Group Name: NoName

Members: Nhat Doan, Joseph Dandro, Ali Badr A Aljehani

**Project: Library Database**

**Idea:** A database that helps track access to whom has accessed a record or who maintains a record (such as a library of sorts). There would be a superclass of users with a unique ID, with subclasses of a curator and visitor. The record itself is an object that has a unique reference id and can be classified as only a single subclass type of item (book, scientific journal, newspaper article, ect.) with its own respective attributes.

**Requirements:**

1. Users can be either or [overlap] Curator and User
2. A trigger can be used to insure a newly added item is only a single type of item. [Disjoint]
3. Items must be maintained by a curator [total participation] (Relation: Last updated)
4. Curators can maintain many items. [carnality]
5. Many records can be accessed by many users (Relation: Last Accessed)

a. Visitors should be able to see the list of records accessed and when

1. Can view the list of items maintained by the curators [can be materialized views]

a. Triggers to keep updated is materialized view

1. Use a X language to insert an item that is maintained

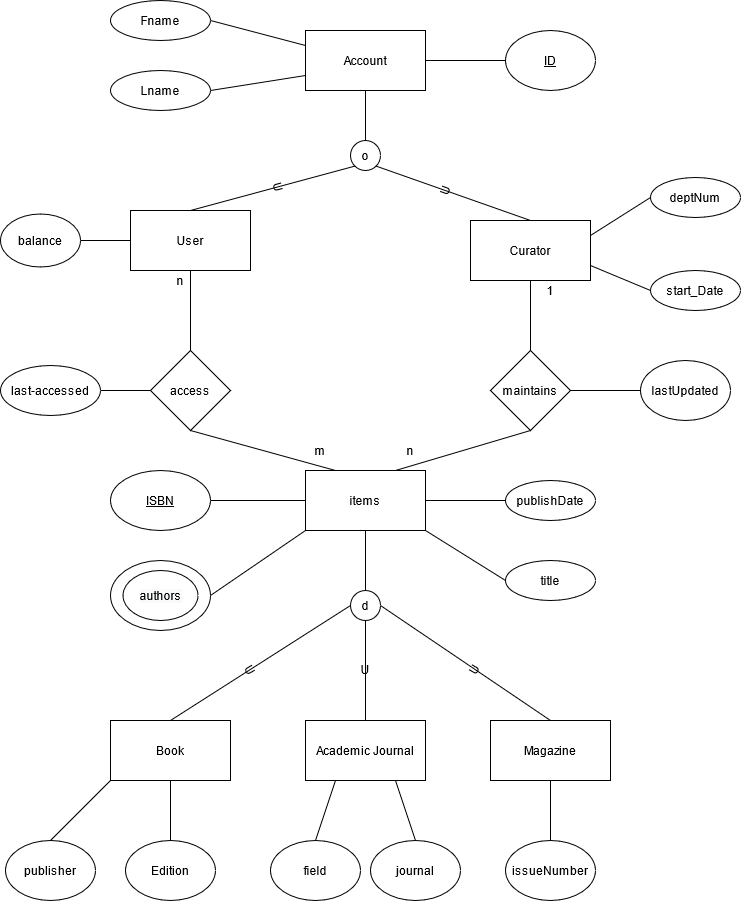
a. Trigger to update #4

b. The query | view as needed.

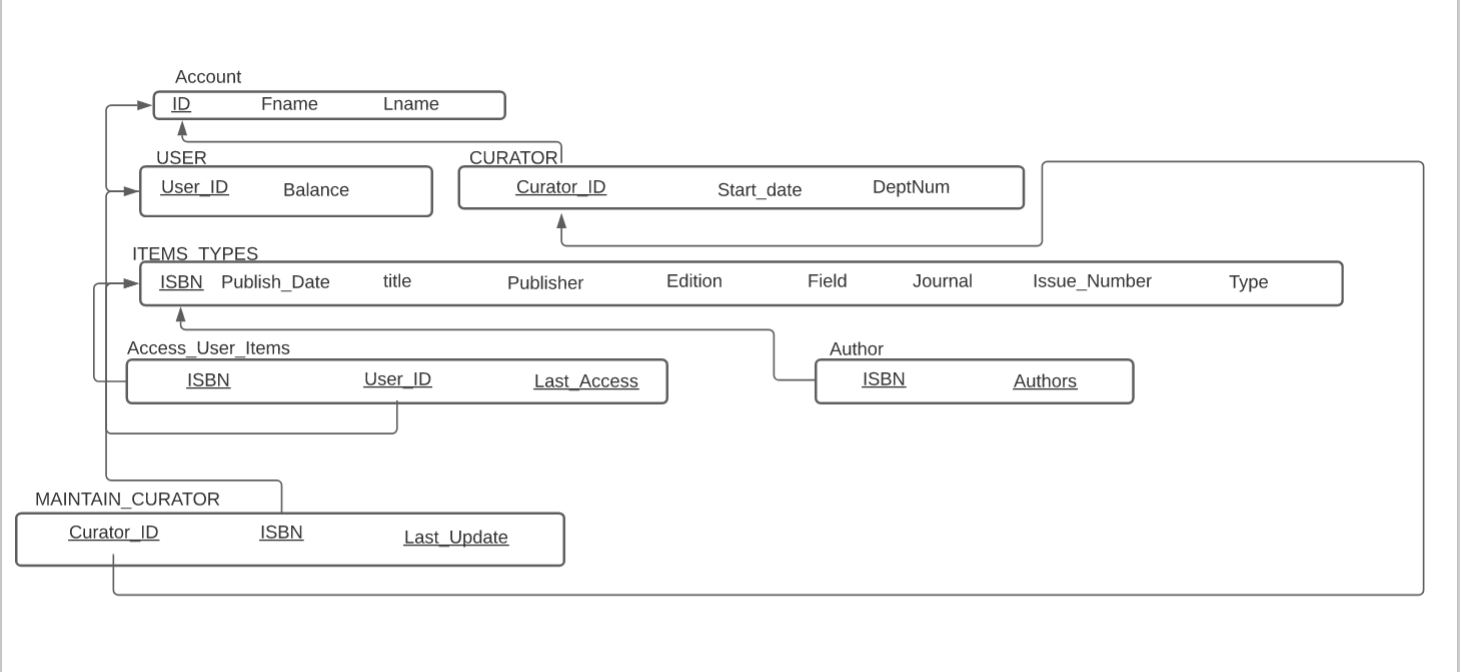
1. Use a stored procedure to create a random barcode when a customer rent an item and use that same barcode to return that item
2. Use a function to delete a certain barcode after a customer returns an item.

As per instructions - Something web browser friendly for the front-end

**EER Diagram**

****

**Relational Diagram**

****

**SQL statements**

/\*Creating Table\*/

/\*Drop tables if exist\*/

DROP TABLE IF EXISTS `ACCOUNT`;

DROP TABLE IF EXISTS `USER`;

DROP TABLE IF EXISTS `CURATOR`;

DROP TABLE IF EXISTS `AUTHOR`;

DROP TABLE IF EXISTS `ITEMS`;

DROP TABLE IF EXISTS `ACCESS\_ITEMS`;

DROP TABLE IF EXISTS `MAINTAIN\_ITEMS`;

/\*CREATE ACCOUNT TABLE\*/

CREATE TABLE ACCOUNT(

ID INT NOT NULL,

LNAME VARCHAR(10) NOT NULL,

FNAME VARCHAR(10) NOT NULL,

PRIMARY KEY (ID)

);

/\*CREATE USER TABLE\*/

CREATE TABLE USER(

USER\_ID INT NOT NULL,

BALANCE INT NOT NULL,

PRIMARY KEY (USER\_ID),

FOREIGN KEY (USER\_ID) REFERENCES ACCOUNT(ID) ON DELETE CASCADE ON UPDATE CASCADE

);

/\*CREATE TABLE CURATOR\*/

CREATE TABLE CURATOR(

CURATOR\_ID INT NOT NULL,

DEPT\_NUM INT NOT NULL,

START\_DATE DATE NOT NULL,

PRIMARY KEY (CURATOR\_ID),

FOREIGN KEY (UCURATOR\_ID) REFERENCES ACCOUNT(ID) ON DELETE CASCADE ON UPDATE CASCADE

);

/\*CREATE ITEMS TABLE\*/

CREATE TABLE ITEMS(

ISBN INT NOT NULL,

PUBLISH\_DATE DATE NOT NULL,

TITLE VARCHAR(100) NOT NULL,

PUBLISHER VARCHAR(100),

EDITION VARCHAR(10),

FIELD VARCHAR(30),

JOURNAL VARCHAR(100),

ISSUE\_NUMBER INT,

TYPE ENUM('BOOK','ACADEMIC\_JOURNAL','MAGAZINE'),

PRIMARY KEY (ISBN)

);

/\*CREATE TABLE AUTHOR\*/

CREATE TABLE AUTHOR(

ISBN INT NOT NULL,

AUTHOR VARCHAR(100) NOT NULL,

PRIMARY KEY (ISBN, AUTHOR),

FOREIGN KEY (ISBN) REFERENCES ITEMS(ISBN) ON DELETE CASCADE ON UPDATE CASCADE

);

/\*CREATE TABLE MAINTAIN\_ITEMS\*/

CREATE TABLE MAINTAIN\_ITEMS(

ISBN INT NOT NULL,

URATOR\_ID INT NOT NULL,

LAST\_UPDATE DATE NOT NULL,

PRIMARY KEY (CURATOR\_ID,ISBN, LAST\_UPDATE),

FOREIGN KEY (CURATOR\_ID) REFERENCES CURATOR(CURATOR\_ID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ISBN) REFERENCES ITEMS(ISBN) ON DELETE CASCADE ON UPDATE CASCADE

);

/\*CREATE TABLE ACCESS\_ITEMS\*/

CREATE TABLE ACCESS\_ITEMS(

ISBN INT NOT NULL,

USER\_ID INT NOT NULL,

LAST\_ACCESS DATE NOT NULL,

PRIMARY KEY (CURATOR\_ID,ISBN, LAST\_ACCESS),

FOREIGN KEY (USER\_ID) REFERENCES USER(USER\_ID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ISBN) REFERENCES ITEMS(ISBN) ON DELETE CASCADE ON UPDATE CASCADE

);

**Hosting the project**

We are planning to host our website based on html, css, JS on github. Then we will use PHP for back-end connected to a database hosted on an ECS account.