**Name of the Problem 01:**

Using Microsoft SQL server, construct a database named“LibraryAssignment” and using this database create three tables satisfying the properties as listed and then populated them with records.

Table: **Books**

|  |  |
| --- | --- |
| **Field** | **Requirement** |
| Book ID | 1. Primary key. 2. Only values starting with “B” and then having 5 digits are allowed. |
| Title | Can not be left blank. |
| Author | Can not be left blank. |
| Publisher | Can be left blank. |
| Category | Must be restricted to “Tech”, “Busi” or “Arts”. |
| Price | Can not be negative. |
| No. book in hand | Can not be negative. |

# Table: Members

|  |  |
| --- | --- |
| **Field** | **Requirments** |
| Member ID | 1. Primary key. 2. Only values starting with either “S” or “T” and then having 5 digits are allowed. |
| First Name | Can not be left blank |
| Last name | Can not be left blank |
| Address | Can be left blank |
| No. of currently issued books | Can not be negative |
| Issue ceiling | Can not be negative |

Table: Transaction

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Transaction ID | 1. Primary key.   2. Only values starting with “T” and then having 5 digits are allowed. |
| Transaction date | Default is current date. |
| Transaction type | Must be restricted to either “I” or “R”. |
| Member ID | Foreign key with appropriate reference. |
| Book ID | Foreign key with appropriate reference. |

**Source Code:**

createdatabase LibraryAssignment

use LibraryAssignment

create table Books(

BookId char(6)primary key check(BookId like('B[0-9][0-9][0-9][0-9][0-9]')),

Title varchar(50)not null,

Author varchar(50)not null,

Publisher varchar(50),

Category char(4)check((Category like'Tech')or(Category like'Busi')or(Category like'Arts')),

Price money check(Price>=0),

NoOfBookInHand int check(NoOfBookInHand>=0)

)

droptable Books

select\*from Books

insert into Books values('B11111','Thinking in JAVA','Paul Deitel','Welly','Tech', 300, 2)

insert into Books values('B22245','Programming in C','Herbert Schild','Welly','Tech', 150, 4)

insert into Books values('B45678','Microprocessors Detais','Berry B Berry','Pearson','Tech', 350, 3)

insert into Books values('B89754','Probability in Statistics','Norendro mallik','','Busi', 100, 4)

insert into Books values('B85246','Engineering Cad', 'Someone','', 'Arts', 50, 5)

createtable Members(

MemberID char(6)primary key check((MemberID like('S[0-9][0-9][0-9][0-9][0-9]'))or(MemberID like('T[0-9][0-9][0-9][0-9][0-9]'))),

FirstName varchar(50)not null,

LastName varchar(50)not null,

Address varchar(50),

NoOfCurrentlyIssuedBooks int check(NoOfCurrentlyIssuedBooks>=0),

IssueCeiling int check(IssueCeiling>=0)

)

drop table Members

select\*from Members

insert into Members values('T12345','Tarun','Debnath','', 5,2)

insert into Members values('S12345','someone','Other Cast','Pabna', 65, 5)

insert into Members values('T54123','anyone','last name','kaliganj', 50, 2)

insert into Members values('T98745','Kamal','Hasan','Australia', 10,4)

insert into Members values('S74589','Rofiq','Uddin','Dhaka', 4, 1)

create table Transactions(

TransactionID char(6)primary key check(TransactionID like('T[0-9][0-9][0-9][0-9][0-9]')),

TransactionDate smalldatetime default getdate(),

TransactionType char(1)check((TransactionType like('I'))or(TransactionType like('R'))),

MemberId char(6)foreign key references Members(MemberID),

BookId char(6)foreign key references Books(BookId)

)

drop table Transactions

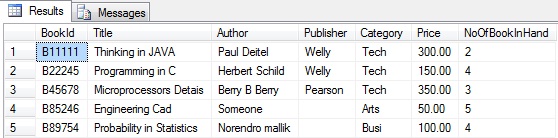
select\*from Transactions

insert into Transactions values('T12345','','I','T12345','B11111')

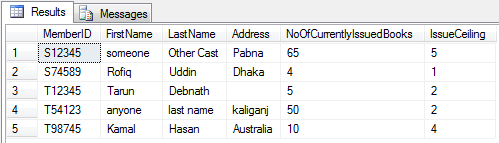
insert into Transactions values('T12414','','R','S12345','B45678')

**Output**

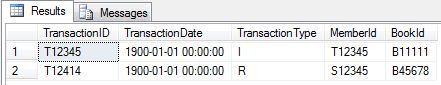
Books table:



Members table:



Transactions table:



**Name of the Problem 02:**

Using Microsoft SQL server, construct a database named ‘‘Shop’’ and using this database create three tables satisfying the properties as listed below and then populate them with records.

**Table: Product**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Product ID | 1. Primary key.  2. Only values starting with an alphabet and then having 5 digits are allowed. |
| Product Name | Can not be left blank |
| Manufacturer | Can be left blank |
| Origin | Must be restricted to “Local” or “Foreign” |
| Price | Can not be negative |
| Quantity in hand | Can not be negative |

**Table: Customers**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Customer ID | 1. Primary key.   2. Only values starting with “C” and then having 6 digits are allowed. |
| First Name | Can not be left blank |
| Last Name | Can be left blank |
| City | Default is “Dhaka”. |
| Total sale quantity | Can not be negative |

**Table : Transactions**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Transaction ID | 1. Primary key.  2. Only values starting with “T” and then having 5 digits are allowed. |
| Transaction date | Default is current date. |
| Customer ID | Foreign key with appropriate reference. |
| Product ID | Foreign key with appropriate reference. |

**Source Code:**

createdatabase Shop

use Shop

createtable Product(

ProductID char(6)primarykeycheck(ProductID like('[A-Z][0-9][0-9][0-9][0-9][0-9]')),

ProductName varchar(50)notnull,

Manufacturer varchar(50),

Origin varchar(7)check((Origin like'Local')or(Origin like'Foreign')),

Price moneycheck(Price>=0),

QuantityInHand intcheck(QuantityInHand>=0)

)

droptable Product

select\*from Product

insertinto Product values('C12345','Gigabyte motherbord','','Local',300,5)

insertinto Product values('m12346','Asus Moniter','Asus','Foreign',200,8)

insertinto Product values('P54321','3.07 GHz','Intel','Foreign', 100, 10)

insertinto Product values('K54321','PS-2','Jetway','Local', 6, 100)

insertinto Product values('M98560','USB','Logitec','Foreign', 3, 100)

createtable Customers(

CustomerID char(7)primarykeycheck(CustomerID like('C[0-9][0-9][0-9][0-9][0-9][0-9]')),

FirstName varchar(50)notnull,

LastName varchar(50),

City varchar(50)default'Dhaka',

TotalSaleQuantity intcheck(TotalSaleQuantity>=0)

)

droptable Customers

select\*from Customers

insertinto Customers values('c123456','Tarun','Debnath','', 2)

insertinto Customers values('c458795','Raho','cast','Rajshahi', 45)

insertinto Customers values('c128556','Karim','','Khulna', 5)

insertinto Customers values('c593456','Rasedul','Islam','Dhaka', 3)

insertinto Customers values('c973456','Musfiq','Rahim','Dhaka', 0)

createtable Transactions(

TransactionID char(6)primarykeycheck((TransactionID like('T[0-9][0-9][0-9][0-9][0-9]'))),

TransactionDate smallDateTimedefaultgetdate(),

CustomerID char(7)foreignkeyreferences Customers(CustomerID),

ProductID char(6)foreignkeyreferences Product(ProductID)

)

droptable Transactions

droptable Customers

droptable Product

select\*from Transactions

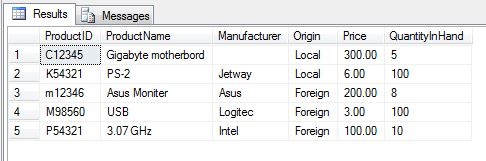
insertinto Transactions values('t89764','','c128556','m12346')

insertinto Transactions values('t12764','','c123456','M98560')

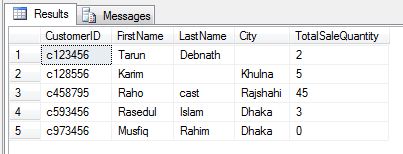
insertinto Transactions values('t12475','','c458795','P54321')

**Output:**

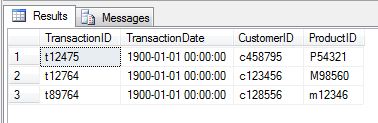
Product table:



Customers table:



Transactions table:

****

**Name of the Problem 03:**

Using Microsoft SQL server, construct a database named ‘’ToutorialService’’ and using this database create three tables satisfying the properties as listed below and then populate them with records.

**Table: Participants**

|  |  |
| --- | --- |
| **Field** | **Reqirements** |
| Participant ID | 1. Primary key.  2. Only values starting with “P” and then having 5 digits are allowed. |
| First Name | Can not be left blank |
| Last Name | Can be left blank |
| Address | Can not be left blank |
| Age | Can not be negative |
| Total courses registered | Can not be negative |

**Table: Courses**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Course ID | 1. Primary key.  2. Only values starting with either “T” or “N” and then having 3 digits are allowed. |
| Title | Can not be left blank |
| Type | Must be restricted to “Tech” or “Non-Tech” |
| Total Participant | Can not be negative |
| Fee |  |

**Source Code:**

createdatabase ToutorialService

use ToutorialService

createtable Participants(

ParticipantID char(6)primarykeycheck(ParticipantID like'P[0-9][0-9][0-9][0-9][0-9]'),

FirstName varchar(50)notnull,

LastName varchar(50),

Address varchar(50)notnull,

Age intcheck(Age>=0),

TotalCoursesRegistered intcheck(TotalCoursesRegistered>=0)

)

droptable Participants

select\*from Participants

insertinto Participants values('P12345','Tarun','Debnath','kaliganj', 24, 6)

insertinto Participants values('P13345','Rahim','Talukder','kaliganj', 18, 2)

insertinto Participants values('P14345','Karim','','Pabna', 30, 5)

insertinto Participants values('P15345','Rafi','Islam','jessore', 24, 6)

createtable Courses(

CourseID char(4)check((CourseID like'T[0-9][0-9][0-9]')or(CourseID like'N[0-9][0-9][0-9]')),

Title varchar(50)notnull,

Types varchar(8)check((Types like'Tech')or(Types like'Non-Tech')),

TotalParticipant intcheck(TotalParticipant>=0),

Fee moneycheck(Fee>=0)

)

droptable Courses

select\*from Courses

insertinto Courses values('T123','Programming in C','Tech', 12, 0)

insertinto Courses values('T113','Programming in Java','Tech', 12, 5000)

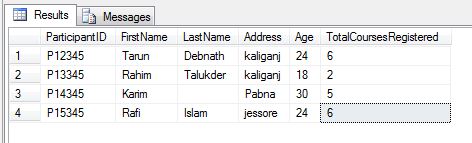
insertinto Courses values('T124','Android programming','Tech', 12, 10000)

insertinto Courses values('T125','Engineering Mathematics','Non-Tech', 12, 0)

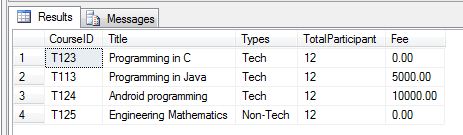
select CourseID, Title, TotalParticipant, Fee from Courses where Types='Tech'

**Output:**

Participants table:



Courses table:



**Name of the Problem 04:**

Write a stored procedure that prints out book categories, total number of books available and the average price of books under each category in the following format.(Database will be collected from problem 1)

**Source Code:**

-- ================================================

SETANSI\_NULLSON

GO

SETQUOTED\_IDENTIFIERON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

createprocedure q3

as

begin

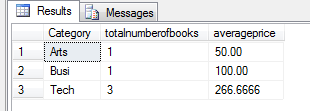
select Category ,count(Price)as totalnumberofbooks,AVG(Price)as averageprice from Books groupby Category

end

exec q3

dropprocedure q3

**Output:**



**Name of the Problem 05:**

Write a stored procedure that accepts a number and prints book categories and the average price of books of those categories only if the average price exceeds the given number(Database will be collected from problem 1).

**Source Code:**

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SETANSI\_NULLSON

GO

SETQUOTED\_IDENTIFIERON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATEPROCEDURE q1

@n int

AS

BEGIN

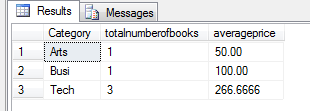
select Category ,count(Price)as totalnumberofbooks,AVG(Price)as averageprice from Books groupby Category havingavg(Price)>= @n

end

exec q1 50

dropprocedure q1

**Output:**



**Name of the Problem 06:**

Write a stored procedure that accepts a number and starts increasing the issue ceiling for each member by 1 until the average issue ceiling goes beyond the accepted number (Database will be collected from problem 1).

**Source Code:**

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SETANSI\_NULLSON

GO

SETQUOTED\_IDENTIFIERON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATEPROCEDURE q4

@n int

AS

while( @n >(selectavg(IssueCeiling)from Members))

begin

update Members set IssueCeiling = IssueCeiling + 1

end

--drop procedure p2

---select avg(Issue\_Ceiling) from Members

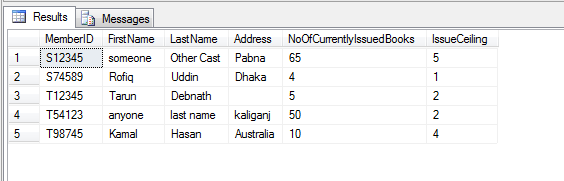
exec q4 25

select \*from Members

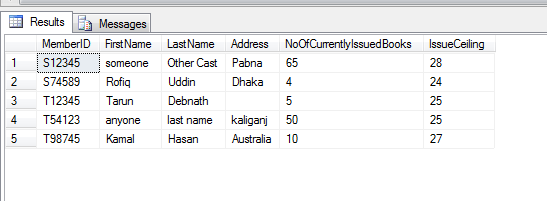
dropprocedure q4

**Output:**

**Before the change:**



**After the change:**

****

**Name of the Problem 07:**

**(i)** Using Microsoft SQL server, construct a database named ‘Shop’ and using this database perform the following tasks.

(ii) Create three tables satisfying the properties as listed below and then populate them with records.

**Table: Product**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Product ID | 1. Primary key.  2. Only values starting with an alphabet and then having 5 digits are allowed. |
| Product Name | Can not be left blank |
| Manufacturer | Can be left blank |
| Origin | Must be restricted to “Local” or “Foreign” |
| Price | Can not be negative |
| Quantity in hand | Can not be negative |

**Table: Customers**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Customer ID | 1. Primary key.   2. Only values starting with “C” and then having 6 digits are allowed. |
| First Name | Can not be left blank |
| Last Name | Can be left blank |
| City | Default is “Dhaka”. |
| Total sale quantity | Can not be negative |

**Table: Transactions**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Transaction ID | 1. Primary key.  2. Only values starting with “T” and then having 5 digits are allowed. |
| Transaction date | Default is current date. |
| Customer ID | Foreign key with appropriate reference. |
| Product ID | Foreign key with appropriate reference. |

**Trigger**

1.Write a trigger that performs necessary updating of the Customers table and the Products table whenever a transaction takes place.

**Source Code:**

dropdatabase shop

createdatabase shop

use shop

createtable Product(

product\_name varchar(15),

product\_id varchar(5)primarykeycheck(product\_id like('P[0-9][0-9][0-9][0-9]')),

total\_product\_in int

)

droptable Product;

insertinto Product(product\_name,product\_id,total\_product\_in)values('book','P1111',12);

insertinto Product(product\_name,product\_id,total\_product\_in)values('pen','P1112',12);

insertinto Product(product\_name,product\_id,total\_product\_in)values('glass','P1113',12);

select\*from Product;

createtable Customer(

customer\_name varchar(15),

customer\_id varchar(6)primarykeycheck(customer\_id like('C[0-9][0-9][0-9][0-9][0-9]')),

total\_prodect int

)

insertinto Customer(customer\_name,customer\_id,total\_prodect)values('a','C11111',0);

insertinto Customer(customer\_name,customer\_id,total\_prodect)values('b','C11112',0);

insertinto Customer(customer\_name,customer\_id,total\_prodect)values('c','C11113',0);

select\*from Customer;

createtable Trans(

Transaction\_Id char(6)primarykeycheck(Transaction\_Id like('T[0-9][0-9][0-9][0-9][0-9]')),

Transaction\_Date smalldatetimedefaultgetdate(),

customer\_id varchar(6)foreignkeyreferences Customer ( customer\_id),

product\_id varchar(5)foreignkeyreferences Product (product\_id)

)

droptable Trans;

insertinto Trans(Transaction\_Id,customer\_id,product\_id)values('T11111','C11111','P1111');

insertinto Trans(Transaction\_Id,customer\_id,product\_id)values('T11112','C11112','P1111');

insertinto Trans(Transaction\_Id,customer\_id,product\_id)values('T11113','C11112','P1112');

select\*from Trans;

droptable Customer

--use Library

-----------

CREATETRIGGER tr

ON Trans

forinsert

AS

declare @ci varchar(6)

declare @pi varchar(5)

BEGIN

select @ci=customer\_id from Trans

select @pi=product\_id from Trans

update Product set total\_product\_in=total\_product\_in-1 where product\_id=@pi

update Customer set total\_prodect=total\_prodect+1 where customer\_id=@ci

END

droptrigger tr

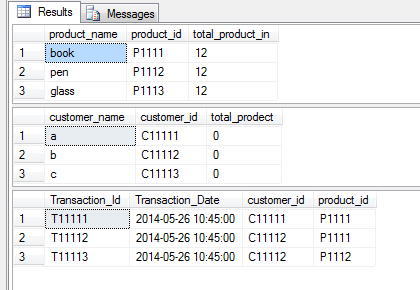
select\*from Product

select\*from Customer

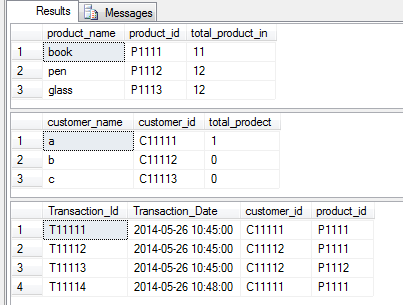
select\*from Trans

insertinto Trans(Transaction\_Id,customer\_id,product\_id)values('T11114','C11111','P1111');

**Output:**



**After Trigger:**



**Name of the Problem 08:**

**(i)** Using Microsoft SQL server, construct a database named ‘ToutorialService’ and using this database perform the following tasks.

(ii) Create three tables satisfying the properties as listed below and then populate them with records.

**Table: Participants**

|  |  |
| --- | --- |
| **Field** | **Reqirements** |
| Participant ID | 1. Primary key.  2. Only values starting with “P” and then having 5 digits are allowed. |
| First Name | Can not be left blank |
| Last Name | Can be left blank |
| Address | Can not be left blank |
| Age | Can not be negative |
| Total courses registered | Can not be negative |

**Table: Courses**

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Course ID | 1. Primary key.  2. Only values starting with either “T” or “N” and then having 3 digits are allowed. |
| Title | Can not be left blank |
| Type | Must be restricted to “Tech” or “Non-Tech” |
| Total Participant | Can not be negative |
| Fee |  |

**Table: Registration**

|  |  |
| --- | --- |
| **Field** | **Requirement** |
| Registration ID | 1. Primary key.  2. Only values starting with “R” and then having 5 digits are allowed. |
| Registration date | Default in current date |
| Course ID | Foreign key with appropriate reference. |
| Participant ID | Foreign key with appropriate reference. |

**Trigger**

**1**.Write a trigger hat performs necessary updating of the Participants table and the Courses table whenever a registration takes place.

**Source Code:**

dropdatabase ToutorialService

createdatabase ToutorialService

use ToutorialService

createtable Participants (

Participant\_ID varchar(6)primarykeycheck(Participant\_ID like('P[0-9][0-9][0-9][0-9][0-9]')or Participant\_ID like('K[0-9][0-9][0-9][0-9][0-9]')),

First\_Name varchar(12)notnull,

Last\_Name varchar(6)null,

Address varchar(100)notnull,

Age intcheck(Age>=0)default 10,

Total\_courses\_registered intcheck(Total\_courses\_registered>=0)

)

select\*from Participants

insertinto Participants (Participant\_ID,First\_Name,Last\_Name,Address,Total\_courses\_registered)values('k12345','s','s','a',5)

insertinto Participants values('k11111','k','e','m',22,4)

droptable Participants

createtable Courses(

Course\_id varchar(4)primarykeycheck( Course\_id like('T[0-9][0-9][0-9]')or Course\_id like('N[0-9][0-9][0-9]')),

Title varchar(50)notnull,

Types varchar(8)check((Types like'Tech')or(Types like'Non-Tech')),

TotalParticipant intcheck(TotalParticipant>=0),

Fee intcheck(Fee>=0)

)

select\*from Courses

insertinto Courses values('T111','mama','Tech',6,100)

insertinto Courses values('T112','mami','Non-Tech',24,10)

createtable Registration(

Registration\_ID varchar(6)primarykeycheck( Registration\_ID like('R[0-9][0-9][0-9][0-9][0-9]')),

Registration\_date smalldatetimedefaultgetdate(),

Course\_id varchar(4)foreignkeyreferences Courses(Course\_id),

Participant\_ID varchar(6)foreignkeyreferences Participants(Participant\_ID)

)

select\*from Registration

insertinto Registration values('R11111','','T111','k12345')

insertinto Registration values('R11112','','T112','k11111')

droptable Registration

CREATETRIGGER tr

ON Registration

forinsert

AS

declare @ci varchar(4)

declare @pi varchar(6)

BEGIN

select @ci=Course\_id from Registration

select @pi=Participant\_ID from Registration

update Courses set TotalParticipant=TotalParticipant-1 where Course\_id=@ci

update Participants set Total\_courses\_registered=Total\_courses\_registered+1 where Participant\_ID=@pi

END

droptrigger tr

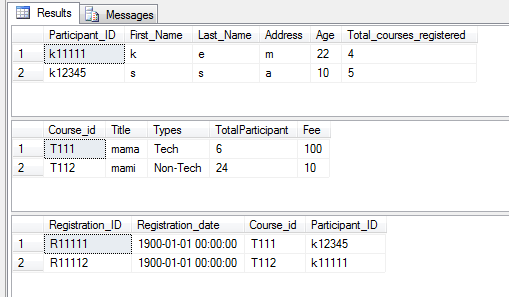
select\*from Participants

select\*from Courses

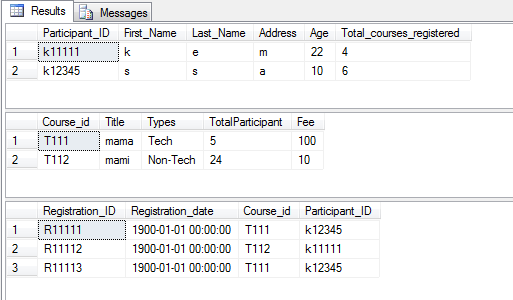
select\*from Registration

insertinto Registration values('R11113','','T111','k12345')

**Output:**



**2.After trigger:**



**Name of the Problem 09:**

(i) Using Microsoft SQL server, construct a database named ‘Library’ and using this database perform the following tasks.

1. Create three tables satisfying the properties as listed below and then populate them with records.

Table: **Books**

|  |  |
| --- | --- |
| **Field** | **Requirement** |
| Book ID | 1. Primary key. 2. Only values starting with “B” and then having 5 digits are allowed. |
| Title | Can not be left blank. |
| Author | Can not be left blank. |
| Publisher | Can be left blank. |
| Category | Must be restricted to “Tech”, “Busi” or “Arts”. |
| Price | Can not be negative. |
| No. book in hand | Can not be negative. |

# Table: Members

|  |  |
| --- | --- |
| **Field** | **Requirments** |
| Member ID | 1. Primary key. 2. Only values starting with either “S” or “T” and then having 5 digits are allowed. |
| First Name | Can not be left blank |
| Last name | Can not be left blank |
| Address | Can be left blank |
| No. of currently issued books | Can not be negative |
| Issue ceiling | Can not be negative |

Table: Transaction

|  |  |
| --- | --- |
| **Field** | **Requirements** |
| Transaction ID | 1. Primary key.   2. Only values starting with “T” and then having 5 digits are allowed. |
| Transaction date | Default is current date. |
| Transaction type | Must be restricted to either “I” or “R”. |
| Member ID | Foreign key with appropriate reference. |
| Book ID | Foreign key with appropriate reference. |

**Trigger**

3. Write trigger that performs necessary updating of the Members table whenever a transaction takes place.

**Source Code:**

createdatabase Library

use Library

createtable Books(

BookId char(6)primarykeycheck(BookId like('B[0-9][0-9][0-9][0-9][0-9]')),

Title varchar(50)notnull,

Author varchar(50)notnull,

Publisher varchar(50),

Category char(4)check((Category like'Tech')or(Category like'Busi')or(Category like'Arts')),

Price moneycheck(Price>=0),

NoOfBookInHand intcheck(NoOfBookInHand>=0)

)

droptable Books

insertinto Books values('B11111','Thinking in JAVA','Paul Deitel','Welly','Tech', 300, 2)

insertinto Books values('B22245','Programming in C','Herbert Schild','Welly','Tech', 150, 4)

insertinto Books values('B45678','Microprocessors Detais','Berry B Berry','Pearson','Tech', 350, 3)

insertinto Books values('B89754','Probability in Statistics','Norendro mallik','','Busi', 100, 4)

insertinto Books values('B85246','Engineering Cad', 'Someone','', 'Arts', 50, 5)

createtable Members(

MemberID char(6)primarykeycheck((MemberID like('S[0-9][0-9][0-9][0-9][0-9]'))or(MemberID like('T[0-9][0-9][0-9][0-9][0-9]'))),

FirstName varchar(50)notnull,

LastName varchar(50)notnull,

Address varchar(50),

NoOfCurrentlyIssuedBooks intcheck(NoOfCurrentlyIssuedBooks>=0),

IssueCeiling intcheck(IssueCeiling>=0)

)

droptable Members

insertinto Members values('T12345','Tarun','Debnath','', 5,2)

insertinto Members values('S12345','someone','Other Cast','Pabna', 65, 5)

insertinto Members values('T54123','anyone','last name','kaliganj', 50, 2)

insertinto Members values('T98745','Kamal','Hasan','Australia', 10,4)

insertinto Members values('S74589','Rofiq','Uddin','Dhaka', 4, 1)

createtable Transactions(

TransactionID char(6)primarykeycheck(TransactionID like('T[0-9][0-9][0-9][0-9][0-9]')),

TransactionDate smalldatetimedefaultgetdate(),

TransactionType char(1)check((TransactionType like('I'))or(TransactionType like('R'))),

MemberId char(6)foreignkeyreferences Members(MemberID),

BookId char(6)foreignkeyreferences Books(BookId)

)

droptable Transactions

insertinto Transactions values('T12345','','I','T12345','B11111')

insertinto Transactions values('T12414','','R','S12345','B45678')

select\*from Books

select\*from Members

select\*from Transactions

droptable Members

droptable Books

CREATETRIGGER tr

ON Transactions

forinsert

AS

declare @mi varchar(6)

BEGIN

select @mi=MemberID from Transactions

update Members set IssueCeiling=IssueCeiling+1 where MemberID=@mi

END

droptrigger tr

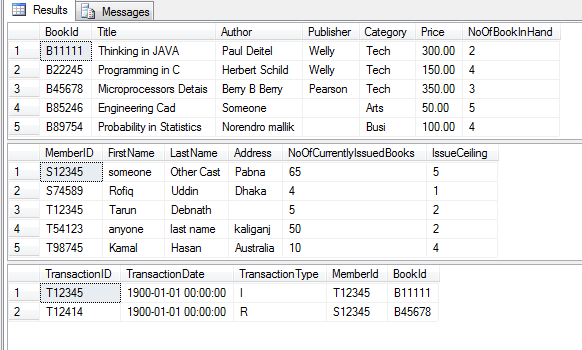
q

select\*from Members

select\*from Transactions

insertinto Transactions values('T12415','','R','S12345','B45678')

**Output:**



**After trigger:**

