

DEBRE BERHA UNIVERSITY

College of Computing

Department of Computer Science

Case Study Assignment for 2nd Year Computer Science Students

Course Title: Fundamentals of Database Systems Course Code: CoSc2041

GROUP TWO

|  |  |  |
| --- | --- | --- |
| ID | FIRST NAME | LAST NAME |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Submission date: January 25/05/2017 E.C

Presentation date: 26/05/2017 E.C

General Instractions.

* Identify Entities,attributes and relationships.
* Draw entity relationship diagrams.
* Map the entity relationship diagram to relational schema.
* Implement the relational schema using SQL server.

# DBU Tech Library Case Study

The DBU Tech Library (DTL) has approximately 16,000 members, 100,000 titles, and 250,000 volumes (an average of 2.5 copies per book). About 10% of the volumes are out on loan at any one time. The librarians ensure that the books that members want to borrow are available when the members want to borrow them. Also, the librarians must know how many copies of each book are in the library or out on loan at any given time. A catalog of books is available online that lists books by author, title, and subject area. For each title in the library, a book description is kept in the catalog; the description ranges from one sentence to several pages. The reference librarians want to be able to access this description when members request information about a book. Library staff includes chief librarian, departmental associate librarians, reference librarians, check-out staff, and library assistants.

Books can be checked out for 21 days. Members are allowed to have only five books out at a time. Members usually return books within three to four weeks. Most members know that they have one week of grace before a notice is sent to them, so they try to return books before the grace period ends. About 5% of the members have to be sent reminders to return books. Most overdue books are returned within a month of the due date. Approximately 5% of the overdue books are either kept or never returned. The most active members of the library are defined as those who borrow books at least ten times during the year. The top 1% of membership does 15% of the borrowing, and the top 10% of the membership does 40% of the borrowing. About 20% of the members are totally inactive in that they are members who never borrow.

To become a member of the library, applicants fill out a form including their SSN, campus and home mailing addresses, and phone numbers. The librarians issue a numbered, machine-readable card with the member’s photo on it. This card is good for four years. A month before a card expires, a notice is sent to a member for renewal. Lecturers at the institute are considered automatic members. When a new faculty member joins the institute, his or her information is pulled from the employee records and a library card is mailed to his or her campus address. Lecturers are allowed to check out books for three-month intervals and have a two-week grace period. Renewal notices to Lecturers are sent to their campus address.

The library does not lend some books, such as reference books, rare books, and maps. The librarians must differentiate between books that can be lent and those that cannot be lent. In addition, the librarians have a list of some books they are interested in acquiring but cannot obtain, such as rare or out of-print books and books that were lost or destroyed but have not been replaced. The librarians must have a system that keeps track of books that cannot be lent as well as books that they are interested in acquiring. Some books may have the same title; therefore, the title cannot be used as a means of identification. Every book is identified by its International Standard Book Number (ISBN), a unique international code assigned to all books. Two books with the same title can have different ISBNs if they are in different languages or have different bindings (hardcover or softcover). Editions of the same book have different ISBNs.

The proposed database system must be designed to keep track of the members, the books, the catalog, and the borrowing activity.

**Identify Entities,attributes and relationships**.

***Entities and Attributes***

***1, Member***:

* Member ID (Primary Key)
* SSN
* First Name
* Last Name
* Campus Address
* Home Address
* Phone Number
* Membership Type (e.g., "Regular" or "Lecturer")
* Card Issued Date
* Card Expiry Date
* Status (e.g., "Active" or "Inactive")
* Email
* Photo
* Borrowing History
* Penalty Balance

***2, Book***:

* Book ID (Primary Key)
* ISBN
* Title
* Author
* Subject Area
* Binding Type (e.g., "Hardcover", "Softcover")
* Is Lendable (Yes/No)
* Copies Available
* Copies On Loan

***3, Catalog****:*

* Catalog ID (Primary Key)
* ISBN (Foreign Key to Book ISBN)
* Description

***4, Borrowing Activity***:

* Transaction ID (Primary Key)
* Member ID (Foreign Key to Member Member ID)
* Book ID (Foreign Key to Book Book ID)
* Borrow Date
* Due Date (*Deadline to return the book*)
* Return Date (*Actual return date of the book*)
* Status (e.g., "Borrowed", "Returned", "Overdue")

***5****,* ***Staff*** *:*

* Staff ID (Primary Key)
* Frist Name
* Last Name
* Role (e.g., Chief Librarian, Associate Librarian)
* Email
* Phone Number
* Shift (e.g., Morning, Evening)
* Date Hired (an employee officially begins working for an organization)

*6,* ***Publisher***:

* Publisher ID (Primary Key)
* FristaName
* LastName
* Address
* Phone Number
* Email

***7, Notifications:***

* Notification ID (Primary Key)
* Member ID (Foreign Key)
* Notification Date
* Notification Type (e.g., Overdue Reminder, Membership Renewal)
* Status (e.g., Sent, Resolved)

***Relationships****:*

***1.*Book - Catalog:**

One-to-one. A book (identified by ISBN) has one catalog entry.

2.**Book - Publisher:**

Many-to-one. A book is published by one publisher.

3.**Book – Borrowing Activity**

One-to-many. A book can be borrowed multiple times (in different transactions).

4.**Member – Borrowing Activity**

One-to-many. A member can borrow multiple books (in different transactions).

5.**Member - Notifications:**

One-to-many. A member can receive multiple notifications.

6.**Staff - Catalog (Adding to Catalog):**

One-to-many (or potentially many-to-many if multiple staff can add the same book). A staff member can add multiple books to the catalog.

7.**Staff - Member (Membership Management):** One-to-many. A staff member can manage multiple members (create or update accounts).

8.**Staff – Borrowing Activity (Loan Processing):**

One-to-many. A staff member can process multiple loans.

9.**Staff - Book (Book Acquisition/Ordering):**

One-to-many (or potentially many-to-many, if multiple staff can be involved in a single order). A staff member can place multiple book orders.

10.**Staff - Notifications (Managing Notifications):**

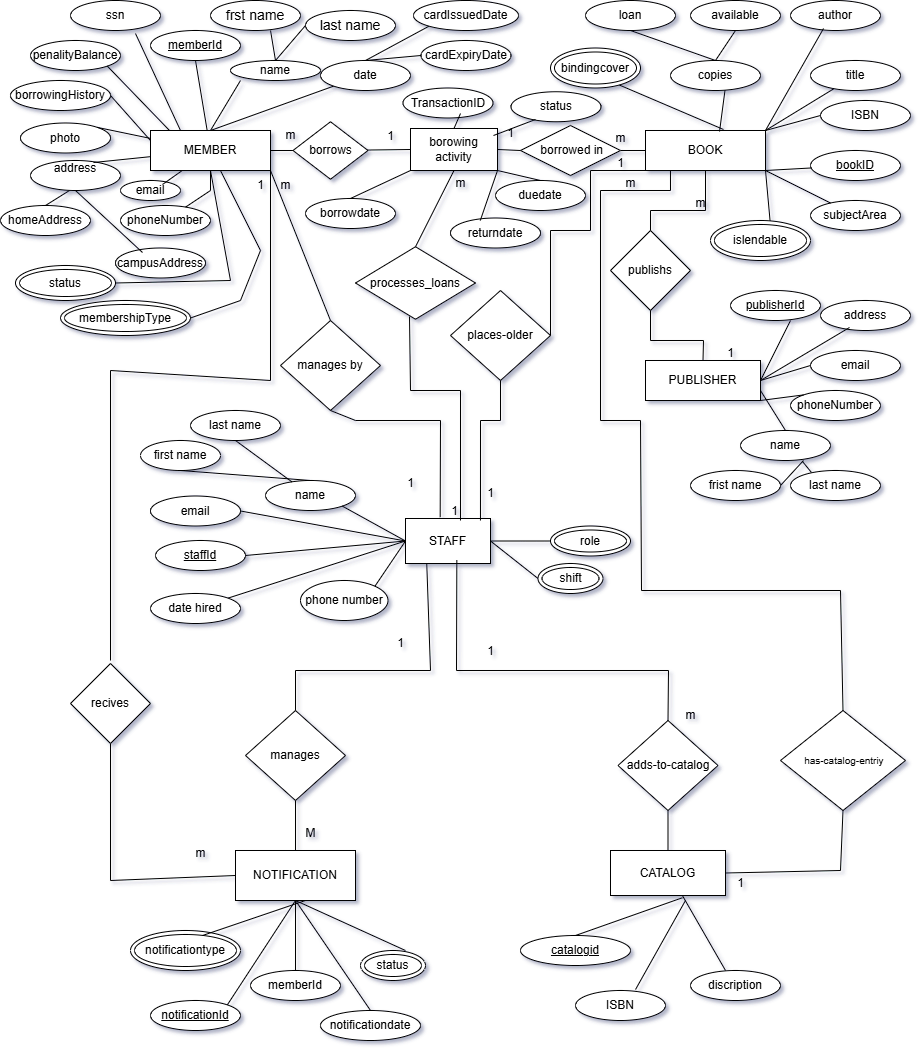
One-to-many. A staff member can manage multiple notifications.



Summary:

|  |  |  |
| --- | --- | --- |
| *Entity1* | *Relationship name* | *Entity2* |
| *Book* | *Has catalog entry* | *Catalog* |
| *Book* | *Published by* | *Publisher* |
| *Book* | *Borrow in* | *Borrowing activity* |
| *Member* | *Borrows* | *Borrowing activity* |
| *Member* | *Receives* | *Notification* |
| *Staff* | *Add to catalogs* | *Catalog* |
| *Staff* | *Manages* | *Member* |
| *Staff* | *Processes loans* | *Borrowing activity* |
| *Staff* | *Place orders* | *Book* |
| *Staff* | *Manages* | *Notification* |
|  |  |  |

* Draw entity relationship diagrams:



Map the entity relationship diagram to relational schema

MEMBER

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

BOOK

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

CATALOG

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

BORROWINGACTIVITY

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

*STAFF*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

*PUBLISHER*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

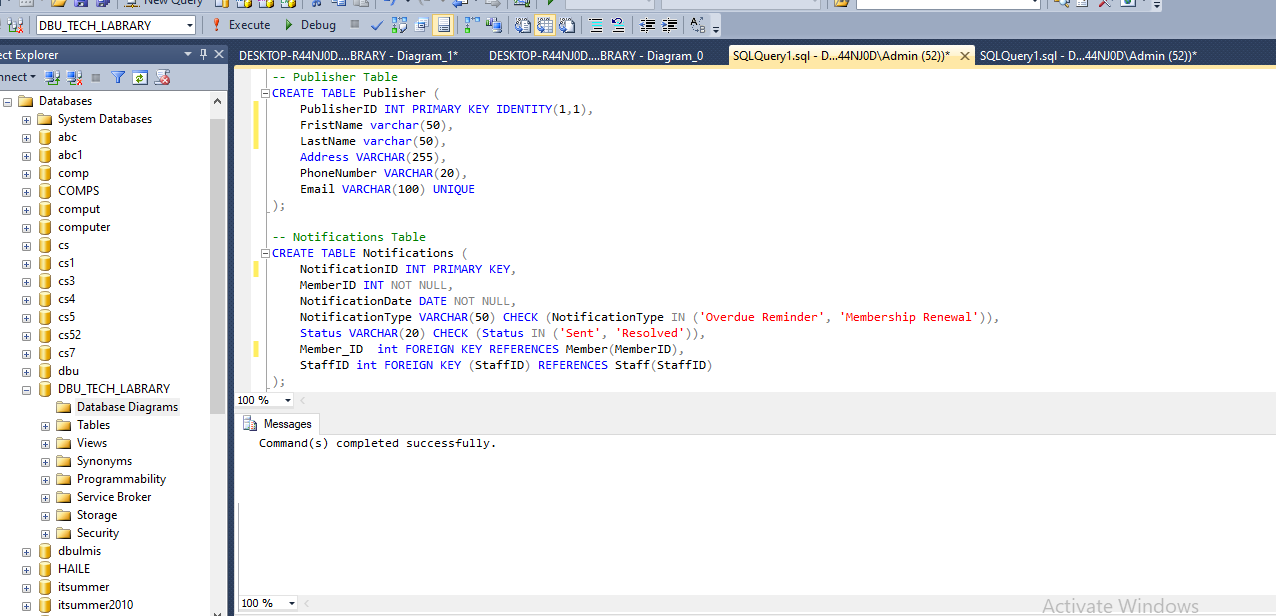
*NOTIFICATION*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

* Implement the relational schema using SQL server.

|  |
| --- |
| create database DBU\_TECH\_LABRARY  USE DBU\_TECH\_LABRARY          -- Member Table  CREATE TABLE Member (  MemberID INT PRIMARY KEY IDENTITY(1,1),  SSN VARCHAR(20) UNIQUE NOT NULL,  FirstName VARCHAR(50) NOT NULL,  LastName VARCHAR(50) NOT NULL,  CampusAddress VARCHAR(255),  HomeAddress VARCHAR(255),  PhoneNumber VARCHAR(20),  MembershipType VARCHAR(20) CHECK (MembershipType IN ('Regular', 'Lecturer')),  CardIssuedDate DATE,  CardExpiryDate DATE,  Status VARCHAR(20) CHECK (Status IN ('Active', 'Inactive')),  Email VARCHAR(100) UNIQUE,  Photo VARCHAR(255),  PenaltyBalance DECIMAL(10,2) DEFAULT 0.00,  StaffID int FOREIGN KEY REFERENCES Staff(StaffID)  );    -- Book Table  CREATE TABLE Book (  BookID INT PRIMARY KEY IDENTITY(1,1),  ISBN VARCHAR(20) UNIQUE NOT NULL,  Title VARCHAR(255) NOT NULL,  BindingType VARCHAR(20) CHECK (BindingType IN ('Hardcover', 'Softcover')),  IsLendable BIT NOT NULL,  PublicationYear INT,  PublisherID int FOREIGN KEY REFERENCES Publisher(PublisherID),  StaffID int foreign key REFERENCES Staff(StaffID)  );  -- Catalog Table  CREATE TABLE Catalog (  CatalogID INT PRIMARY KEY IDENTITY(1,1),  ISBN VARCHAR(20) NOT NULL,  Description TEXT,  FOREIGN KEY (ISBN) REFERENCES Book(ISBN),  StaffID int FOREIGN KEY REFERENCES Staff(StaffID)  );  -- BorrowingActivity (Loan) Table  CREATE TABLE BorrowingActivity (  TransactionID INT PRIMARY KEY IDENTITY(1,1),  BorrowDate DATE NOT NULL,  DueDate DATE NOT NULL,  ReturnDate DATE,  Status VARCHAR(20) CHECK (Status IN ('Borrowed', 'Returned', 'Overdue')),  StaffID INT, -- Foreign Key for Staff who processed the loan  FOREIGN KEY (MemberID) REFERENCES Member(MemberID),  FOREIGN KEY (BookID) REFERENCES Book(BookID),  FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)  );  -- Staff Table  CREATE TABLE Staff (  StaffID INT PRIMARY KEY IDENTITY(1,1),  FirstName VARCHAR(50) NOT NULL,  LastName VARCHAR(50) NOT NULL,  Role VARCHAR(50),  Email VARCHAR(100) UNIQUE,  PhoneNumber VARCHAR(20),  Shift VARCHAR(20),  DateHired DATE  );  -- Publisher Table  CREATE TABLE Publisher (  PublisherID INT PRIMARY KEY IDENTITY(1,1),  Name VARCHAR(255) NOT NULL,  Address VARCHAR(255),  PhoneNumber VARCHAR(20),  Email VARCHAR(100) UNIQUE  );  -- Notifications Table  CREATE TABLE Notifications (  NotificationID INT PRIMARY KEY IDENTITY(1,1),  MemberID INT NOT NULL,  NotificationDate DATE NOT NULL,  NotificationType VARCHAR(50) CHECK (NotificationType IN ('Overdue Reminder', 'Membership Renewal')),  Status VARCHAR(20) CHECK (Status IN ('Sent', 'Resolved')),  MemberID int FOREIGN KEY REFERENCES Member(MemberID),  StaffID int FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)  ); |

* 



1.create all table