Campus Item Database:

Database design and implementation

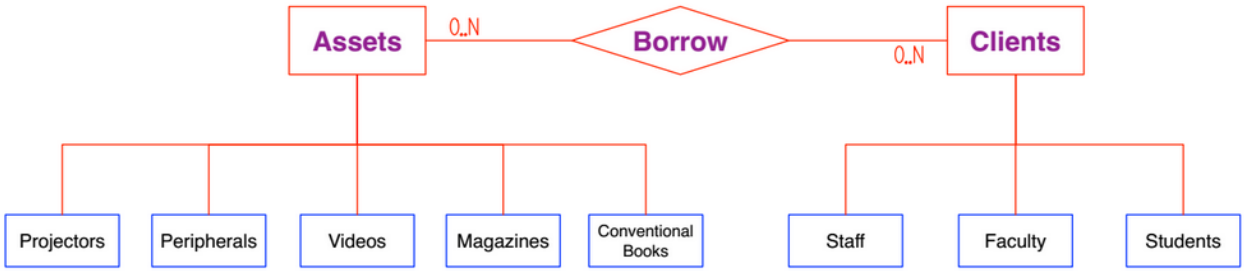
Table of Contents:

Introduction

This document outlines the design for a database check out system. The library already has a functional system, but there is no way in the current system to track miscellaneous items such as DVDs that belong to clubs, projectors, magazines, and other such items.

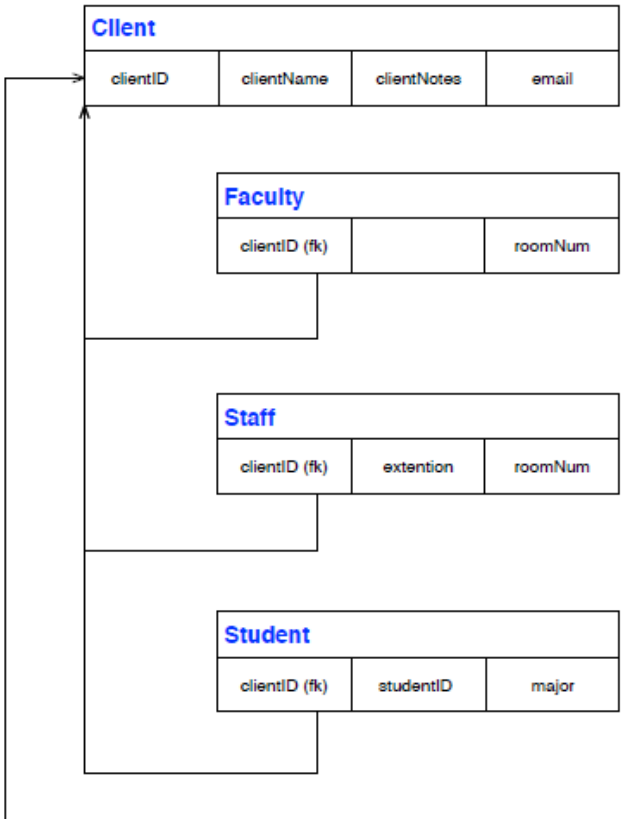
This document presents our own solution for this problem. It is a separate database that will track check out and return of these miscellaneous items. It allows for any period of check out that is not restricted to the rules of the other library system.

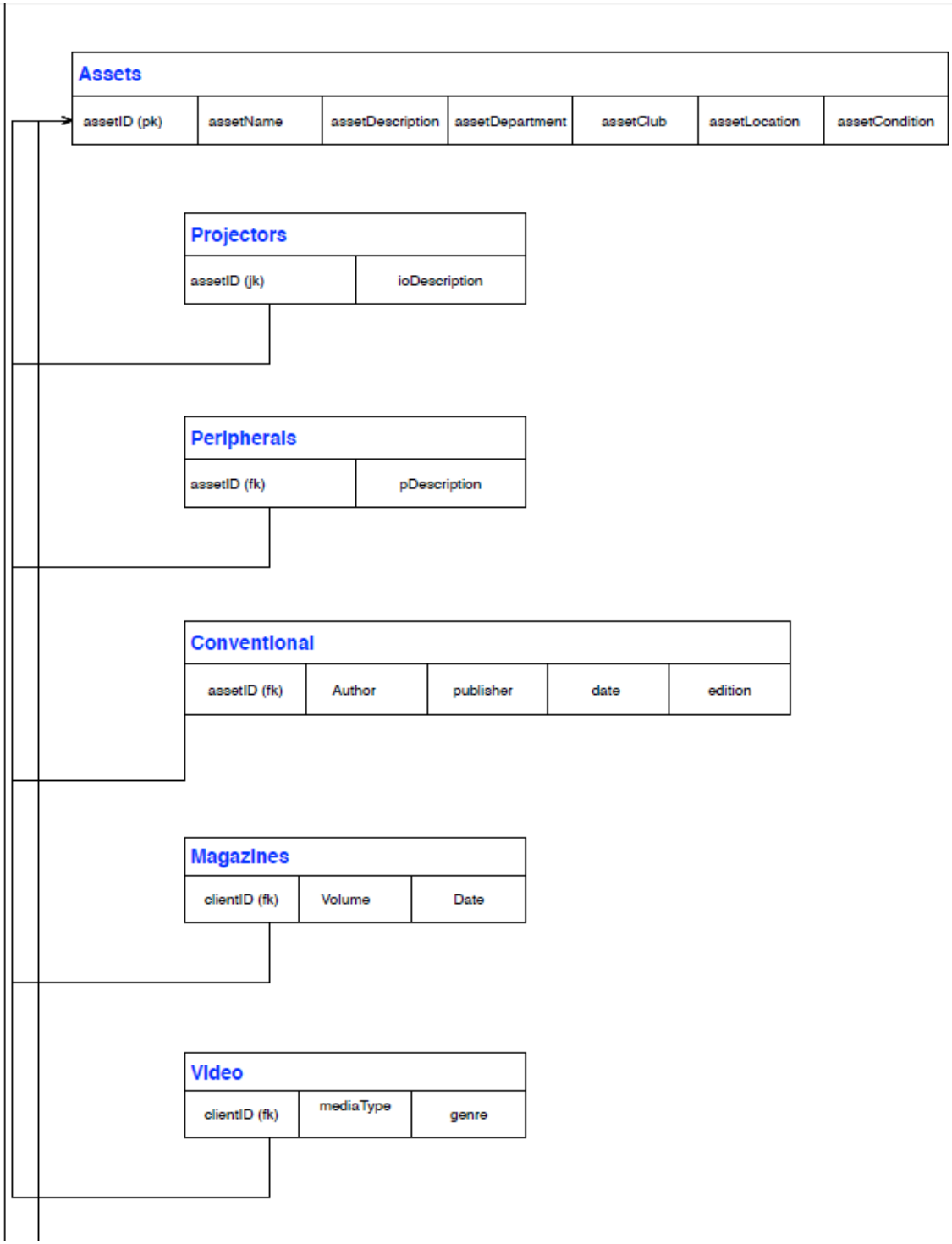
E-R Diagram

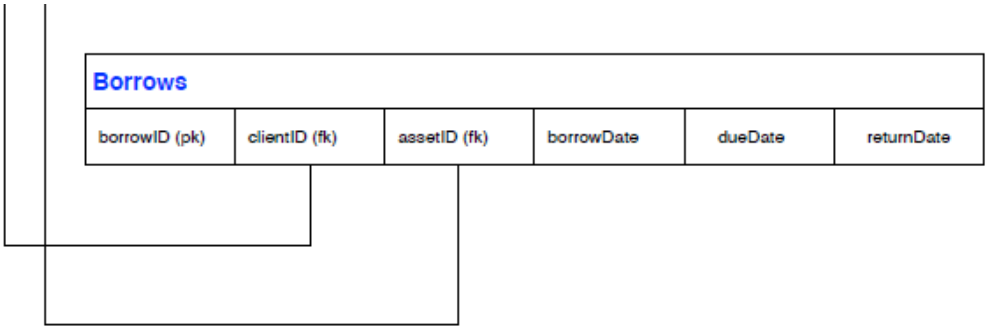


Created by Lucas Guimaraes,
Phung Vuong and Constance Wohlford
Dec 12, 2014

Relational Schema with keys




Assets
assetID (pk)
assetName
assetDescription
assetDepartment
assetClub
assetLocation
assetCondition
Projectors
assetID (jk)
ioDescription
Peripherals
assetID (fk)
pDescription
Conventional
assetID (fk)
Author
publisher
date
edition
Magazines
clientID (fk)
Volume
Date
Video
clientID (fk)
mediaType
genre



Data Dictionary

Client

clientID	Char of 9 positions
clientName	Char of 30
clientNotes	Text variable length

(Subtypes)

Faculty

clientID	Char of 9 positions
roomNumber	Char of 10

Staff

clientID	Char of 9 positions
extention	Char of 5 positions
roomNumber	Char of 10 positions

Student

clientID	Char of 9 positions
studentID	char of 12 positions

major	Char of 20 positions

Assets

assetID	Num of 4 positions
assetName	Char of 20 positions
assetDescription	Char of 40 positions
assetDepartment	Char of 20 positions
assetClub	Char of 20 positions
assetLocation	Char of 20 positions
assetCondition	Text variable length

(Subtypes)

Projectors

ioDescription	Text variable length

Peripherals

pDescriptoin	Text variable length

Conventional

Author	Char of 30 positions
publisher	Char of 20 positions
date	Date()
edition	Char of 10 position

Magazines

Volume	Char of 10 positions
Date	Date()

Video

mediaType (DVD, VHS, Reel)	Char of 10 positions
genre	Char of 20 positions

Player

mediaType	Text of variable length

Borrows

borrowID	Num of 4 positions
clientID	Char of 9 positions
assetID	Num of 4 positions
borrowDate	Date()
dueDate	Date()
returnDate	Date()

Stored Procedure: Reminder Email

If dueDate < sysDate()+2 and returnDate is null, send email to client

SQL: DDL

Create user

```
SQL>
SQL> CREATE USER library IDENTIFIED BY 1 DEFAULT TABLESPACE USERS;

User created.

SQL> GRANT DBA TO library;

Grant succeeded.

SQL>
```

Create Tables

CREATE TABLE Client (clientID VARCHAR(9) NOT NULL, clientName VARCHAR(30), clientNotes CLOB);

CREATE TABLE Faculty (clientID VARCHAR(9) NOT NULL, roomNumber VARCHAR(10));

CREATE TABLE Staff (clientID VARCHAR(9) NOT NULL, extention vARCHAR(5), roomNumber VARCHAR(10));

CREATE TABLE Student (clientID VARcHAR(9) NOT NULL, studentID VARCHAR(12), major VARCHAR(20));

CREATE TABLE Assets (assetID SMALLINT NOT NULL, assetName VARCHAR(20), assetDescription VARCHAR(40),

assetDepartment VARCHAR(20), assetClub VARCHAR(20), assetLocation VARCHAR(20),

assetCondition CLOB);

CREATE TABLE Projectors (assetID SMALLINT NOT NULL, ioDescription CLOB);

CREATE TABLE Peripherals (assetID SMALLINT NOT NULL, pDescription CLOB);

CREATE TABLE Conventional (assetID SMALLINT NOT NULL, Author VARCHAR(30), publisher VARCHAR(20),

cDate date, edition VARCHAR(10));

CREATE TABLE Magazines (assetID SMALLINT NOT NULL, Volume VARCHAR(10), mDate DATE);

CREATE TABLE Video (assetID SMALLINT NOT NULL, mediaType vARCHAR(10), genre VARCHAR(20));

CREATE TABLE Player (assetID SMALLINT NOT NULL, mediaType CLOB);

CREATE TABLE Borrows (borrowID SMALLINT NOT NULL, clientID VARCHAR(9), assetID SMALLINT,

borrowDate DATE, dueDate DATE, returnDate DATE);

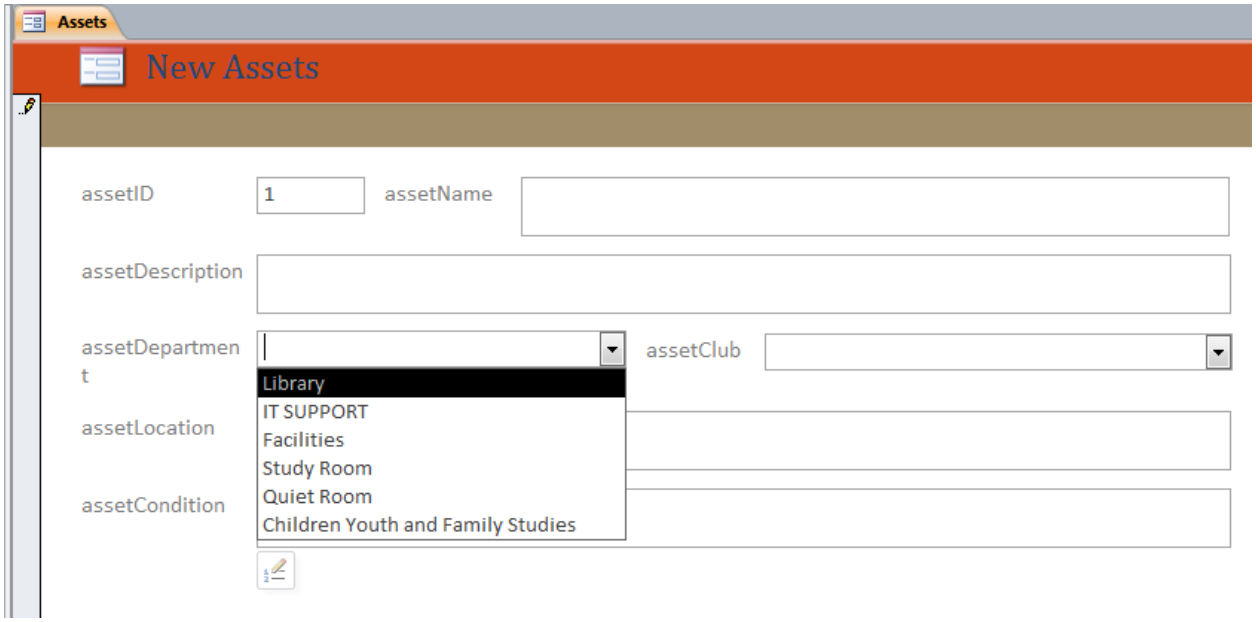
SQL: DML

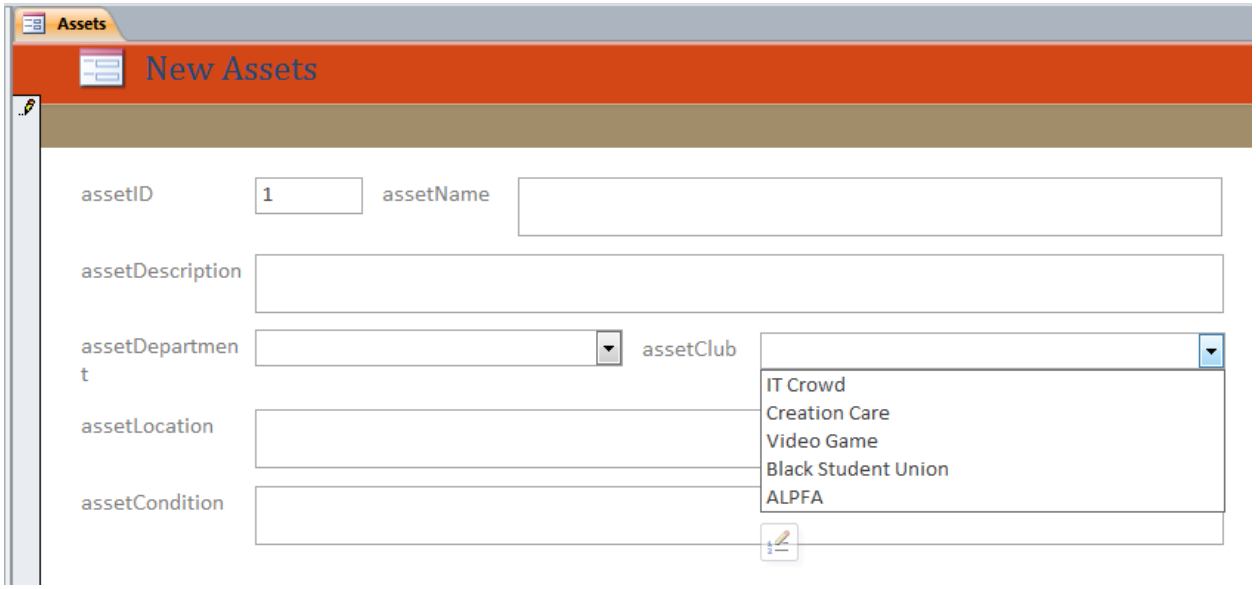
Insert Rows

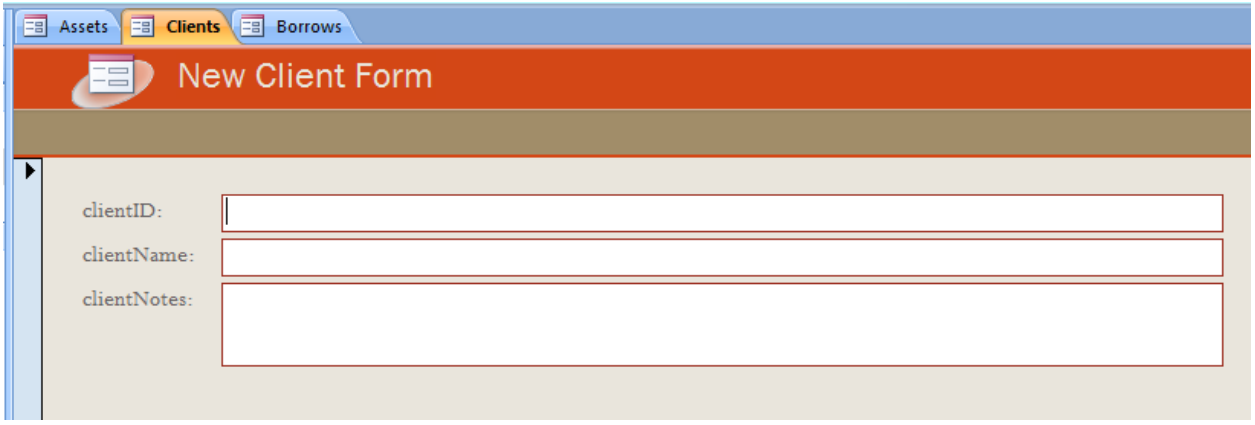
INSERT INTO Client VALUES ('TLC001', 'Connie Wohlford', NULL);
INSERT INTO Client VALUES ('TLC002', 'Phung Vuong', 'Graduating Spr 2015');
INSERT INTO Client VALUES ('TLC003', 'Lucas Guimaraes', NULL);
INSERT INTO Client VALUES ('TLC004', 'Keyre Figueroa', NULL);

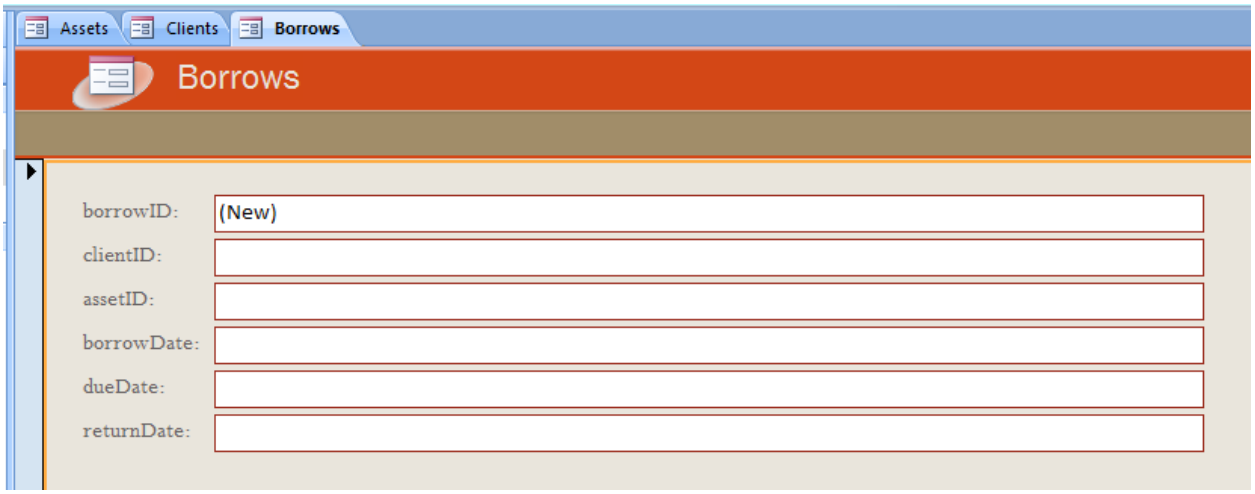
```
INSERT INTO Client VALUES ('TLC005', 'Diamond Atkins', NULL);
INSERT INTO Client VALUES ('TLC006', 'Misael Salmeron', NULL);
INSERT INTO Client VALUES ('TLC007', 'Mario Guimaraes', NULL);
INSERT INTO Client VALUES ('TLC008', 'Mark Barnum', NULL);
INSERT INTO Client VALUES ('TLC009', 'Anne Reinisch', NULL);

INSERT INTO Faculty VALUES ('TLC007', '440');
INSERT INTO Faculty VALUES ('TLC008', '435');
INSERT INTO Staff VALUES ('TLC009', '5555', '310');
INSERT INTO Student VALUES ('TLC001', 'WO00', 'CIS');
INSERT INTO Student VALUES ('TLC002', 'VU00', 'CIS');
INSERT INTO Student VALUES ('TLC003', '201400', 'CIS');
INSERT INTO Student VALUES ('TLC004', 'FI00', 'CIS');
INSERT INTO Student VALUES ('TLC005', 'AT00', 'CIS');
INSERT INTO Student VALUES ('TLC006', 'SA00', 'CIS');

INSERT INTO Assets VALUES (1001, 'Eagle Eye', NULL, NULL, 'IT CROWD', '335', 'like new');
INSERT INTO Assets VALUES (1002, 'Projector 1', 'Portable projector', 'Library', NULL, 'Library',
    'good');
INSERT INTO Assets VALUES (1003, 'VGA Cable 1', 'VGA F to F, 15F', 'Library', NULL, 'Library',
    'good');
INSERT INTO Assets VALUES (1004, 'DVD VHS Player 1', 'DVD and VHS Player', 'IT SUPPORT',
    NULL, 'IT SUPPORT', 'dvd works, not sure about VHS');
INSERT INTO Video VALUES (1001, 'DVD', 'Fiction');
INSERT INTO Projectors VALUES (1002, 'VGA M, HDMI');
INSERT INTO Peripherals VALUES (1003, VGA F to F, 15F');
INSERT INTO Player VALUES (1004, 'DVD, VHS');

INSERT INTO Borrows VALUES (1001, 'TLC001', 1003, TO_DATE('12012014', 'mmddyyyy'),
    TO_DATE('12032014', 'mmddyyyy'), NULL);
INSERT INTO Borrows VALUES (1002, 'TLC002', 1004, TO_DATE('12012014', 'mmddyyyy'),
    TO_DATE('12052014', 'mmddyyyy'), TO_DATE('12042014', 'mmddyyyy'));
INSERT INTO Borrows VALUES (1003, 'TLC003', 1001, TO_DATE('12022014', 'mmddyyyy'),
    TO_DATE('12122014', 'mmddyyyy'), NULL);
INSERT INTO Borrows VALUES (1004, 'TLC004', 1002, TO_DATE('12062014', 'mmddyyyy'),
    TO_DATE('12092014', 'mmddyyyy'), NULL);
INSERT INTO Borrows VALUES (1005, 'TLC005', 1004, TO_DATE('12072014', 'mmddyyyy'),
    TO_DATE('12312014', 'mmddyyyy'), NULL);
```

List of Forms









CRUD Matrix (Forms x Tables)

	Client table	Asset Table	Borrows table
New CLIENT form	CRU		
New Asset form		CRU	
Borrows form	R	R	CRU

SQL: Queries (Statements and results)

```
Create view unreturned (clientName, assetName, duedate) AS
    SELECT clientName, AB.assetName, dueDate
    FROM client,
    (SELECT assetName, borrows.clientID, borrows.assetID, borrows.dueDate FROM assets, borrows
```

```
WHERE assets.assetID = borrows.assetID and borrows.returnDate IS Null) AB
WHERE client.clientID=AB.clientID;
```

```
SQL> Create view unreturned (clientName, assetName, duedate) AS
  2  SELECT clientName, AB.assetName, dueDate
  3  FROM client,
  4  (SELECT assetName, borrows.clientID, borrows.assetID, borrows.dueDate FROM
assets, borrows
  5  WHERE assets.assetID = borrows.assetID and borrows.returnDate IS Null) AB
  6  WHERE client.clientID=AB.clientID;

View created.

SQL> select * from unreturned;

CLIENTNAME                     ASSETNAME          DUEDATE
------------------------------ ------------------ ---------
Connie Wohlford                VGA Cable 1        03-DEC-14
Lucas Guimaraes                Eagle Eye          12-DEC-14
Keyre Figueroa                 Projector 1        09-DEC-14
Diamond Atkins                 DVD VHS Player 1   31-DEC-14
```

```
CREATE VIEW dueSoon (clientName, assetName, dueDate) AS
    SELECT clientName, assetName, dueDate
    FROM client, assets, borrows
    WHERE borrows.returnDate is NULL AND dueDate < (SYSDATE+4)
    AND client.clientID=borrows.clientID AND assets.assetID=borrows.assetID;
```

```
SQL> CREATE VIEW dueSoon (clientName, assetName, dueDate) AS
  2  SELECT clientName, assetName, dueDate
  3  FROM client, assets, borrows
  4  WHERE borrows.returnDate is NULL AND dueDate < (SYSDATE+4)
  5  AND client.clientID=borrows.clientID AND assets.assetID=borrows.assetID;

View created.

SQL> SELECT * FROM dueSoon;

CLIENTNAME                     ASSETNAME          DUEDATE
------------------------------ ------------------ ---------
Connie Wohlford                VGA Cable 1        03-DEC-14
Lucas Guimaraes                Eagle Eye          12-DEC-14
Keyre Figueroa                 Projector 1        09-DEC-14
```

SQL: Triggers, Stored Procedures, and Functions

This function returns the number of instances of a given item that are unreturned. Ideally this should be 0 or 1.

```
DROP FUNCTION numAlreadyCheckedOut;
    CREATE FUNCTION numAlreadyCheckedOut(assetIDtoCheck IN SMALLINT)
    RETURN number
    IS
    borrowed number:=0;
    BEGIN
```

```
                SELECT count(borrowID) INTO borrowed FROM borrows WHERE
                assetIDtoCheck = assetID and returnDate is NULL;
                Return borrowed;
        END;
/
```

This trigger returns an error if the number already checked out for a certain asset is more than zero. Otherwise it lets the asset be checked out again.

```
CREATE OR REPLACE TRIGGER AlreadyCheckedOut
        BEFORE INSERT ON borrows
        for each row
        BEGIN
                IF (numAlreadyChecked(:new.assetID) > 0) THEN
                        RAISE_APPLICATION_ERROR(-1, 'Already checked out');
                END IF;
        END;
/
```

Conclusion:

The next step in this project is to implement this database with some real data and do a trial run.