

```
create table table_publisher(Publisher_name varchar2(100) PRIMARY KEY,Publisher_address
varchar2(100) not null,publisher_phone varchar2(20) not null);
```

Table created.

```
SQL> desc table_publisher;
```

Name	Null?	Type

PUBLISHER_NAME	NOT NULL	VARCHAR2(100)
PUBLISHER_ADDRESS	NOT NULL	VARCHAR2(100)
PUBLISHER_PHONE	NOT NULL	VARCHAR2(20)

```
SQL> SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DEMOUSER';
```

GRANTEE PRIVILEGE	ADM	COM	INH

DEMOUSER CREATE TRIGGER	NO	NO	NO
DEMOUSER CREATE PROCEDURE	NO	NO	NO
DEMOUSER CREATE SEQUENCE	NO	NO	NO
DEMOUSER CREATE VIEW	NO	NO	NO
DEMOUSER CREATE TABLE	NO	NO	NO
DEMOUSER CREATE SESSION	NO	NO	NO

6 rows selected.

```
SQL> create sequence book_id_seq
```

```
2 start with 1
```

```
3 increment by 1
```

```
4 nocache;
```

Sequence created.

```
SQL> CREATE TABLE table_book (
  2  BookID NUMBER PRIMARY KEY,
  3  Book_Title VARCHAR2(100) NOT NULL,
  4  PublisherName VARCHAR2(100) NOT NULL
  5 );
```

Table created.

```
SQL> create or replace trigger book_id_trigger
  2 before insert on table_book
  3 for each row
  4 begin
  5 :NEW.BookID := book_id_seq.nextval;
  6 end;
  7 /
```

Trigger created.

```
SQL> CREATE TABLE table_library_branch (
  2  library_branch_BranchID INT PRIMARY KEY,
  3  library_branch_BranchName VARCHAR2(100) NOT NULL,
  4  library_branch_BranchAddress VARCHAR2(200) NOT NULL
  5 );
```

Table created.

```
SQL> desc table_library_branch;
```

Name	Null?	Type

LIBRARY_BRANCH_BRANCHID		NOT NULL NUMBER(38)

LIBRARY_BRANCH_BRANCHNAME	NOT NULL VARCHAR2(100)
LIBRARY_BRANCH_BRANCHADDRESS	NOT NULL VARCHAR2(200)

```
SQL> CREATE SEQUENCE borrower_seq
2 START WITH 100
3 INCREMENT BY 1;
```

Sequence created.

```
SQL> CREATE TABLE table_borrower (
2 CardNo INT PRIMARY KEY,
3 BorrowerName VARCHAR2(100) NOT NULL,
4 BorrowerAddress VARCHAR2(200) NOT NULL,
5 BorrowerPhone VARCHAR2(50) NOT NULL
6 );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER borrower_trigger
2 BEFORE INSERT ON table_borrower
3 FOR EACH ROW
4 BEGIN
5 :NEW.CardNo := borrower_seq.NEXTVAL;
6 END;
7 /
```

Trigger created.

```
SQL> CREATE TABLE table_book_copies (
2 CopiesID INT PRIMARY KEY NOT NULL,
3 BookID INT NOT NULL,
```

```
4  BranchID INT NOT NULL,  
5  No_Of_Copies INT NOT NULL  
6 );
```

Table created.

```
SQL> CREATE SEQUENCE author_seq  
2  START WITH 1  
3  INCREMENT BY 1;
```

Sequence created.

```
SQL> CREATE TABLE table_book_authors (  
2  AuthorID INT PRIMARY KEY NOT NULL,  
3  BookID INT NOT NULL,  
4  AuthorName VARCHAR(50) NOT NULL,  
5  CONSTRAINT fk_book_id3 FOREIGN KEY (BookID) REFERENCES table_book(BookID) ON DELETE  
CASCADE  
6 );
```

Table created.

```
SQL> CREATE SEQUENCE isbn_seq  
2  START WITH 1000000000000  
3  INCREMENT BY 1;
```

Sequence created.

```
SQL> CREATE TABLE book (  
2  isbn CHAR(13) PRIMARY KEY NOT NULL,  
3  title VARCHAR2(80) NOT NULL,
```

```
4  author VARCHAR2(80) NOT NULL,  
5  category VARCHAR2(80) NOT NULL,  
6  price NUMBER(4) NOT NULL CHECK (price >= 0),  
7  copies NUMBER(10) NOT NULL CHECK (copies >= 0)  
8 );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER isbn_trigger  
2 BEFORE INSERT ON book  
3 FOR EACH ROW  
4 BEGIN  
5   :NEW.isbn := isbn_seq.NEXTVAL;  
6 END;  
7 /
```

Trigger created.

```
SQL> CREATE TABLE book_issue (  
2  issue_id NUMBER PRIMARY KEY NOT NULL,  
3  member VARCHAR2(20) NOT NULL,  
4  book_isbn CHAR(13) NOT NULL,  
5  due_date DATE NOT NULL,  
6  last_reminded DATE DEFAULT NULL  
7 );
```

Table created.

```
SQL> ALTER TABLE book_issue  
2 ADD (price NUMBER(4) NOT NULL CHECK (price >= 0));
```

Table altered.

```

SQL> CREATE OR REPLACE TRIGGER issue_book
2 BEFORE INSERT ON book_issue
3 FOR EACH ROW
4 DECLARE
5     v_price NUMBER;
6 BEGIN
7     :NEW.due_date := SYSDATE + 20;
8
9     -- Assuming PRICE is already in the book_issue table
10    v_price := :NEW.price;
11
12    -- If you need to deduct a balance or manage other logic,
13    -- you may need to implement that logic here or remove it if unnecessary.
14
15    -- Assuming there is a field to track copies in book_issue, update copies if needed.
16    -- You might want to adjust this logic based on your requirements.
17 END;
18 /

```

Trigger created.

```

SQL> CREATE TABLE librarian (
2     id NUMBER(11) NOT NULL,
3     username VARCHAR2(20) NOT NULL,
4     password CHAR(40) NOT NULL,
5     CONSTRAINT pk_librarian PRIMARY KEY (id)
6 );

```

Table created.

```

SQL> CREATE SEQUENCE member_seq

```

```
2  START WITH 1
3  INCREMENT BY 1;
```

Sequence created.

```
SQL> CREATE TABLE member (
2  id NUMBER(11) NOT NULL,
3  username VARCHAR2(20) NOT NULL,
4  password CHAR(40) NOT NULL,
5  name VARCHAR2(80) NOT NULL,
6  email VARCHAR2(80) NOT NULL,
7  balance NUMBER(4) NOT NULL,
8  CONSTRAINT pk_member PRIMARY KEY (id)
9 );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER member_id_trigger
2  BEFORE INSERT ON member
3  FOR EACH ROW
4  BEGIN
5  :NEW.id := member_seq.NEXTVAL;
6  END;
7  /
```

Trigger created.

```
SQL> CREATE TABLE pending_book_requests (
2  request_id NUMBER(11) NOT NULL,
3  member VARCHAR2(20) NOT NULL,
4  book_isbn CHAR(13) NOT NULL,
```

```
5  time TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
6  PRIMARY KEY (request_id) -- Adding a primary key constraint
7 );
```

Table created.

```
SQL> CREATE TABLE pending_registrations (
2  username VARCHAR2(30) NOT NULL,
3  password CHAR(20) NOT NULL,
4  name VARCHAR2(40) NOT NULL,
5  email VARCHAR2(80) NOT NULL, -- Adjusted to allow for longer email addresses
6  balance NUMBER(10),          -- Using NUMBER instead of INT
7  time TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
8  PRIMARY KEY (username)      -- Adding a primary key constraint on username
9 );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER add_member
2  AFTER INSERT ON member
3  FOR EACH ROW
4  BEGIN
5  DELETE FROM pending_registrations
6  WHERE username = :NEW.username;
7  END;
8  /
```

Trigger created.

```
SQL> CREATE OR REPLACE TRIGGER remove_member
2  AFTER DELETE ON member
```



```
3 FOR EACH ROW
4 BEGIN
5     DELETE FROM pending_book_requests
6     WHERE member = :OLD.username;
7 END;
8 /
```

Trigger created.