CMPT 354 Mini Project

Rauf Shimarov (301397321)

Praneer Shrestha (301361473)

Project Specifications (STEP 2)

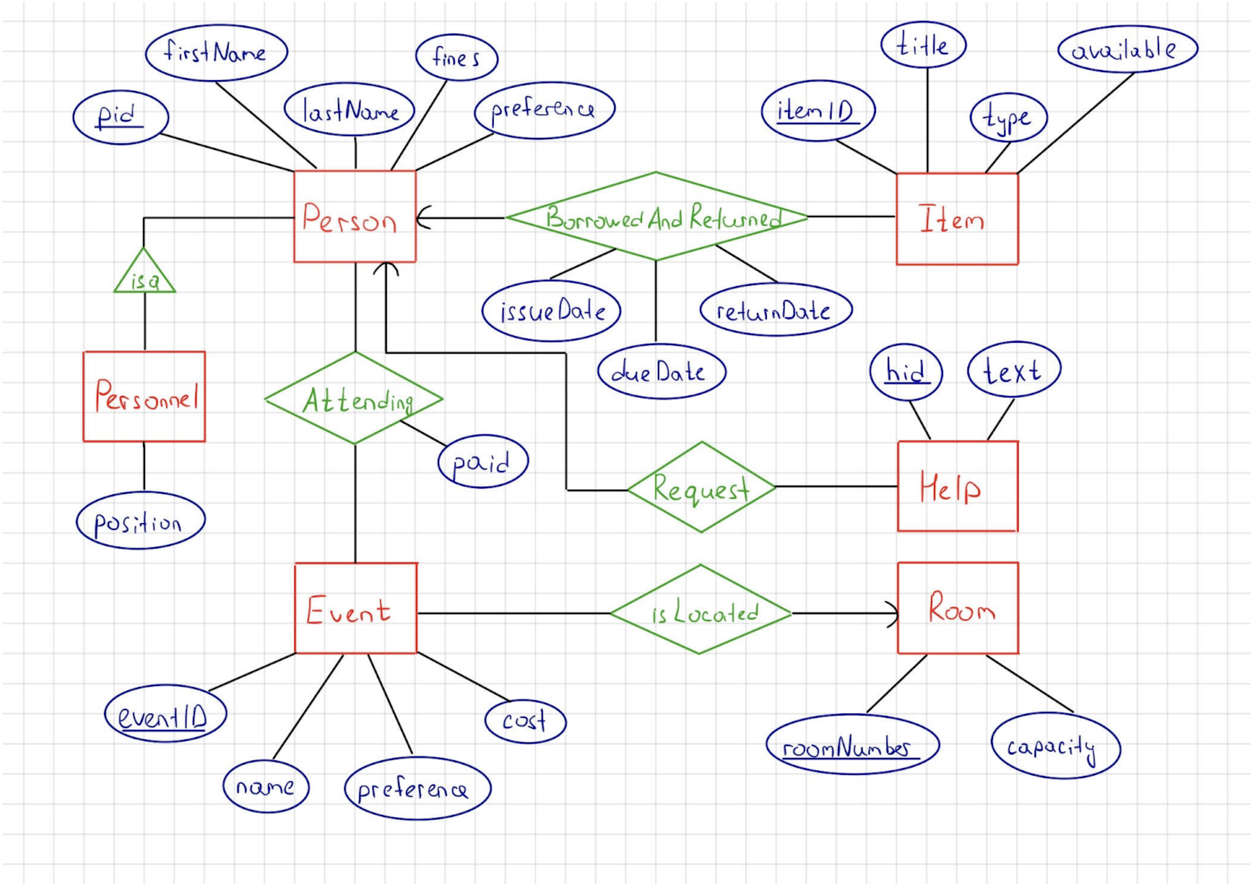
- Library has print books, online books, magazines, scientific journals, CDs, records, etc.
- People can borrow the items from library and return by the due date.
- People may be subject to fines if they do not return items by the due date.
- Library also holds book clubs, book related events, art shows, film screenings, etc.
- Library events are recommended for specific audiences.
- Library events are held on library social rooms.
- People can attend library events for free.
- Library also has personnel and record keeping for personnel.
- Library also keeps records of items (books, etc.) that might be added to library in the future.

Our extensions to the minimum project specifications:

- Library identified items by their unique id numbers.
- Each item can have a title and will always have an availability to be borrowed.
- People cannot borrow items that are currently borrowed.
- Due dates can be extended for items people have borrowed.
- Once borrowed, the item becomes unavailable, and vice versa, once returned the item becomes available.[1]
- The due dates that items have to be returned by can never be before the issue date.[2]
- Members that return an item passed its due date will be fined $10.[3]
- Items are due after two weeks of the issue date (the date it was borrowed).[4]
- Members of the library have a unique member id number.
- People have first names, last names, fines and preferences for events.
- Personnel are also members of the library, and have positions in the library.
- An event is identified through a unique id number and has a name.
- Events have a cost which can be 0, that people have to pay in order to attend.[5]
- Events that are free will automatically assume that people attending have paid.
- Rooms are identified by their unique id numbers and contain a certain capacity.
- Library stores all questions and help requested by members.

[1] See trigger created in Appendix (page 7-8)
[2] See constraint created in Appendix (page 5)
[3] This is handled through the library.py program
[4] This is handled through the library.py program
[5] See trigger created in Appendix (page 8)

E/R Diagram (STEP 3)



Analysis of anomalies (STEP 4)

Schemas:

1. Person (pid, firstName, lastName, preference, fines)
2. Personnel (pid $^{\text{FK-Person}}$, position)
3. Item (itemID, title, type, available)
4. Event (eventID, name, preference, cost)
5. Room (roomNumber, capacity)
6. Help (hid, text)
7. Attending (pid $^{\text{FK-Person}}$, eventID$^{\text{FK-Event}}$, paid)
8. BorrowedAndReturned (itemID$^{\text{FK-Item}}$, pid $^{\text{FK-Person}}$, issueDate, dueDate, returnDate)
9. IsLocated (eventID$^{\text{FK-Event}}$, roomNumber$^{\text{FK-Room}}$)
10. Request (pid $^{\text{FK-Person}}$, hid$^{\text{FK-Help}}$)

FD’s:

- Person
    - pid -> firstName, lastName, preference, fines
- Personnel
    - pid -> position
- Item
    - itemID -> title, type, available
- Event
    - eventId -> name, preference, cost
- Room
    - roomNumber -> capacity
- Help
    - hid -> text
- Attending
    - pid, eventID -> paid
- BorrowedAndReturned
    - pid, itemID -> issueDate, dueDate, returnDate
- Is located
    - eventID -> roomNumber, roomNumber -> event ID
- Request
    - pid -> hid, pid-> hid hold

The left side of pid -> personType, firstName, lastName, fines FD is a superset of Person relation’s key set. Hence, Person relation is in BCNF. Similarly, since the left side of each FD in Personnel, Event, Item, Room, Attending, Help is a superset of the key set of the corresponding relation, these relations are in BCNF.

Since the BorrowedAndReturned relation has no primary keys, the key set of this relation is empty. Any set is a superset of an empty set. Hence, the left side of pid, itemID -> issueDate, dueDate, returnDate FD is a superset of the (empty) key set, and consequently, BorrowedAndReturned is in BCNF

Also, since both sides of all FD’s in isLocated and Request relations are the keys, these two relations are in BCNF.

All relations are in BCNF, therefore, there are no anomalies in our database design.

Appendix

Creation of the Library Database - SQL Commands

Creation of Entities:

Item

```
CREATE TABLE Item (
        itemID INT,
        title VARCHAR(30),
        type VARCHAR(30),
        available BOOLEAN NOT NULL DEFAULT 1,
        PRIMARY KEY (itemID)
);
```

BorrowedAndReturned

```
CREATE TABLE BorrowedAndReturned (
        itemID INT,
        pid INT,
        issueDate INT(8) NOT NULL,
        dueDate INT(8) NOT NULL,
        returnDate INT(8) DEFAULT NULL,
        FOREIGN KEY (itemID) REFERENCES Item(itemID),
        FOREIGN KEY (pid) REFERENCES Person(pid),
        CONSTRAINT dateCheck CHECK (issueDate <= dueDate)
);
```

Person

```
CREATE TABLE Person (
        pid INT,
        firstName VARCHAR (30),
        lastName VARCHAR (30),
        preference VARCHAR(30),
        fines REAL DEFAULT 0,
        PRIMARY KEY (pid)
);
```

Personnel

```
CREATE TABLE Personnel (
	pid INT,
	position VARCHAR(30),
	PRIMARY KEY (pid),[6]
	FOREIGN KEY (pid) REFERENCES Person(pid)
);
```

Attending

```
CREATE TABLE Attending (
	pid INT,
	eventID INT,
	paid BOOLEAN DEFAULT 0,
	PRIMARY KEY (pid, eventID),[7]
	FOREIGN KEY (pid) REFERENCES Person(pid),
	FOREIGN KEY (eventID) REFERENCES Event(eventID)
);
```

Event

```
CREATE TABLE Event (
	eventID INT,
	name VARCHAR(30),
	preference VARCHAR (30) DEFAULT NULL,
	cost REAL NOT NULL,
	PRIMARY KEY (eventID)
);
```

Room

```
CREATE TABLE Room (
	roomNumber INT,
	capacity INT,
	PRIMARY KEY (roomNumber)
);
```

[6] Made (pid) a primary key so that the same person can register to become a volunteer at most once.
[7] Made (pid, eventID) a primary key so that the same person cannot register to an event more than once.

LocatedIn

```
CREATE TABLE LocatedIn (
        eventID INT,
        roomNumber INT,
        PRIMARY KEY (eventID, roomNumber),
        FOREIGN KEY (eventID) REFERENCES Event (eventID),
        FOREIGN KEY (roomNumber) REFERENCES Room (roomNumber )
);
```

Help

```
CREATE TABLE Help (
        hid INT,
        text VARCHAR(100) NOT NULL,
        PRIMARY KEY (hid)
);
```

Request

```
CREATE TABLE Request (
        pid INT,
        hid INT,
        PRIMARY KEY (pid, hid),
        FOREIGN KEY (pid) REFERENCES Person(pid),
        FOREIGN KEY (hid) REFERENCES Help(hid)
);
```

Creation of Triggers:

```
CREATE TRIGGER is_borrowed
AFTER INSERT ON
BorrowedAndReturned
BEGIN
UPDATE Item SET available = 0 WHERE  Item.itemID = New.itemID AND
Item.available = 1;
END;
```

```
CREATE TRIGGER is_available
AFTER UPDATE ON
BorrowedAndReturned
WHEN
New.returnDate IS NOT NULL
BEGIN
UPDATE Item SET available = 1 WHERE  Item.itemID = New.itemID ;
END;

CREATE TRIGGER has_paid
AFTER INSERT ON
Attending
BEGIN
UPDATE Attending SET paid=1
WHERE Attending.eventID =
        (SELECT eventID
        FROM Event
        Where cost=0);
END;
```