TABLE:

CREATE TABLE "MEMBER"

(

"ID" NUMBER NOT NULL,

"NAME" VARCHAR2(64) NOT NULL,

"USERNAME" VARCHAR2(64) NOT NULL,

"PASSWORD" VARCHAR2(64) NOT NULL,

"PHONE" VARCHAR2(64) NOT NULL,

"EMAIL" VARCHAR2(64) NOT NULL,

"TYPE" VARCHAR2(16) NOT NULL,

PRIMARY KEY ("ID") ENABLE

)

CREATE TABLE "BOOK"

(

"ID" NUMBER NOT NULL,

"NAME" VARCHAR2(64) NOT NULL,

"AUTHOR" VARCHAR2(64) NOT NULL,

"PUBLISHER" VARCHAR2(64) NOT NULL,

"BOOK\_SELF\_ID" NUMBER NOT NULL,

"TOTAL\_QUANTITY" NUMBER NOT NULL,

"AVAILABLE\_QUANTITY" NUMBER NOT NULL,

"ISSUED\_QUANTITY" NUMBER NOT NULL,

"AVAILABLE" NUMBER(1,0) NOT NULL,

PRIMARY KEY ("ID") ENABLE,

FOREIGN KEY ("BOOK\_SELF\_ID") REFERENCES BOOK\_SELF("ID") ENABLE

)

CREATE TABLE "BOOK\_SELF"

(

"ID" NUMBER NOT NULL,

"NAME" VARCHAR2(64) NOT NULL,

"CATAGORY" VARCHAR2(64) NOT NULL,

"NUMB" VARCHAR2(64) NOT NULL,

PRIMARY KEY ("ID") ENABLE

)

CREATE TABLE "HISTORY"

(

"HID" NUMBER NOT NULL,

"MID" NUMBER NOT NULL,

"BID" NUMBER NOT NULL,

"ISSUE\_DATE" VARCHAR2(64) NOT NULL,

"DUE\_DATE" VARCHAR2(64) NOT NULL,

"RETURN\_DATE" VARCHAR2(64) NOT NULL,

"STATUS" VARCHAR2(64) NOT NULL,

PRIMARY KEY ("HID") ENABLE,

FOREIGN KEY ("BID") REFERENCES BOOK("ID") ENABLE,

FOREIGN KEY ("MID") REFERENCES MEMBER("ID") ENABLE

)

VIEW:

CREATE OR REPLACE FORCE VIEW SYSTEM.VIEW\_HISTORY("ID", "MNAME", "EMAIL", "PHONE", "BNAME", "AUTHOR","SELF\_NAME","TOTAL\_BOOK","AVAILABLE\_BOOK","ISSUED\_BOOK","BOOK\_MODE","ISSUE\_DATE","RETURN\_DATE","DUE\_DATE","STATUS") AS

SELECT h.hid,m.name,m.email,m.phone,b.name,b.author,b.self\_name,b.total\_quantity,b.available\_quantity,b.issued\_quantity,b.available,h.issue\_date,h.return\_date,h.due\_date,h.status FROM member m,history h,view\_book b where m.id = h.mid and b.id = h.bid;

CREATE OR REPLACE FORCE VIEW SYSTEM.VIEW\_BOOK("ID", "NAME", "AUTHOR", "PUBLISHER", "TOTAL\_QUANTITY", "AVAILABLE\_QUANTITY", "ISSUED\_QUANTITY", "AVAILABLE", "SELF\_ID", "SELF\_NAME","SELF\_NO","CATEGORY") AS

SELECT b.id,b.name,b.author,b.publisher,b.total\_quantity,b.available\_quantity,b.issued\_quantity,b.available,bs.id,bs.name,bs.category,bs.numb FROM book b

JOIN book\_self bs

ON bs.id = b.book\_self\_id;

SEQUENCE:

CREATE SEQUENCE SYSTEM.MEMBER\_SEQ MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

CREATE SEQUENCE "SYSTEM"."BOOK\_SEQ" MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

CREATE SEQUENCE SYSTEM.BOOK\_SELF\_SEQ MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

CREATE SEQUENCE SYSTEM.History\_SEQ MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

TRIGGER:

CREATE OR REPLACE TRIGGER SYSTEM.MEMBER\_TRI

before insert on "MEMBER"

for each row

begin

if :NEW."ID" is null then

select "MEMBER\_SEQ".nextval into :NEW."ID" from dual;

end if;

end;

ALTER TRIGGER SYSTEM.MEMBER\_TRI ENABLE;

CREATE OR REPLACE TRIGGER SYSTEM.BOOK\_TRI

before insert on "BOOK"

for each row

begin

if :NEW."ID" is null then

select "BOOK\_SEQ".nextval into :NEW."ID" from dual;

end if;

end;

ALTER TRIGGER SYSTEM.BOOK\_TRI ENABLE;

CREATE OR REPLACE TRIGGER SYSTEM.BOOK\_SELF\_TRI

before insert on "BOOK\_SELF"

for each row

begin

if :NEW."ID" is null then

select "BOOK\_SELF\_SEQ".nextval into :NEW."ID" from dual;

end if;

end;

ALTER TRIGGER SYSTEM.BOOK\_SELF\_TRI ENABLE;

CREATE OR REPLACE TRIGGER SYSTEM.HISTORY\_TRI

before insert on "HISTORY"

for each row

begin

if :NEW."HID" is null then

select "HISTORY\_SEQ".nextval into :NEW."HID" from dual;

end if;

end;

ALTER TRIGGER SYSTEM.HISTORY\_TRI ENABLE;

FUNCTIONS & PROCEDURES:

CREATE OR REPLACE FUNCTION Validate\_User(user IN varchar2)

RETURN boolean IS

b boolean := false;

num number(1) := 0;

BEGIN

SELECT count(\*) into num

FROM member where member.username = user;

IF num = 1 THEN

b := true;

ELSE

b := false;

END IF;

RETURN b;

END;

CREATE OR REPLACE FUNCTION Return\_User\_Type(user IN varchar2,pass IN varchar2)

RETURN number

IS

num number(2):=0;

typ MEMBER.TYPE%TYPE;

cursor c is SELECT MEMBER.TYPE FROM MEMBER WHERE MEMBER.USERNAME = user AND MEMBER.PASSWORD = pass;

BEGIN

OPEN c;

FETCH c into typ;

CLOSE c;

IF typ='admin' THEN

num := 1;

ELSIF typ='user' THEN

num := 2;

ELSIF typ='librarian' THEN

num := 3;

ELSE

num := 0;

END IF;

RETURN num;

END;

CREATE OR REPLACE FUNCTION Return\_System\_Date()

RETURN HISTORY.ISSUE\_DATE%TYPE

IS

dat HISTORY.ISSUE\_DATE%TYPE;

cursor c is SELECT TO\_CHAR(SYSDATE, 'DD-MON-YY HH12:MI:SS AM')FROM DUAL;

BEGIN

OPEN c;

FETCH c into dat;

CLOSE c;

RETURN dat;

END;

CREATE OR REPLACE FUNCTION Return\_Due\_Date(d in number)

RETURN HISTORY.ISSUE\_DATE%TYPE

IS

dat HISTORY.ISSUE\_DATE%TYPE;

cursor c is SELECT TO\_CHAR(SYSDATE+d, 'DD-MON-YY HH12:MI:SS AM')FROM DUAL;

BEGIN

OPEN c;

FETCH c into dat;

CLOSE c;

RETURN dat;

END;

CREATE OR REPLACE FUNCTION CAL\_AVAILABLE\_QTY (bid in number, qty in number)

return number

as

a\_qty number := 0;

begin

select available\_quantity into a\_qty from book where id=bid;

a\_qty := a\_qty - qty;

return a\_qty;

end;

CREATE OR REPLACE FUNCTION CAL\_ISSUED\_QTY (bid in number, qty in number)

return number

as

i\_qty number := 0;

begin

select issued\_quantity into i\_qty from book where id=bid;

i\_qty := i\_qty + qty;

return i\_qty;

end;

CREATE OR REPLACE PROCEDURE CAL\_BOOK\_QTY (bid in number, qty in number)

as

qt number := 0;

begin

qt := CAL\_ISSUED\_QTY(bid,qty);

update book set issued\_quantity=qt where id=bid;

qt:=CAL\_AVAILABLE\_QTY(bid,qty);

update book set available\_quantity=qt where id=bid;

end;

CREATE OR REPLACE FUNCTION Compare\_Date(da in HISTORY.ISSUE\_DATE%TYPE)

RETURN number

IS

dat date;

s date;

cursor d is SELECT TO\_DATE(SYSDATE, 'DD-MON-YY HH12:MI:SS AM')FROM DUAL;

cursor c is SELECT TO\_DATE(da, 'DD-MON-YY HH12:MI:SS AM') FROM dual;

BEGIN

OPEN d;

FETCH d into s;

CLOSE d;

OPEN c;

FETCH c into dat;

CLOSE c;

if dat >= s then

return 1;

else

return 0;

end if;

END;