

CSE 5330 Database Systems - Project 2 Part 1 Report

Navneeth Krishna
1002050459

1 Introduction

In the first part of the Project 2, the following assumptions were made after going through the project requirements as specified. This write-up describes the assumptions, the identified strong entities, weak entities and the relationships between the identified entities. The write-up also specifies how the EER diagram was mapped to a relational schema and also shows the different create table SQL commands used to create the tables.

2 EER Diagram

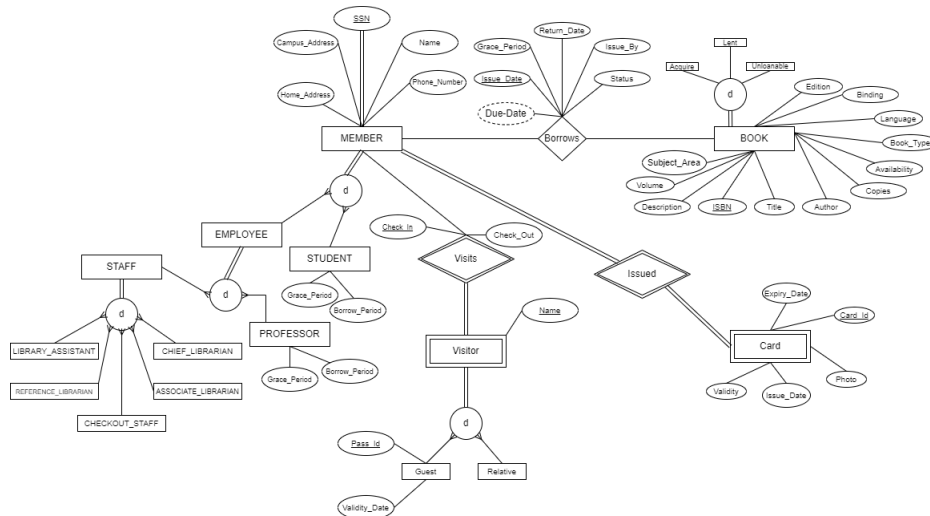


Figure 1: EER Diagram

3 Strong Entities

The following strong entities have been identified in the project:

1. **MEMBER:**

I have chosen to use the SSN, which is specific to each member, as the primary key for this object. Name, campus address, home address, and phone number are the other fields of this entity. There are separate organizations that specialize on Member, Employee, and Student. Grace Period and Borrow Period apply to the student. Employee, Staff, and Professor specializations are divided into separate organizations. Grace Period and Borrow Period are available to Professor. Chief Librarian, Associate Librarian, Reference Librarian, Checkout Staff, and Library Assistant are separate entities that specialize in different areas of the staff.

2. **CARD:**

This entity has Card_Id as its primary key. The other attributes are Issue_date, Expiry_date, Photo, Validity. Every card entry should have a corresponding entry in Member table.

3. **BOOK:**

The ISBN is exclusive to each book, is the primary key for this object. Title, Volume, Author, Language, Edition, Binding, Description, Subject Area, Book Type, and Availability are the other fields. The availability of a book is indicated by the attribute availability. Book Type refers to the category of books, such as magazines, maps, or reference books. There are various organizations that specialize in buying, lending, and loaning books.

4 Weak Entities

The following weak entities are identified in the project:

1. **VISITOR:**

This visitor maintains a record of each members', their relatives', or visitors' visits. Visitor's Name, Pass Id (Only For Guests), Validity Date (Pass Validity Date), Check In, and Check Out are the attributes of this entity. In the Member table, each visitor will have a corresponding record. There are separate organizations that specialize in visitors, guests, and relatives. The separated entities will be tracked by Visitor Type. Self, visitor, and relative.

2. **COPY:**

This entity has the records for all the books in the library. Its primary key is Book_Id. And other attributes of this entity are ISBN, Status. Every Copy entry should have a corresponding entry in Book table. The detail description of book is available in Book table and the details about the number of copies and their status is in Copy Table.

5 Relationships

The following relationships have been identified in the project:

1. **BORROWS:**

This is a one-to-many relationship between entities MEMBER and COPY, one member can borrow up to five copies at any point in time. Due date is derived from Borrow_Period and Issue_Date. Return date is when the book is returned by the member. Grace_Period date is the date calculated based on issue date and Grace_Period. Status is to keep track of the book whether it was returned and still is borrowed. Issued_By is the SSN number of the staff who issued the book to the member.

2. **ISSUED:**

This is a one-to-one relationship between entities MEMBER and CARDS. Both the entities are totally participating in this relationship as Cards is weak entity. Every card entry should have a corresponding Member in Member table.

3. **HAVE:**

This is a One-to-many relationship between entities BOOK and COPY. Both the entities are totally participating in this relationship. Every entry in Copy table should have corresponding entry in the book table.

4. **VISITS:**

Visits have a recursive relation when the member itself is visiting. Visitors is weak entity and Visits ties Visitor and Member entities. As every visitor should have a corresponding member in the MEMBER table. Visitors and Member entities are totally participating in this relationship.

6 Relational Diagram



Figure 2: Relational Schema Diagram

7 Mapping EER Diagram to Relational Schema

The following tables have been recognised when I mapped the EER diagram to relational schema:

1. MEMBER:

This is a strong entity. It has SSN as primary key and other attributes same as that in EER diagram. I merged all the specialized table into Member table so that it is easier. I took Member_Type attribute to specify Employee or Student. I took Employee_Type attribute to specify Staff or Professor. I took Staff_Type to specify chief librarian, departmental associate librarian, reference librarian, checkout staff, and library assistant. The attributes of Professor and Student entity are merged as Grace_Period and Borrow_Period attribute. Members can borrow max 5 book at any given point in time. Students can borrow books for 21 days. Students have a Grace Period of 1 week and Professors have a grace period of 2 weeks. Professors can borrow books for 3 months.

2. BOOK:

This is a strong entity. It has ISBN as primary key and other attributes same as in EER diagram. I have merged all the specialized table into Book table. I have taken Status_Type attribute to specify whether Acquire, Lent or loaned books. Book_Type to specify whether it's a reference book or magazine or a Map, etc.

3. **COPY:**

This table is directly mapped from the EER diagram. It has primary key as Book_Id and other attributes same as that in EER diagram. I added a foreign key ISBN which refers to the ISBN in BOOK table. Status gives the information about the book Availability. Book_Id is Unique for every book. Each copy of a book will have different Book_Id.

4. **CARD:**

This table is directly mapped from the EER diagram. It has primary key as Card_Id and other attributes same as that in EER diagram. I added a foreign key M.SSN which refers to the SSN in Member table to remove the weak entity relationship.

5. **BORROW:**

This table has primary keys M.SSN, BOOK_ID, ISSUE_DATE and other attributes same as that in EER. I added a foreign key BOOK_ID which refers to the BOOK_ID in COPY table, M.SSN which refers to the SSN in MEMBER table and ISSUED_BY which refers to the SSN in MEMBER table which specifies who issued the book. Book_Id uniquely identifies each book in the library thus it can keep track of every loaned book in the library. Due date is derived from the Issue_date and Grace period depending upon whether the member is a Student or a Professor. Return Date is when the member returned the book. The return date can be before or after due date. Status keeps the track of whether the book is returned or not.

6. **VISITS:**

This table has primary keys M.SSN, CHECK_IN and other attributes same as that in EER diagram. I added a foreign key M.SSN which refers to the SSN in MEMBER table. To overcome the weak entity, I mapped each entry to a Member.SSN. I merged all the Specialization table into Visits table. I took Visitor_Type attribute to specify whether Guest or Relative. Every visitor will have a Name. If the visitor is Guest, then he/she will have Visitor_Pass and that pass will be valid for only 1 day, that day is specifies by Validity_Date attribute.

8 CREATE TABLE SQL commands

The following are the create table commands used in creating the database tables:

```
CREATE TABLE MEMBER
(
SSN INT,
NAME VARCHAR(50),
HOME_ADDRESS VARCHAR(1000),
CAMPUS_ADDRESS VARCHAR(1000),
```

```

PHONE_NO VARCHAR(50),
MEMBER_TYPE VARCHAR(50),
GRACE_PERIOD INT,
BORROW_PERIOD INT,
EMPLOYEE_TYPE VARCHAR(50),
STAFF_TYPE VARCHAR(50),
PRIMARY KEY(SSN)
);

CREATE TABLE BOOK
(
ISBN INT,
TITLE VARCHAR(100),
BINDING VARCHAR(20),
LANGUAGE VARCHAR(100),
EDITION VARCHAR(30),
AUTHOR VARCHAR(50),
DESCRIPTION LONGTEXT,
VOLUME VARCHAR(30),
SUBJECT_AREA VARCHAR(100),
BOOK_TYPE VARCHAR(30),
AVAILABILITY CHAR(1) NOT NULL CHECK(AVAILABILITY IN('Y', 'N')),
STATUS_TYPE VARCHAR(30),
PRIMARY KEY(ISBN)
);

CREATE TABLE CARD
(
CARD_ID INT,
PHOTO BLOB,
ISSUE_DATE DATE,
EXPIRY_DATE DATE,
VALIDITY DATE,
M_SSN INT,
PRIMARY KEY (CARD_ID),
FOREIGN KEY (M_SSN) REFERENCES MEMBER(SSN) ON DELETE CASCADE
);

CREATE TABLE COPY(
BOOK_ID INT,
ISBN INT NOT NULL,
STATUS VARCHAR(20),
PRIMARY KEY(BOOK_ID),
FOREIGN KEY(ISBN) REFERENCES BOOK(ISBN) ON DELETE CASCADE
);

```

```

CREATE TABLE BORROW
(
BOOK_ID INT,
M_SSN INT,
ISSUED_BY INT,
ISSUE_DATE DATE,
DUE_DATE DATE,
RETURN_DATE DATE,
GRACE_PERIOD DATE,
STATUS VARCHAR(20),
PRIMARY KEY(BOOK_ID,M_SSN,ISSUE_DATE),
FOREIGN KEY(BOOK_ID) REFERENCES COPY(BOOK_ID),
FOREIGN KEY(M_SSN) REFERENCES MEMBER(SSN),
FOREIGN KEY(ISSUED_BY) REFERENCES MEMBER(SSN)
);

```

```

CREATE TABLE VISITS
(
M_SSN INT,
CHECK_IN DATE,
CHECK_OUT DATE,
NAME VARCHAR(50),
PASS_ID INT,
VISITOR_TYPE VARCHAR(50),
VALIDITY_DATE DATE,
PRIMARY KEY (M_SSN,CHECK_IN),
FOREIGN KEY (M_SSN)REFERENCES MEMBER(SSN)
);

```