**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

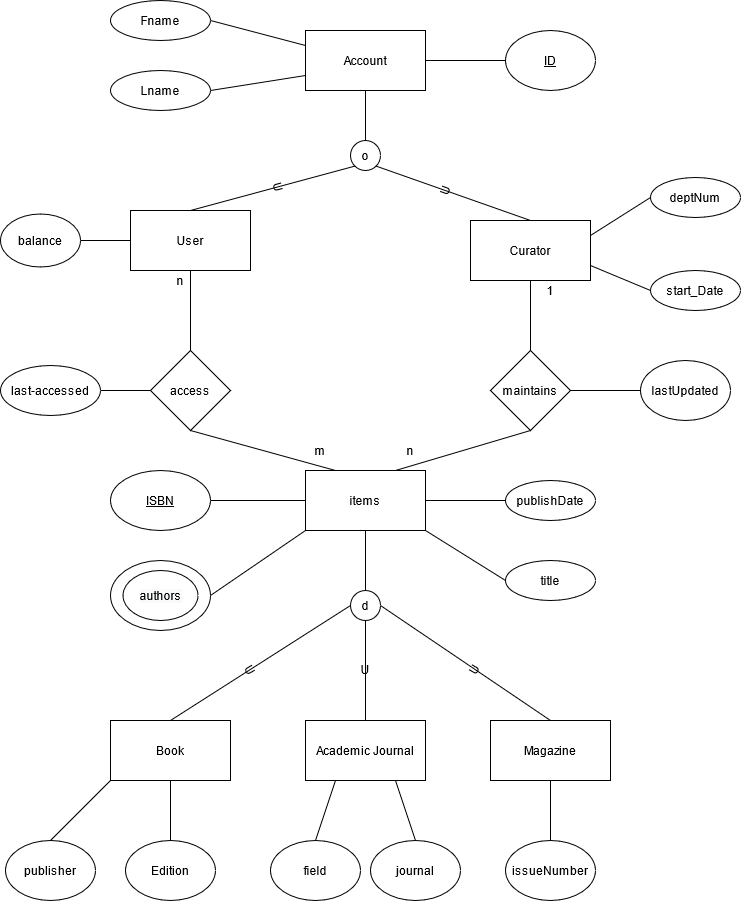
1. Can view the list of all the books from items table (name query)
2. Use a X language to insert an item that is maintained

a. Trigger to update #4

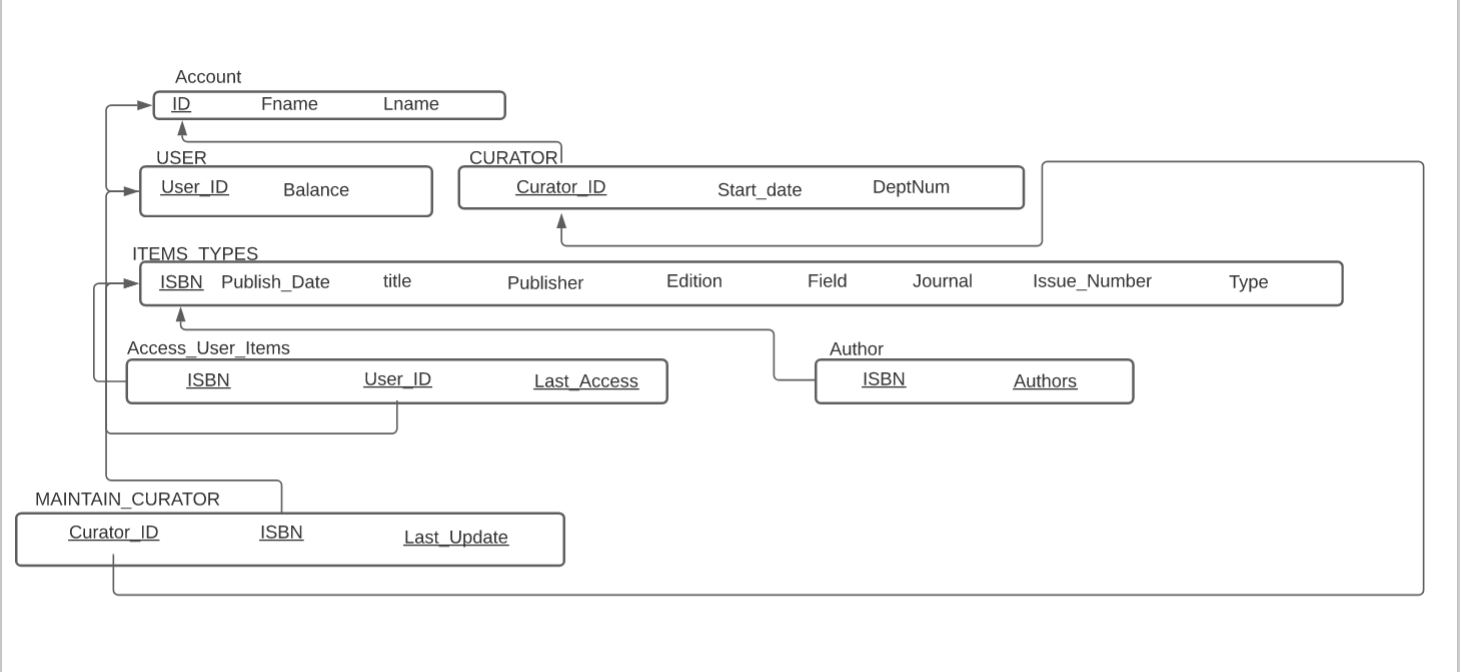
1. Use a stored procedure to create a random barcode when a customer rent an item and use that same barcode to return that item

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 `ITEMS`;

DROP TABLE IF EXISTS `ACCESS\_ITEMS`;

DROP TABLE IF EXISTS `MAINTAIN\_ITEMS`;

DROP TABLE IF EXISTS `AUTHOR`;

/\*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,

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,

PRIMARY KEY (CURATOR\_ID),

FOREIGN KEY (CURATOR\_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,

TITTLE 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'),

BAR\_CODE VARCHAR(50),

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,

CURATOR\_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 (USER\_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

);

**/\*Populate Data\*/**

INSERT INTO `ACCOUNT` (`ID`, `LNAME`, `FNAME`) VALUES ('1', 'nhat', 'doan');

INSERT INTO `ACCOUNT` (`ID`, `LNAME`, `FNAME`) VALUES ('2', 'trinh', 'nguyen');

INSERT INTO `USER` (`USER\_ID`) VALUES ('1');

INSERT INTO `USER` (`USER\_ID`) VALUES ('2');

INSERT INTO `CURATOR` (`CURATOR\_ID`) VALUES ('1');

INSERT INTO `ITEMS` (`ISBN`, `PUBLISH\_DATE`, `TITTLE`, `PUBLISHER`, `EDITION`, `FIELD`, `JOURNAL`, `ISSUE\_NUMBER`, `TYPE`, `BAR\_CODE`) VALUES ('1', '2002-04-25', 'Wild Nature', 'Google', 'v1', NULL, NULL, NULL, 'BOOK',NULL);

INSERT INTO `ITEMS` (`ISBN`, `PUBLISH\_DATE`, `TITTLE`, `PUBLISHER`, `EDITION`, `FIELD`, `JOURNAL`, `ISSUE\_NUMBER`, `TYPE`, `BAR\_CODE`) VALUES ('2', '2021-04-25', 'The Whole New World', 'apple', 'v2', NULL, NULL, NULL, 'BOOK',NULL);

INSERT INTO `ITEMS` (`ISBN`, `PUBLISH\_DATE`, `TITTLE`, `PUBLISHER`, `EDITION`, `FIELD`, `JOURNAL`, `ISSUE\_NUMBER`, `TYPE`, `BAR\_CODE`) VALUES ('3', '2021-01-25', 'Stocks to the moon', 'NBC', NULL, 'economy', 'e market', NULL, 'ACADEMIC\_JOURNAL',NULL);

INSERT INTO `ITEMS` (`ISBN`, `PUBLISH\_DATE`, `TITTLE`, `PUBLISHER`, `EDITION`, `FIELD`, `JOURNAL`, `ISSUE\_NUMBER`, `TYPE`, `BAR\_CODE`) VALUES ('4', '2002-04-22', 'Free donuts after Covid-19 vacination ', 'youtube', NULL, NULL, NULL, '10101', 'MAGAZINE',NULL);

INSERT INTO `AUTHOR` (`ISBN`, `AUTHOR`) VALUES ('4', 'Alexandra IV');

INSERT INTO `AUTHOR` (`ISBN`, `AUTHOR`) VALUES ('3', 'Daniel Kim');

INSERT INTO `AUTHOR` (`ISBN`, `AUTHOR`) VALUES ('2', 'Jasmine Milktea');

INSERT INTO `AUTHOR` (`ISBN`, `AUTHOR`) VALUES ('1', 'Baby Shark');

INSERT INTO `MAINTAIN\_ITEMS` (`ISBN`, `CURATOR\_ID`, `LAST\_UPDATE`) VALUES ('4', '1', '2021-04-26');

INSERT INTO `ACCESS\_ITEMS` (`ISBN`, `USER\_ID`, `LAST\_ACCESS`) VALUES ('1', '2', '2021-04-24');

**/\*Name Query\*/**

CREATE VIEW ALL\_BOOKS AS

SELECT \*

FROM ITEMS

WHERE ITEMS.TYPE = “BOOK”

**/\*Trigger\*/**

CREATE TRIGGER CREATE\_BAR\_CODE

AFTER INSERT ON ACCESS\_ITEMS FOR EACH ROW

CALL GENERATE\_BAR\_CODE(@RESULT);

UPDATE ITEMS SET ITEMS.BAR\_CODE = @RESULT WHERE ITEMS.ISBN = ISBN;

CREATE TRIGGER REMOVE\_BAR\_CODE

AFTER UPDATE ON ACCESS\_ITEMS FOR EACH ROW

UPDATE ITEMS SET ITEMS.BAR\_CODE = NULL WHERE ITEMS.ISBN = NEW.ISBN;

**/\*PROCEDURE\*/**

CREATE PROCEDURE GENERATE\_BAR\_CODE(OUT RESULT VARCHAR(50))

BEGIN

SET @RESULT = ‘RANDOM STRING WITH 50 CHARACTERS’;

END

**Hosting the project**

We are planning to host our website based on html, css, JS and use PHP for back-end connected to a database hosted on an AWS account through EC2 instance.

URL to the project: <http://nonamecsc174csus.tk/>

There are some features from the website:

* Create a new book
* Delete a book
* Edit information from a book