY KEATING               40 IROQUOIS ST                              651651678
JENNY123           KJENNY6    30-APR-17       105
      7 OLIVIA SANDERS              75 ST. ALPHONSUS ST                          879061237
OLIVIA123          SOLIVIA7   28-FEB-18       106
      8 MONICA MARK                 890 PETERBOROUGH ST
879025497 MONICA123          MMONICA8   15-JAN-19       107
     10 RACHEL KAREN                12 SUMMER ST                                879065497
RACHEL123          KRACHEL0   01-SEP-19       109
      3 CHRISTINA LUDDINGTON        1186 BOYLSTON ST
879054670 CHRISTINA123       LCHRISTNA3 21-MAY-17       102
      9 STEPHANIE NIELSON           165 RIVERWAY PKWY
879089097 STEPHANIE123       NSTEPHNI9  25-MAR-18       108

10 rows selected.

SQL> column borrowdate format a10
SQL> column returndate format a10
SQL> SELECT * FROM RENT;

SQL> SELECT * FROM RENT;

   CARDID ITEMID BORROWDATE RETURNDATE
---------- ------ ---------- ----------
       101 B2A123 10-DEC-19  20-DEC-19
       102 B1Z123 10-NOV-19  25-NOV-19
       114 B1B234 01-OCT-19  21-OCT-19
       105 B1H123 02-DEC-19  25-DEC-19
       110 B1L321 04-DEC-19  26-DEC-19
       112 B1P321 11-DEC-19  29-DEC-19

6 rows selected.

SQL> SELECT * FROM LOCATION;

ADDRESS
--------------------------------------------------
FIRST FLR, SEC A
SECOND FLR, SEC B
THIRD FLR, SEC B
FIRST FLR, SEC C
SECOND FLR, SEC A

SQL> SELECT * FROM BOOK;
```

```
ISBN BOOKID STATE      A DAMAGECOST   LOSTCOST ADDRESS
---- ------ ---------- - ---------- ---------- ------------------------------------------------
A123 B1A123 GOOD       A          5         20 FIRST FLR, SEC A
A123 B2A123 NEW        O          6         30 SECOND FLR, SEC B
B234 B1B234 NEW        A          2         15 THIRD FLR, SEC B
C321 B1C321 BAD        A          1         10 FIRST FLR, SEC C
H123 B1H123 GOOD       A          3         15 SECOND FLR, SEC A
Z123 B1Z123 GOOD       O          4         20 THIRD FLR, SEC B
L321 B1L321 NEW        O          4         20 FIRST FLR, SEC A
P321 B1P321 USED       A          2         12 SECOND FLR, SEC A

8 rows selected.

SQL> SELECT * FROM CARD;

   CARDID S      FINES
---------- - ----------
       100 A          0
       101 A          0
       102 A          0
       103 A         13
       104 A          0
       105 A          0
       106 B          0
       107 B         10
       108 B         20
       109 B          0
       110 A          0

   CARDID S      FINES
---------- - ----------
       111 A          0
       112 A          6
       113 A          0
       114 A          0

15 rows selected.


SQL> select cardID, name from employee join card on employee.cardNumber=card.cardID;

   CARDID NAME
---------- ------------------------------
       110 SAM
       111 WES
       112 ASHER
```

```
      113 VICTOR
      114 SABRINA


SQL> select * from book where avalability='O';

ISBN BOOKID STATE      A DAMAGECOST   LOSTCOST ADDRESS
---- ------ ---------- - ---------- ---------- ----------------------------------------------
A123 B2A123 NEW        O          6         30 SECOND FLR, SEC B
Z123 B1Z123 GOOD       O          4         20 THIRD FLR, SEC B
L321 B1L321 NEW        O          4         20 FIRST FLR, SEC A

SQL> CREATE DIRECTORY ext_table_dir AS
'C:\Users\sahan\Desktop\branch_data_load.csv';

Directory created.

SQL> grant read on directory ext_table_dir to lib_adm
  2  ;

Grant succeeded.

SQL>  create table branch_load(
  2   name varchar2(25),
  3   adress varchar2(40),
  4   phoneNumber number
  5   )
  6   organization external (
  7    type oracle_loader
  8    default directory ext_table_dir
  9    access parameters (
 10     nobadfile
 11     fields terminated by ',')
 12     location('branch_data_load.csv')
 13     )
 14     reject limit unlimited
 15     /

Table created.

SQL> desc branch_load;
 Name                                                              Null?    Type
 ----------------------------------------------------------------------------------------------- --
------ -------------------------------------------------------------------------
 NAME
VARCHAR2(25)
```

```
 ADRESS
VARCHAR2(40)
 PHONENUMBER
NUMBER

SQL> column name format a30
SQL> desc branch_load;
 Name                                                                                                    Null?    Type
 ----------------------------------------------------------------------------------------------------------------- --
------ ----------------------------------------------------------------------
 NAME
VARCHAR2(25)
 ADRESS
VARCHAR2(40)
 PHONENUMBER
NUMBER

SQL> set linesize 100
SQL> desc branch_load;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------------
 NAME                                               VARCHAR2(25)
 ADRESS                                             VARCHAR2(40)
 PHONENUMBER                                        NUMBER


SQL>   create or replace view rent_customer as
  2    select cardID,name
  3    from customer inner join rent
  4    on rent.cardID=customer.cardNumber;

View created.

SQL> select * from rent_customer;

    CARDID NAME
---------- ------------------------------
       101 ALEX WILLIAMS
       105 JENNY KEATING
       102 CHRISTINA LUDDINGTON

SQL>   create or replace view rent_customer as
  2    select cardID,name
  3    from customer inner join rent
  4    on rent.cardID=customer.cardNumber;
```

View created.

```
SQL>
SQL>   select branch.name, count(*),
  2      to_char((count(*)/emp.cnt)*100,'99.99')||'%' Employee_Percentage
  3    from branch,
  4    employee,
  5     (select count(*) cnt
  6   from employee) emp
  7   where branch.name=employee.branchname
  8   group by branch.name, emp.cnt
  9   /

NAME                         COUNT(*) EMPLOYE
---------------------------- ---------- -------
COMPUTER SCIENCE                      1  20.00%
MATHEMATICS                           1  20.00%
PHYSICS                               1  20.00%
CHEMISTRY                             1  20.00%
BIOLOGY                               1  20.00%

SQL> column employee_percentage format a20;
SQL>  /

NAME                         COUNT(*) EMPLOYEE_PERCENTAGE
---------------------------- ---------- --------------------
COMPUTER SCIENCE                      1  20.00%
MATHEMATICS                           1  20.00%
PHYSICS                               1  20.00%
CHEMISTRY                             1  20.00%
BIOLOGY                               1  20.00%

SQL>   select branch.name, count(*),
  2      to_char((count(*)/emp.cnt)*100,'99.99')||'%' Employee_Percentage
  3    from branch,
  4    employee,
  5     (select count(*) cnt
  6   from employee) emp
  7   where branch.name=employee.branchname
  8   group by branch.name, emp.cnt
  9   /

NAME                         COUNT(*) EMPLOYEE_PERCENTAGE
---------------------------- ---------- --------------------
COMPUTER SCIENCE                      1  20.00%
MATHEMATICS                           1  20.00%
```

```
PHYSICS                        1  20.00%
CHEMISTRY                         1  20.00%
BIOLOGY                         1  20.00%

SQL>  create index cust_idx
  2   on customer(name);

Index created.

SQL>  create materialized view renters_mv
  2   build immediate
  3   refresh on commit
  4   as
  5   select cardID, count(*) Rentals
  6   from rent
  7   group by cardID;

Materialized view created.

SQL> select * from renters_mv;

   CARDID    RENTALS
---------- ----------
      101          1
      102          1
      105          1
      110          1
      112          1
      114          1

6 rows selected.

SQL>  declare
  2    cursor card_cur(p_status in varchar2)
  3    is select *
  4    from card
  5    where card.status=p_status;
  6
  7    l_card card%rowtype;
  8    begin
  9     dbms_output.put_line('Getting Blocked card owners');
 10      open card_cur('B');
 11      loop
 12       fetch card_cur into l_card;
 13  exit when card_cur%notfound;
 14  dbms_output.put('Card ID ' || l_card.cardID || ' is ');
```

```
 15         dbms_output.put_line(l_card.status);
 16  end loop;
 17  close card_cur;
 18        end;
 19       /

PL/SQL procedure successfully completed.

SQL> set serveroutput on
SQL>  declare
  2    cursor card_cur(p_status in varchar2)
  3    is select *
  4    from card
  5    where card.status=p_status;
  6
  7    l_card card%rowtype;
  8    begin
  9     dbms_output.put_line('Getting Blocked card owners');
 10     open card_cur('B');
 11     loop
 12      fetch card_cur into l_card;
 13  exit when card_cur%notfound;
 14  dbms_output.put('Card ID ' || l_card.cardID || ' is ');
 15         dbms_output.put_line(l_card.status);
 16  end loop;
 17  close card_cur;
 18        end;
 19       /
Getting Blocked card owners
Card ID 106 is B
Card ID 107 is B
Card ID 108 is B
Card ID 109 is B

PL/SQL procedure successfully completed.

SQL> INSERT INTO Rent VALUES (103, 'B1C321', to_date('19-NOV-2019','dd-MON-yyyy'),
to_date('19-DEC-2019','dd-MON-yyyy'));

1 row created.


SQL> CREATE OR REPLACE PROCEDURE handle_Returns(l_ItemID IN VARCHAR2)
  2  IS
  3    l_rented NUMBER;
  4    l_book NUMBER;
```

```
 5
 6  BEGIN
 7    SELECT COUNT(*) INTO l_rented
 8    FROM rent
 9    WHERE itemid LIKE l_ItemID;
10
11    SELECT COUNT(*) INTO l_book
12    FROM book
13    WHERE bookid LIKE l_ItemID;
14
15    IF l_rented > 0 THEN
16      DELETE FROM rent
17      WHERE itemid = l_ItemID;
18      IF l_Book > 0 THEN
19        UPDATE book
20        SET avalability = 'A'
21        WHERE bookid LIKE l_ItemID;
22        DBMS_OUTPUT.PUT_LINE('The book ' || l_ItemID || ' is now available.');
23      END IF;
24    ELSE
25      DBMS_OUTPUT.PUT_LINE('This item is not rented at the moment');
26    END IF;
27    EXCEPTION WHEN no_data_found THEN
28    DBMS_OUTPUT.PUT_LINE('Item ID incorrect');
29  END;
30  /
```

Procedure created.

```
SQL> execute handle_Returns(103);
This item is not rented at the moment

PL/SQL procedure successfully completed.

SQL> execute handle_Returns('B1C321');
The book B1C321 is now available.

PL/SQL procedure successfully completed.

SQL> INSERT INTO Rent VALUES (103, 'B1C321', to_date('19-NOV-2019','dd-MON-yyyy'),
to_date('19-DEC-2019','dd-MON-yyyy'));

1 row created.

SQL> CREATE OR REPLACE PROCEDURE handle_Returns(l_ItemID IN VARCHAR2)
  2  IS
```

```
 3   l_rented NUMBER;
 4   l_book NUMBER;
 5
 6  BEGIN
 7    SELECT COUNT(*) INTO l_rented
 8    FROM rent
 9    WHERE itemid LIKE l_ItemID;
10
11    SELECT COUNT(*) INTO l_book
12    FROM book
13    WHERE bookid LIKE l_ItemID;
14
15    IF l_rented > 0 THEN
16      DELETE FROM rent
17      WHERE itemid = l_ItemID;
18      IF l_Book > 0 THEN
19        UPDATE book
20        SET avalability = 'A'
21        WHERE bookid LIKE l_ItemID;
22        DBMS_OUTPUT.PUT_LINE('The book ' || l_ItemID || ' is now available.');
23      END IF;
24    ELSE
25      DBMS_OUTPUT.PUT_LINE('This item is not rented at the moment');
26    END IF;
27    EXCEPTION WHEN no_data_found THEN
28    DBMS_OUTPUT.PUT_LINE('Item ID incorrect');
29  END;
30  /

Procedure created.

SQL> execute handle_Returns('B1C321');
The book B1C321 is now available.

PL/SQL procedure successfully completed.

SQL> CREATE OR REPLACE TRIGGER modify_Fines
 2  AFTER DELETE
 3  ON rent
 4  FOR EACH ROW
 5  DECLARE
 6    l_CardID NUMBER;
 7    l_ItemID VARCHAR2(6);
 8    l_Book NUMBER;
 9    l_damage NUMBER;
10  BEGIN
```

```
 11   SELECT cardid, itemid INTO l_CardID, l_ItemID
 12   FROM rent
 13   WHERE cardid LIKE :old.cardid;
 14
 15   SELECT COUNT(*) INTO l_Book
 16   FROM book
 17   WHERE bookid LIKE l_ItemID;
 18
 19     IF sysdate > :old.returndate THEN
 20     ELSIF l_Book > 0 THEN
 21       SELECT damageCost INTO l_damage
 22       FROM book
 23       WHERE bookid LIKE l_ItemID;
 24     END IF;
 25
 26     UPDATE card
 27     SET status = 'B', fines = (fines + l_damage)
 28     WHERE cardid LIKE l_CardID;
 29   ELSE
 30     DBMS_OUTPUT.PUT_LINE('The item has been return before deadline');
 31   END IF;
 32  END;
 33  /
CREATE OR REPLACE TRIGGER modify_Fines
                  *
ERROR at line 1:
ORA-04089: cannot create triggers on objects owned by SYS


SQL> INSERT INTO Book VALUES ('Q123', 'B2H123', 'GOOD', 'A', 3, 15, 'SECOND FLR,
SEC A');

1 row created.

SQL> COMMIT;

Commit complete.

SQL> INSERT INTO Book VALUES ('D123', 'B2Z123', 'GOOD', 'O', 4, 20, 'THIRD FLR, SEC
B');

1 row created.

SQL> SAVEPOINT A;

Savepoint created.
```

```
SQL> INSERT INTO Book VALUES ('E321', 'B2L321', 'NEW', 'O', 4, 20, 'FIRST FLR, SEC
C');

1 row created.

SQL> INSERT INTO Book VALUES ('F321', 'B2P321', 'USED', 'A', 2, 12, 'SECOND FLR,
SEC B');

1 row created.

SQL> ROLLBACK TO A;

Rollback complete.

SQL> select * from book;

ISBN BOOKID STATE      A DAMAGECOST   LOSTCOST ADDRESS
---- ------ ---------- - ---------- ---------- --------------------------------------------------
A123 B1A123 GOOD       A          5         20 FIRST FLR, SEC A
A123 B2A123 NEW        O          6         30 SECOND FLR, SEC B
B234 B1B234 NEW        A          2         15 THIRD FLR, SEC B
C321 B1C321 BAD        A          1         10 FIRST FLR, SEC C
H123 B1H123 GOOD       A          3         15 SECOND FLR, SEC A
Z123 B1Z123 GOOD       O          4         20 THIRD FLR, SEC B
L321 B1L321 NEW        O          4         20 FIRST FLR, SEC A
P321 B1P321 USED       A          2         12 SECOND FLR, SEC A
Q123 B2H123 GOOD       A          3         15 SECOND FLR, SEC A
D123 B2Z123 GOOD       O          4         20 THIRD FLR, SEC B

10 rows selected.

SQL> CREATE OR REPLACE FUNCTION Emp_details (empid in number)
  2  RETURN VARCHAR2
  3  IS emp VARCHAR2(200);
  4  BEGIN
  5  SELECT '1)Name-' ||Employee.name|| ' 2)Address -' || Employee.employeeAddress || '
3)Work Location -' ||Employee.branchName into emp
  6  from Employee, Branch where
  7  Employee.branchname=Branch.name
  8  and Employee.EmployeeID=empid;
  9  RETURN(emp);
 10  END Emp_details;
 11  /

Function created.
```

```
SQL> select emp_details(211) as "Employee Address" FROM DUAL;

Employee Address
--------------------------------------------------------------------------------------------------
1)Name-SAM 2)Address -150 PARKMAN ST HOUSE 3)Work Location -COMPUTER
SCIENCE


SQL> declare
  2   type  book_condition is record(
  3     bookID varchar2(6),
  4     state varchar2(10),
  5     avalability varchar2(1)
  6     );
  7     type book_condition_refcur_type is ref cursor
  8     return book_condition;
  9
 10      book_condition_refcur book_condition_refcur_type;
 11      book_cond book_condition;
 12
 13   begin
 14     open book_condition_refcur for
 15      select bookID, state, avalability
 16  from book b, rent r
 17  where b.bookID=r.ItemID
 18  order by 1;
 19
 20  fetch book_condition_refcur into book_cond;
 21  while book_condition_refcur%found loop
 22    dbms_output.put_line(book_cond.bookID||' is in '||book_cond.state||' and has avalability '||
book_cond.avalability);
 23    fetch book_condition_refcur into book_cond;
 24  end loop;
 25  end;
 26  /
B1B234 is in NEW and has avalability A
B1H123 is in GOOD and has avalability A
B1L321 is in NEW and has avalability O
B1P321 is in USED and has avalability A
B1Z123 is in GOOD and has avalability O
B2A123 is in NEW and has avalability O

PL/SQL procedure successfully completed.

SQL> declare
```

```
  2  l_bookid varchar2(10);
  3  l_dc number;
  4  begin
  5  l_bookid := 'B1A123';
  6  l_dc := 'A';
  7
  8  exception
  9  when VALUE_ERROR then
 10  dbms_output.put_line('We encountered the VALUE_ERROR exception');
 11  end;
 12  /
We encountered the VALUE_ERROR exception

PL/SQL procedure successfully completed.

SQL> show pdbs;

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         5 LIBRARY_PROJECT                READ WRITE NO
```