# PHASE 3: Procedures and Triggers

**Project: Library management system**

**Submitted to: Ms. Rameen Rana**

**Submitted by:**

**Huzaifa Hussnain \_ L164345**

**Mirsub Fazlani \_ l164382**

**Hammad Ikhlaq \_ L164281**

**Stored Procedures:**

-- this procedure will get the books issued by the member whose id is given --

create procedure booksBorrowed @id int

as begin

select bi.Book\_ISBN as ISBN , b.Title,bi.IssueDate,bi.DueDate

from BooksIssued bi join book b on bi.Book\_ISBN=b.ISBN

where bi.Member\_Id= @id

end

-- procedure to search a book by title---

create procedure searchByTitle @title varchar(30)

as begin

select \* from book where title = @title

end

-- procedure to search a book by author---

create procedure searchByAuthor @author varchar(30)

as begin

select \* from book where author = @author

end

-- procedure to search a book by ISBN---

create procedure searchByISBN @isbn char(13)

as begin

select \* from book where ISBN = @isbn

end

-- procedure to borrow a book---

create procedure issueBook @id int, @isbn char(13)

as begin

insert into BooksIssued

values (@id,@isbn,'','') -- trigger will automatically handle the last two values --

end

-- procedure to queue a book---

create procedure queueBook @id int, @isbn char(13)

as begin

insert into BookQueued

values (@id,@isbn,'') -- trigger will automatically handle the last value --

end

-- procedure to place a request for a book---

create procedure requestBook @id int, @isbn char(13),@title varchar(30),@author varchar(30)

as begin

insert into BookRequest

values (@id,@isbn,@title,@author)

end

**Triggers:**

-- trigger on BookRequest--

alter trigger insertionInBookRequests on BookRequest

instead of insert

as begin

-- check if the person trying to place a request is not ban--

declare @id int

select @id=inserted.Member\_id -- id of the member who placed request--

from inserted

declare @banStatus varchar(12);

select @banStatus=[status]

from Member

where id = @id

if(@banStatus='ban' or @banStatus='Ban' or @banStatus='BAN')

begin

Print 'MEMEBR IS BANNED FROM USING LIBRARY SERVICES'

end

else -- now perform other checks --

begin

declare @isbn char(13)

select @isbn= inserted.isbn

from inserted

if exists ( select isbn from book where ISBN= @isbn)

begin

print 'LIBRARY ALREADY HAS THIS BOOK.SO INSTEAD OF PLACING REQUEST QUEUE THE BOOK'

end

else

begin

if exists (select isbn from BookRequest where Member\_Id=@id and ISBN =@isbn)

begin

print 'REQUEST FOR THE SAME BOOK BY THE SAME MEMBER ALREADY EXISTS'

end

else

begin

declare @title varchar(30), @author varchar(30)

select @title=inserted.Booktitle,@author=inserted.Author -- selecting value for title and author--

from inserted

insert into BookRequest

values (@id,@isbn,@title,@author)

print 'BOOK REQUEST PLACED SUCCESSFULLY.'

end

end

end

end

-- trigger on booksIssued to perform checks while inerting values in it--

create trigger insertionInBooksIssued on BooksIssued

instead of insert

as begin

declare @id int

select @id=inserted.Member\_id

from inserted

-- checking for whether a person is ban or not--

declare @banStatus varchar(12);

select @banStatus=[status]

from Member

where id = @id

if(@banStatus='ban' or @banStatus='Ban' or @banStatus='BAN')

begin

Print 'MEMEBR IS BANNED FROM USING LIBRARY SERVICES'

end

else

begin -- person is not ban --

declare @isbn char(13)

select @isbn=inserted.Book\_ISBN

from inserted

--- if the same request by same person does not exit--

if not exists ( select Book\_ISBN from BooksIssued where @id=Member\_Id and @isbn = Book\_ISBN )

begin -- begin if not exist --

--- checking if the book exists in library ---

if exists ( select ISBN from book where ISBN=@isbn) -- book exists --

begin

-- CHECKING IF THE BOOK HAS SOME COPIES AVAILABLE IN LIBRARY---

declare @copies int

select @copies=CopiesAvailable

from book

where ISBN=@isbn

if(@copies>0) -- copies of book available --

begin

-- now an entry will be added to booksIssued--

declare @rank int,@date date

select @rank=RankNum

from Member

where ID=@id

if(@rank=1) -- member is a teacher--

begin

insert into BooksIssued

values(@id,@isbn,getdate(),getdate()+60) -- teacher can issue a book for 60 days--

end

else -- member is student --

begin

insert into BooksIssued

values(@id,@isbn,getdate(),getdate()+15) -- student can issue a book for 15 days--

end

-- the number of copies of the book issued is to be decremenetd by 1--

update book

set CopiesAvailable = CopiesAvailable-1

where ISBN= @isbn

end --if copies>0 --

else

begin

print 'THERE IS NO COPY THE BOOK AVAILABLE YET! YOU CAN "QUEUE" THE BOOK'

end

end -- if exists --

else

begin -- book does not exists--

print 'LIBRARY DOES NOT HAS THE BOOK YOU WANTED TO ISSUE.YOU CAN PLACE A REQUEST FOR THE BOOK'

end

end -- end if not exist --

else

begin -- same request already exists --

print 'YOU HAVE ALREADY ISSUED THE SAME BOOK AND NOT RETURNED YET!'

end

end -- end else of ban --

end

-- trigger on BookQueued--

alter trigger insertionInBookQueued on BookQueued

instead of insert

as begin

-- check if the person trying to place a request is not ban--

declare @id int

select @id=inserted.Member\_id -- this will get the of the member who placed request--

from inserted

declare @banStatus varchar(12);

select @banStatus=[status]

from Member

where id = @id

if(@banStatus='ban' or @banStatus='Ban' or @banStatus='BAN')

begin

Print 'MEMEBR IS BANNED FROM USING LIBRARY SERVICES'

end

else -- now perform other checks --

begin

declare @isbn char(13)

select @isbn= inserted.isbn

from inserted

if not exists ( select isbn from book where ISBN= @isbn)

begin

print 'LIBRARY DOES NOT HAS THIS BOOK.SO INSTEAD OF QUEUING THE BOOK PLACE A REQUEST FOR THIS BOOK'

end

else

begin

if exists (select isbn from BookQueued where Member\_Id=@id and ISBN =@isbn)

begin

print 'YOU HAVE ALREADY QUEUED THE SAME BOOK'

end

else

begin

insert into BookQueued

values (@id,@isbn,getdate())

print 'BOOK QUEUED SUCCESSFULLY.'

end

end

end

end