

LIBRARY MANAGEMENT SYSTEM

Submitted to Dr.Kim
CS 157A Project
Fall 2015

Submitted by:

Aishwarya Venkateswaran

Jordan Petersen

Brendon Yim

Table of Contents

[Features and Dataset Info](#)

[Database schema](#)

[Screenshots of all relations after populating initial data](#)

[List of 18 distinct functions excluding functions done by DBA](#)

[All SQL select statements](#)

[All SQL update statements](#)

[All SQL delete statements](#)

[All SQL insert statements](#)

[All SQL triggers](#)

[All SQL stored procedures](#)

[Screenshots to demonstrate the functionality](#)

Our group created a library management system [with GUI] that provides the following features to users:

All Features:

1. User shall sign in [Will need sql query] (by userID)
2. Admin shall be able to add new user:
3. Admin shall be able to delete a user:
4. Users shall be able to update information. [Preferred branch]
5. Users shall be able to search for a book using the title, author or category. Search can be done in any combination i.e. only one or two or three of these parameters can be given. Users should be able to search this in combination.
6. User shall be able to rent a new book (Display books on for the library).
7. Users shall be able to return a book.
8. Admins shall be able to see all distinct users who have rented a book.
9. Users can rate a book.
10. Users can update the rating for a book later.
11. Users shall be able to see the most popular books (top 5)
12. Users shall be able to see the highest rated books (top 5)
13. Admin shall be able to retrieve users who have several [two or more] overdue books.
14. Admins can find the total number of loans made at a library.
15. Admin can find users who have given items a higher rating the second time he/she rated it.
16. Users can find other books that have the same author as a given book.
17. Admin can see users who have not loaned any books.
18. Users can view all items that are of audio/video media [including e-books].

Dataset:

Data for the Library Branch was obtained from data.gov It is the actual of New York Brooklynn library. The data set has been reduced.

Data for the Item table was obtained from data.gov. It is the actual library catalog data of the New York Brooklyn library. The data data set has been reduced significantly.

Data was obtained in .csv/.xml files. Then, python scripts and java were used to read the files, clean the data and then insert the data to a database.

<http://catalog.data.gov/dataset/bpl-branches-9e1f2>

<http://catalog.data.gov/dataset/brooklyn-public-library-catalog-f85b8>

Readme for running the program:

Please run the ddl, followed by the triggers, followed by the actual insertions of data. Then, go to the java program, set up the password and add jdbc connector to the runtime library, and finally start the program. Stored procedures [including archiving] work from DBA side. Two of the triggers are invoked by our java-based functionalities [rent a book and return a book]. The other trigger can be tested by the DBA from MySQL.

DATABASE SCHEMA IN THE FORM OF SQL CREATE TABLE

Librarybranch table: Primary Key (LibraryBranchID)

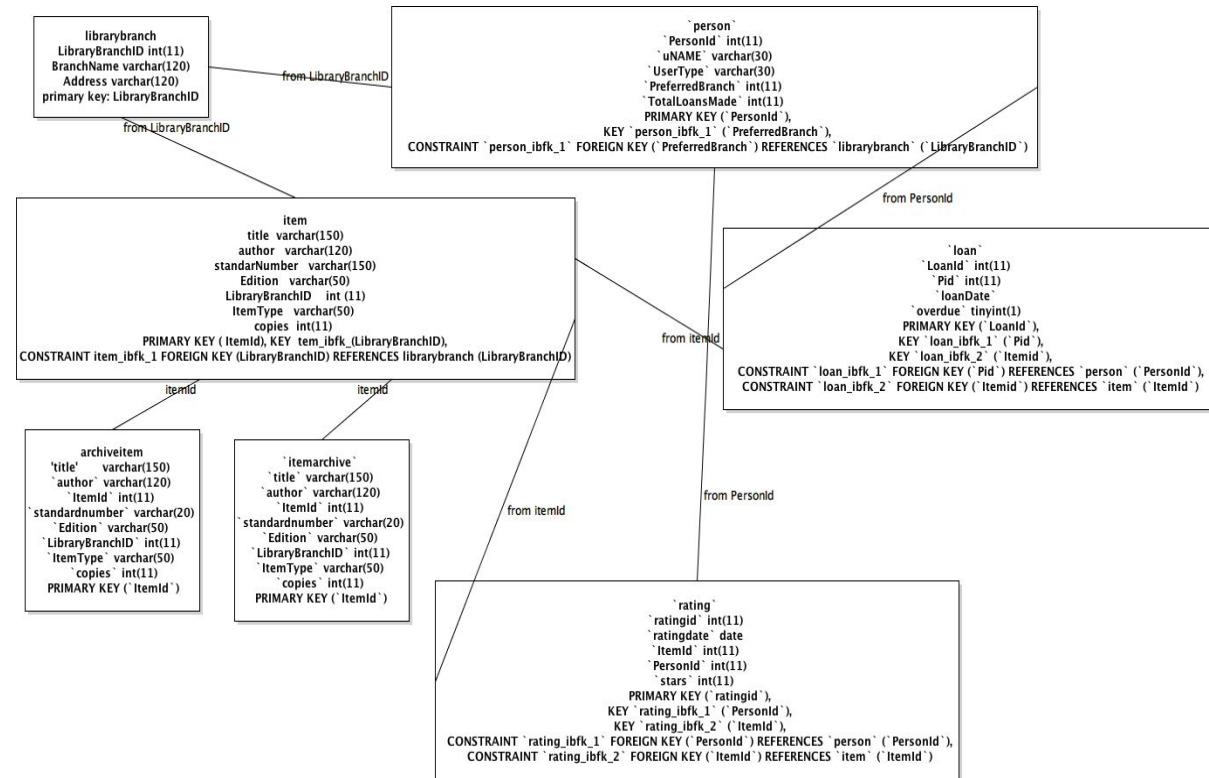
Person table: PRIMARY KEY ('PersonId'), KEY `person_ibfk_1` ('PreferredBranch'),
 CONSTRAINT `person_ibfk_1` FOREIGN KEY ('PreferredBranch') REFERENCES `librarybranch` ('LibraryBranchID')

Item table: PRIMARY KEY ('ItemId'), KEY `item_ibfk_1` ('LibraryBranchID'),
 CONSTRAINT `item_ibfk_1` FOREIGN KEY ('LibraryBranchID') REFERENCES `librarybranch` ('LibraryBranchID')

Archiveitem and itemarchive table: PRIMARY KEY ('ItemId')

Loan table: PRIMARY KEY ('LoanId'),
 KEY `loan_ibfk_1` ('Pid'), KEY `loan_ibfk_2` ('ItemId'),
 CONSTRAINT `loan_ibfk_1` FOREIGN KEY ('Pid') REFERENCES `person` ('PersonId'),
 CONSTRAINT `loan_ibfk_2` FOREIGN KEY ('ItemId') REFERENCES `item` ('ItemId')

Rating table: PRIMARY KEY ('ratingid'),
 KEY `rating_ibfk_1` ('PersonId'),
 KEY `rating_ibfk_2` ('ItemId'),
 CONSTRAINT `rating_ibfk_1` FOREIGN KEY ('PersonId') REFERENCES `person` ('PersonId'),
 CONSTRAINT `rating_ibfk_2` FOREIGN KEY ('ItemId') REFERENCES `item` ('ItemId')



Schema Code

```

drop database library;
CREATE DATABASE IF NOT EXISTS `library` ;
USE `library`;

CREATE TABLE `librarybranch` (
`LibraryBranchID` int(11) NOT NULL AUTO_INCREMENT,
`BranchName` varchar(120) DEFAULT NULL,
`Address` varchar(120) DEFAULT NULL,
PRIMARY KEY (`LibraryBranchID`)
);

CREATE TABLE `item` (
`title` varchar(150) DEFAULT NULL,
`author` varchar(120) DEFAULT NULL,
`ItemId` int(11) NOT NULL AUTO_INCREMENT,
`standardNumber` varchar(150) DEFAULT '-1',
`Edition` varchar(50) DEFAULT NULL,
`LibraryBranchID` int(11) DEFAULT NULL,
`ItemType` varchar(50) DEFAULT NULL,
`copies` int(11) DEFAULT NULL,
`updatedAt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`ItemId`),
KEY `item_ibfk_1`(`LibraryBranchID`),
CONSTRAINT `item_ibfk_1` FOREIGN KEY (`LibraryBranchID`) REFERENCES `librarybranch`
(`LibraryBranchID`)
);

-- Archive item is used for archiving the item table
CREATE TABLE `archiveitem` (
`title` varchar(150) DEFAULT NULL,
`author` varchar(120) DEFAULT NULL,
`ItemId` int(11) NOT NULL AUTO_INCREMENT,
`standardnumber` varchar(20) DEFAULT NULL,
`Edition` varchar(50) DEFAULT NULL,
`LibraryBranchID` int(11) DEFAULT NULL,
`ItemType` varchar(50) DEFAULT NULL,
`copies` int(11) DEFAULT NULL,
`updatedAt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`ItemId`)
);

CREATE TABLE `person` (
`PersonId` int(11) NOT NULL AUTO_INCREMENT,
`uNAME` varchar(30) NOT NULL,
`UserType` varchar(30) DEFAULT NULL,
`PreferredBranch` int(11) DEFAULT NULL,

```

```

`TotalLoansMade` int(11) DEFAULT NULL,
PRIMARY KEY (`PersonId`),
KEY `person_ibfk_1` (`PreferredBranch`),
CONSTRAINT `person_ibfk_1` FOREIGN KEY (`PreferredBranch`) REFERENCES
`librarybranch` (`LibraryBranchID`)
);

```

```

CREATE TABLE `loan` (
`LoanId` int(11) NOT NULL AUTO_INCREMENT,
`Pid` int(11) DEFAULT NULL,
`Itemid` int(11) DEFAULT NULL,
`loanDate` date DEFAULT '0000-00-00',
`overdue` tinyint(1) DEFAULT '0',
PRIMARY KEY (`LoanId`),
KEY `loan_ibfk_1` (`Pid`),
KEY `loan_ibfk_2` (`Itemid`),
CONSTRAINT `loan_ibfk_1` FOREIGN KEY (`Pid`) REFERENCES `person` (`PersonId`),
CONSTRAINT `loan_ibfk_2` FOREIGN KEY (`Itemid`) REFERENCES `item` (`ItemId`)
);

```

```

CREATE TABLE `rating` (
`ratingid` int(11) NOT NULL AUTO_INCREMENT,
`ratingdate` date DEFAULT NULL,
`ItemId` int(11) DEFAULT NULL,
`PersonId` int(11) DEFAULT NULL,
`stars` int(11) DEFAULT NULL,
PRIMARY KEY (`ratingid`),
KEY `rating_ibfk_1` (`PersonId`),
KEY `rating_ibfk_2` (`ItemId`),
CONSTRAINT `rating_ibfk_1` FOREIGN KEY (`PersonId`) REFERENCES `person` (`PersonId`),
CONSTRAINT `rating_ibfk_2` FOREIGN KEY (`ItemId`) REFERENCES `item` (`ItemId`)
);

```

SCREENSHOTS OF ALL RELATIONS AFTER POPULATING INITIAL DATA

Library Branch Table

LibraryBranchID	BranchName	Address
1	Arlington Library	203 Arlington Ave. at Warwick St. Brooklyn, NY11207
2	Bay Ridge Library	7223 Ridge Blvd. at 73rd St. Brooklyn, NY11209
3	Bedford Learning Center	496 Franklin Avenue Brooklyn, NY11238
4	Bedford Library	496 Franklin Avenue Brooklyn, NY11238
5	Borough Park Library	1265 43rd St. at 13th Ave. Brooklyn, NY11219
6	Brighton Beach Library	16 Brighton First Rd. at Brighton Beach Ave. Brooklyn, NY11235
7	Brooklyn Heights Library	280 Cadman Plaza West at Tillary St. Brooklyn, NY11201
8	Brower Park Library	725 St. Marks Ave. at Nostrand Ave. Brooklyn, NY11216
9	Brownsville Library	61 Glenmore Ave. at Watkins St. Brooklyn, NY11212
10	Bushwick Library	340 Bushwick Ave. at Seigel St. Brooklyn, NY11206

Item Table

title	author	itemId	standardNumber	Edition	LibraryBranchID	itemType	copies	updatedAt
Quién ayuda en casa? / Ricardo Al...	Alcántara, Ricardo.	1	8426319173	8a ed.	3	FICTION SPANISH	1	2015-12-02 21:17:48
Alien, Paul Marshall		2	1		10	BOOK	1	2015-12-02 21:17:48
Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	3	NULL	1st ed.	9	BOOK	2	2015-12-02 21:17:48
Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	4	NULL	1st ed.	2	BOOK	3	2015-12-02 21:17:48
Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	5	NULL	1st ed.	1	BOOK	3	2015-12-02 21:17:48
Home town	Amory, Cleveland.	6	NULL	[1st ed.]	6	FICTION	3	2015-12-02 21:17:48
Home town	Amory, Cleveland.	7	NULL	[1st ed.]	8	FICTION	3	2015-12-02 21:17:48
Mid-east: world-center, yesterday...	Anshen, Ruth Nanda.	8	NULL	[1st ed.]	10	FICTION	1	2015-12-02 21:17:48
Mid-east: world-center, yesterday...	Anshen, Ruth Nanda.	9	NULL	[1st ed.]	2	BOOK	3	2015-12-02 21:17:48
Wo shi shui? / pianxie Anxin hult... Anxin.		11	9620818199	Chu ban.	8	CHINESE	3	2015-12-02 21:17:48
Wo shi shui? / pianxie Anxin hult... Anxin.		12	9620818199	Chu ban.	2	CHINESE	3	2015-12-02 21:17:48
Wo shi shui? / pianxie Anxin hult... Anxin.		13	9620818199	Chu ban.	3	CHINESE	2	2015-12-02 21:17:48
The three lives of Harriet Hubbard...	Ayer, Margaret Hubbard.	14	NULL	[1st ed.]	5	BOOK	3	2015-12-02 21:17:48
The three lives of Harriet Hubbard...	Ayer, Margaret Hubbard.	15	NULL	[1st ed.]	7	BOOK	2	2015-12-02 21:17:48
Canapés for the kitties / Marian B...	Babson, Marian.	16	1568955227	Softcover large print ed.	8	FICTION	2	2015-12-02 21:17:48
Economic dynamics an introduc...	Baumol, William J.	17	NULL	3rd ed.	2	BOOK	3	2015-12-02 21:17:48
Economic dynamics an introduc...	Baumol, William J.	18	NULL	3rd ed.	7	BOOK	2	2015-12-02 21:17:48
Good bye, Ava	Bissell, Richard Pike.	19	NULL	[1st ed.]	2	FICTION	2	2015-12-02 21:17:48
Good bye, Ava	Bissell, Richard Pike.	20	NULL	[1st ed.]	4	FICTION	2	2015-12-02 21:17:48
Good bye, Ava	Good bye, Ava [Richard Pike.	21	NULL	[1st ed.]	5	FICTION	3	2015-12-02 21:17:48
The warrior saint	Boiley, R. V. C. (Ronald Victor Cou...	22	NULL	[1st ed.]	9	BOOK	1	2015-12-02 21:17:48
The warrior saint	Boiley, R. V. C. (Ronald Victor Cou...	23	NULL	[1st ed.]	4	BOOK	2	2015-12-02 21:17:48
The warrior saint	Boiley, R. V. C. (Ronald Victor Cou...	24	NULL	[1st ed.]	1	BOOK	2	2015-12-02 21:17:48
The community college	Bogue, Jesse Parker, 1889-1960.	25	NULL	1st ed.	8	BOOK	1	2015-12-02 21:17:48
T. E. Lawrence / Peter Brent intro...	Brent, Peter Ludwig.	26	399115846	1st American ed.	2	BOOK	2	2015-12-02 21:17:48
T. E. Lawrence / Peter Brent intro...	Brent, Peter Ludwig.	27	399115846	1st American ed.	5	BOOK	2	2015-12-02 21:17:48
Before barbed wire / by Mark H. B...	Brown, Mark Herbert, 1900-	28	NULL	[1st ed.]	1	SCIENCE	3	2015-12-02 21:17:48
Before barbed wire / by Mark H. B...	Brown, Mark Herbert, 1900-	29	NULL	[1st ed.]	10	SCIENCE	1	2015-12-02 21:17:48
Men, women, and other anomalies...	Broyard, Anatole.	30	416005314	1st American ed.	9	BOOK	2	2015-12-02 21:17:48
Report from part one. Prefaces by...	Brooks, Gwendolyn, 1917-2000.	31	910296820	[1st ed.]	5	BOOK	1	2015-12-02 21:17:48
Report from part one. Prefaces by...	Brooks, Gwendolyn, 1917-2000.	32	910296820	[1st ed.]	9	BOOK	2	2015-12-02 21:17:48
Violent summer / H.B. Broome	Broome, H. B.	33	385265670	1st ed.	2	FICTION	1	2015-12-02 21:17:48
Violent summer / H.B. Broome	Broome, H. B.	34	385265670	1st ed.	4	FICTION	3	2015-12-02 21:17:48
Violent summer / H.B. Broome	Broome, H. B.	35	385265670	1st ed.	9	FICTION	3	2015-12-02 21:17:48
A choice of attitudes	Burr, Gray.	36	NULL	[1st ed.]	6	BOOK	1	2015-12-02 21:17:48
The affairs of Flavie (Les Héritiers...	Chevallier, Gabriel, 1895-	37	NULL	[1st American ed.]	3	FICTION	2	2015-12-02 21:17:48
Pien se mao / Cheng Yün-ch'in hs...	Cheng, Yün-ch'in.	38	7532025411	Ti 1 pan.	1	CHINESE	1	2015-12-02 21:17:48
Fieldiana: zoology memoirs	Chicago Natural History Museum.	39	NULL	v. 1- Feb. 28	2	BOOK	3	2015-12-02 21:17:48
Fieldiana: zoology memoirs	Chicago Natural History Museum.	40	NULL	v. 1- Feb. 28	3	BOOK	3	2015-12-02 21:17:48
A seed was sown the life, philoso...	Clancy, Robert, 1904-	41	NULL	[2d ed.]	3	BOOK	2	2015-12-02 21:17:48
Systems of land tenure in various...	Cobden Club (London, England)	42	NULL	New ed.	3	BOOK	2	2015-12-02 21:17:48
Ca and the bird	Cronwell, Geoffrey, 1919-	43	NULL	[1st American ed.]	2	FICTION	2	2015-12-02 21:17:48

Archiveitem Table [After running the stored procedure to archive the data in table Item]

	title	author	itemid	standardnumber	edition	librarybranchid	itemtype	copies	updatedat
archiveitem	Quién ayuda en casa? / Ricardo Al...	Alcántara, Ricardo.	1	8426319173	8a ed.	3	FICTION SPANISH	1	2015-12-02 21:17:48
item	Allen, Paul Marshall	4308 A	2	1	A Christian Rosenkreutz anthology	10	BOOK	1	2015-12-02 21:17:48
itemarchive	Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	3	NULL	1st ed.	9	BOOK	2	2015-12-02 21:17:48
librarybranch	Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	4	NULL	1st ed.	2	BOOK	3	2015-12-02 21:17:48
loan	Venezuela, a democracy	Allen, Henry Justin, 1868-1950.	5	NULL	1st ed.	1	BOOK	3	2015-12-02 21:17:48
person	Venezuela, a democracy	Amory, Cleveland.	6	NULL	[1st ed.]	6	FICTION	3	2015-12-02 21:17:48
rating	Home town	Amory, Cleveland.	7	NULL	[1st ed.]	8	FICTION	3	2015-12-02 21:17:48
PROC & FUNCS	Home town	Amory, Cleveland.	8	NULL	[1st ed.]	10	FICTION	1	2015-12-02 21:17:48
CopyToArchive	Home town	Anshen, Ruth Nanda.	9	NULL	[1st ed.]	2	BOOK	3	2015-12-02 21:17:48
FindNumberOfCopies	Mid-east: world-center, yesterday...	Anshen, Ruth Nanda.	10	NULL	[1st ed.]	1	BOOK	1	2015-12-02 21:17:48
	Wo shi shui? / pianxie Anxin huit...	Anxin.	11	9620818199	Chu ban.	8	CHINESE	3	2015-12-02 21:17:48
	Wo shi shui? / pianxie Anxin huit...	Anxin.	12	9620818199	Chu ban.	2	CHINESE	3	2015-12-02 21:17:48
	Wo shi shui? / pianxie Anxin huit...	Anxin.	13	9620818199	Chu ban.	3	CHINESE	2	2015-12-02 21:17:48
	The three lives of Harriet Hubbard...	Ayer, Margaret Hubbard.	14	NULL	[1st ed.]	5	BOOK	3	2015-12-02 21:17:48
	The three lives of Harriet Hubbard...	Ayer, Margaret Hubbard.	15	NULL	[1st ed.]	7	BOOK	2	2015-12-02 21:17:48
	Canapés für die Kitties / Marian B...	Babson, Marian.	16	1568955227	Softcover large print ed.	8	FICTION	2	2015-12-02 21:17:48
	Economic dynamics: an introduc...	Baumol, William J.	17	NULL	3rd ed.	2	BOOK	3	2015-12-02 21:17:48
	Economic dynamics: an introduc...	Baumol, William J.	18	NULL	3rd ed.	7	BOOK	2	2015-12-02 21:17:48
	Good bye, Ava	Bissell, Richard Pike.	19	NULL	[1st ed.]	2	FICTION	2	2015-12-02 21:17:48
	Good bye, Ava	Bissell, Richard Pike.	20	NULL	[1st ed.]	4	FICTION	2	2015-12-02 21:17:48
	Good bye, Ava	Bissell, Richard Pike.	21	NULL	[1st ed.]	5	FICTION	3	2015-12-02 21:17:48
	The warrior saint	Bodley, R. V. C. (Ronald Victor Cou...	22	NULL	[1st ed.]	9	BOOK	1	2015-12-02 21:17:48
	The warrior saint	Bodley, R. V. C. (Ronald Victor Cou...	23	NULL	[1st ed.]	4	BOOK	2	2015-12-02 21:17:48
	The warrior saint	Bodley, R. V. C. (Ronald Victor Cou...	24	NULL	[1st ed.]	1	BOOK	2	2015-12-02 21:17:48
	The community college	Bogue, Jesse Parker, 1889-1960.	25	NULL	1st ed.	8	BOOK	1	2015-12-02 21:17:48
	T. E. Lawrence / Peter Brent intro...	Brent, Peter Ludwig.	26	399115846	1st American ed.	2	BOOK	2	2015-12-02 21:17:48
	T. E. Lawrence / Peter Brent intro...	Brent, Peter Ludwig.	27	399115846	1st American ed.	5	BOOK	2	2015-12-02 21:17:48
	Before barbed wire / by Mark H. B...	Brown, Mark Herbert, 1900-	28	NULL	[1st ed.]	1	SCIENCE	3	2015-12-02 21:17:48
	Before barbed wire / by Mark H. B...	Brown, Mark Herbert, 1900-	29	NULL	[1st ed.]	10	SCIENCE	1	2015-12-02 21:17:48
	Men, women, and other anticlima...	Broyard, Anatole.	30	416005314	1st American ed.	9	BOOK	2	2015-12-02 21:17:48
	Report from part one. Prefaces by...	Brooks, Gwendolyn, 1917-2000.	31	910296820	[1st ed.]	5	BOOK	1	2015-12-02 21:17:48
	Report from part one. Prefaces by...	Brooks, Gwendolyn, 1917-2000.	32	910296820	[1st ed.]	9	BOOK	2	2015-12-02 21:17:48
	Violent summer / H.B. Broome	Broome, H. B.	33	385265670	1st ed.	2	FICTION	1	2015-12-02 21:17:48
	Violent summer / H.B. Broome	Broome, H. B.	34	385265670	1st ed.	4	FICTION	3	2015-12-02 21:17:48
	Violent summer / H.B. Broome	Broome, H. B.	35	385265670	1st ed.	9	FICTION	3	2015-12-02 21:17:48
	A choice of attitudes	Burr, Gray.	36	NULL	[1st ed.]	6	BOOK	1	2015-12-02 21:17:48
	The affairs of Flavie (Les Héritiers...	Chevallier, Gabriel, 1895-	37	NULL	[1st American ed.]	3	FICTION	2	2015-12-02 21:17:48
	Pien se mao / Cheng Yün-ch'in hs...	Cheng, Yün-ch'in.	38	7532025411	Tl 1 pan.	1	CHINESE	1	2015-12-02 21:17:48
	Fieldiana: zoology memoirs	Chicago Natural History Museum.	39	NULL	v. 1-Feb. 28	2	BOOK	3	2015-12-02 21:17:48
	Fieldiana: zoology memoirs	Chicago Natural History Museum.	40	NULL	v. 1-Feb. 28	3	BOOK	3	2015-12-02 21:17:48
	A seed was sown: the life, philoso...	Clancy, Robert, 1904-	41	NULL	[2d ed.]	3	BOOK	2	2015-12-02 21:17:48
	Systems of land tenure in various...	Cobden Club (London, England)	42	NULL	New ed.	3	BOOK	2	2015-12-02 21:17:48
TABLE INFORMATION	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15	Created: 12/2/15
	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB	Engine: InnoDB
	Rows: 251	Rows: 251	Rows: 251	Rows: 251	Rows: 251	Rows: 251	Rows: 251	Rows: 251	Rows: 251
	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB	Size: 64.0 KiB
	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1	Encoding: latin1
	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256	Auto-increment: 256

Loan Table

	loanid	pid	itemid	loanDate	overdue
archiveitem	1	1	2	2015-11-02	0
item	2	1	31	2015-10-05	1
itemarchive	3	2	20	2015-09-03	1
librarybranch	4	3	19	2015-11-27	0
loan	5	4	17	2015-10-19	1
person	6	6	14	2015-09-18	0
rating	7	7	13	2015-11-17	1
PROC & FUNCS	8	9	12	2015-10-05	0
CopyToArchive	9	10	12	2015-09-12	1
FindNumberOfCopies	10	2	10	2015-11-01	0
	11	3	3	2015-10-10	1
	12	3	41	2015-11-14	0
	13	4	44	2015-11-14	0
	14	5	45	2015-11-14	0
	15	8	41	2015-11-14	0
	16	10	41	2015-11-14	0
	17	5	47	2015-11-14	0
	18	5	48	2015-11-14	0
	19	1	49	2015-11-14	0
	20	4	50	2015-11-14	0
	21	10	61	2015-11-14	0
	22	2	2	2015-12-02	1

Person Table

LIBRARY Select Database

Structure Content Relations Triggers Table Info Query

TABLES

- archiveitem
- item
- itemarchive
- librarybranch
- loan
- person**
- rating

PROCS & FUNCS

- CopyToArchive
- FindNumberOfCopies

Search: PersonId =

PersonId	uNAME	UserType	PreferredBranch	TotalLoansMade
1	Amy	A	2	0
2	Anne	U	1	0
3	Ben	A	2	0
4	Cody	U	3	0
5	Brandon	A	5	0
6	Jordan	U	6	0
7	Nathan	A	7	0
8	Alicia	U	8	0
9	Bill	U	9	0
10	Jake	A	10	0

Rating Table

LIBRARY Select Database

Structure Content Relations Triggers Table Info Query

TABLES

- archiveitem
- item
- itemarchive
- librarybranch
- loan
- person
- rating**
- rating

PROCS & FUNCS

- CopyToArchive
- FindNumberOfCopies

Search: ratingid =

ratingid	ratingdate	ItemId	PersonId	stars
1	2015-04-03	2	1	5
2	2014-12-13	3	2	4
3	2015-01-23	10	2	5
4	2013-04-05	12	3	2
5	2015-02-13	40	4	5
6	2012-08-23	50	5	4
7	2015-05-03	71	6	3
8	2014-01-10	81	7	1
9	2011-03-09	41	8	2
10	2010-04-07	31	9	5
11	2015-08-27	51	10	4
12	2013-07-13	61	9	4
13	2012-06-09	50	8	3
14	2015-04-02	2	1	4
15	2015-04-02	3	2	5
16	2015-04-02	4	2	5
17	2015-08-02	4	2	5
18	2015-08-03	5	2	5

LIST OF AT 18 DISTINCT FEATURES

All SQL select statements

-- 1. User shall sign in (by userID)

Select PersonID, uName, UserType, PreferredBranch From person Where PersonID = ?;

-- 5. Users shall be able to search for a book using the title, author or category. Search can be done in any combination i.e. only one or two or three of these parameters can be given. Users should be able to search this in combination.

*SELECT * FROM item WHERE title Like '% ? %' and author = ? and itemType =?;*

-- 8. Admins shall be able to see all distinct users who have rented a book.

select distinct title , uNAME

from person join loan on person.PersonId = loan.pid join item on Loan.itemid = Item.itemid

where UserType ='U'" + "order by title;

-- 11. Users shall be able to see the most popular books (top 5).

select title,count() total*

from loan join item on Loan.itemid=Item.itemid

where ItemType='BOOK'

group by title,author,edition

order by total desc

Limit 5;

-- 12. Users shall be able to see the highest rated books (top 5).

select title,avg(stars)

from rating join item on rating.itemid=Item.itemid

where ItemType='Book'

group by title,author,edition

order by avg(stars) desc

limit 5;

-- 13.Admin shall be able to retrieve users who have several [two or more] overdue books.

select PersonId,uname,count() from Person join Loan on Person.PersonId=Loan.Pid*

where overdue = true group by personid having count()>=2;*

-- 14. Admins can find the total number of loans made at a library.

select count()*

from loan

join

item on item.itemid = loan.itemid

where libraryBranchID = (select librarybranchid from libraryBranch where branchname=?);

-- 15.Admin can find users who have given items a higher rating the second time he/she rated it.

select distinct r1.personid,i1.title,i1.author,i1.edition,r1.stars oldRating,r2.stars newRating

from rating r1 join rating r2 on r1.personid=R2.PERSONID

join Item i1 on

(select distinct title,edition,author from item where itemid=i1.itemid)

= (select distinct title,edition,author from item where itemid=r1.itemid)

join item i2 on

(select distinct title,edition,author from item where itemid=i2.itemid)

= (select distinct title,edition,author from item where itemid=r2.itemid)

where r1.ratingDate<r2.ratingDate and r1.stars<=r2.stars and i1.title=i2.title;

-- 16.Users can find other books that have the same author as a given book.

select distinct title,edition from item old where author = (select distinct author from item where title= ?);

-- 17. Admin can see users who have not loaned any books.

*SELECT * FROM PERSON left outer JOIN LOAN ON person.personid = LOAN.pid where loanid is null;*

-- 18. Users can view all items that are of audio/video media [including e-books].

select distinct title,author,edition,itemtype from item where itemtype like '%ebook%'

union

select distinct title,author,edition,itemtype from item where itemtype like '%audio%'

union

Select distinct title,author,edition,itemtype from item where itemtype like '%cd%'

union

distinct select title,author,edition,itemtype from item where itemtype like '%dvd%';

All SQL delete statements

-- 3. Admin shall be able to delete a user:

DELETE FROM person where PersonId = ?;

-- 7. Users shall be able to return a book.

update item set copies = copies +1 where item.ItemId = ?;

delete from loan where loan.LoanId = ?;

All SQL update statements

-- 4. Users shall be able to update information. [preferred branch]

update person set PreferredBranch = (select libraryBranchID from librarybranch where BranchName=?) where person.PersonId = ?;

-- 6. User shall be able to rent a new book (Display books on for the library).

update item set copies = copies -1 where item.ItemId = ?;

insert into loan SET pid = ?, ItemId = ?, loandate = ?,overdue = false ON DUPLICATE KEY UPDATE ItemId = ?, loandate = ?,overdue = false;

-- 10. Users can update the rating for a book later.

update rating set Stars = ? where RatingId = ?;

All SQL insert statements

-- 9. Users can rate a book.

insert into rating set ratingdate = ?,itemid = ?,personid = ?,stars =?;

-- 6. User shall be able to rent a new book (Display books on for the library).

update item set copies = copies -1 where item.ItemId = ?;

insert into loan SET pid = ?, ItemId = ?, loandate = ?,overdue = false ON DUPLICATE KEY UPDATE ItemId = ?, loandate = ?,overdue = false;

-- 2. Admin shall be able to add new user:

insert into person(uname,usertype,preferredbranch,totalLoansMade) values(?, ?, ?, ?);

All SQL triggers

-- When a loan is made, update the number of copies as well as the total loans made by user.

```
DROP TRIGGER IF EXISTS UpdateUserTotalLoans;
delimiter //
CREATE TRIGGER UpdateUserTotalLoans
BEFORE INSERT ON loan
FOR EACH ROW
BEGIN
UPDATE person SET TotalLoansMade = TotalLoansMade+1 WHERE Personid = new.pid;
END//
delimiter ;
```

-- When a loan is closed, update the number of copies as well as the total loans made by user.

```
DROP TRIGGER IF EXISTS UpdateUserTotalLoansOnDelete;
delimiter //
CREATE TRIGGER UpdateUserTotalLoansOnDelete
BEFORE DELETE ON loan
FOR EACH ROW
BEGIN
UPDATE person SET TotalLoansMade = TotalLoansMade-1 WHERE Personid = old.pid;
END//
delimiter ;
```

-- If a user's name has been changed to paid guy, add "5" star reviews from that person for all items.

-- The logic is that if this person has been paid to write good reviews, we just add his/her "5 star" reviews to all books.

```
DROP TRIGGER IF EXISTS SuperReviewer;
delimiter //
CREATE TRIGGER SuperReviewer
AFTER UPDATE ON person
FOR EACH ROW
BEGIN
IF (new.uname='PaidGuy') then
    insert into rating(personid,ratingdate,ItemId,stars)
    select new.personid,current_date(),itemid,5 from item;
    END IF;
END//
delimiter ;
```

All SQL stored procedures

-- Create a stored procedure that returns the total number of copies of a book our library owns, given a title A. First, find the standard number (S) of A. Then, go through the ITEM table to find tuples where standard Number= S. Then, add the number of copies together to find the total number of copies the whole library system owns.

```
DROP PROCEDURE IF EXISTS FindNumberOfCopies;
DELIMITER //
CREATE PROCEDURE FindNumberOfCopies ( IN itemTitle varchar(150), IN itemEdition
varchar(150), IN itemAuthor varchar(150), OUT result int(11))
BEGIN
select sum(copies) into result from item where title = itemTitle and edition=itemEdition and
author=itemAuthor;
END //
DELIMITER ;
```

```
all FindNumberOfCopies( 'Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result);
select @result;
```

Stored Procedure for archiving

```
DROP PROCEDURE IF EXISTS CopyToArchive;
DELIMITER //
CREATE PROCEDURE CopyToArchive ( IN cutoffDate timestamp)
BEGIN
START TRANSACTION;
insert into
archiveitem(title,author,standardNumber,edition,libraryBranchID,itemtype,copies,updatedAt)
select title,author,standardNumber,edition,libraryBranchID,itemtype,copies,updatedAt from item
where date(updatedAt)<=date(cutoffdate) ;
commit;
END //
DELIMITER ;
```

```
-- select * from archiveitem;
call copyToArchive(current_timestamp() + 1);

select * from archiveitem;
```

SCREENSHOTS TO DEMONSTRATE THE FOLLOWING FUNCTIONALITY

- 1. User shall sign in (by userID)

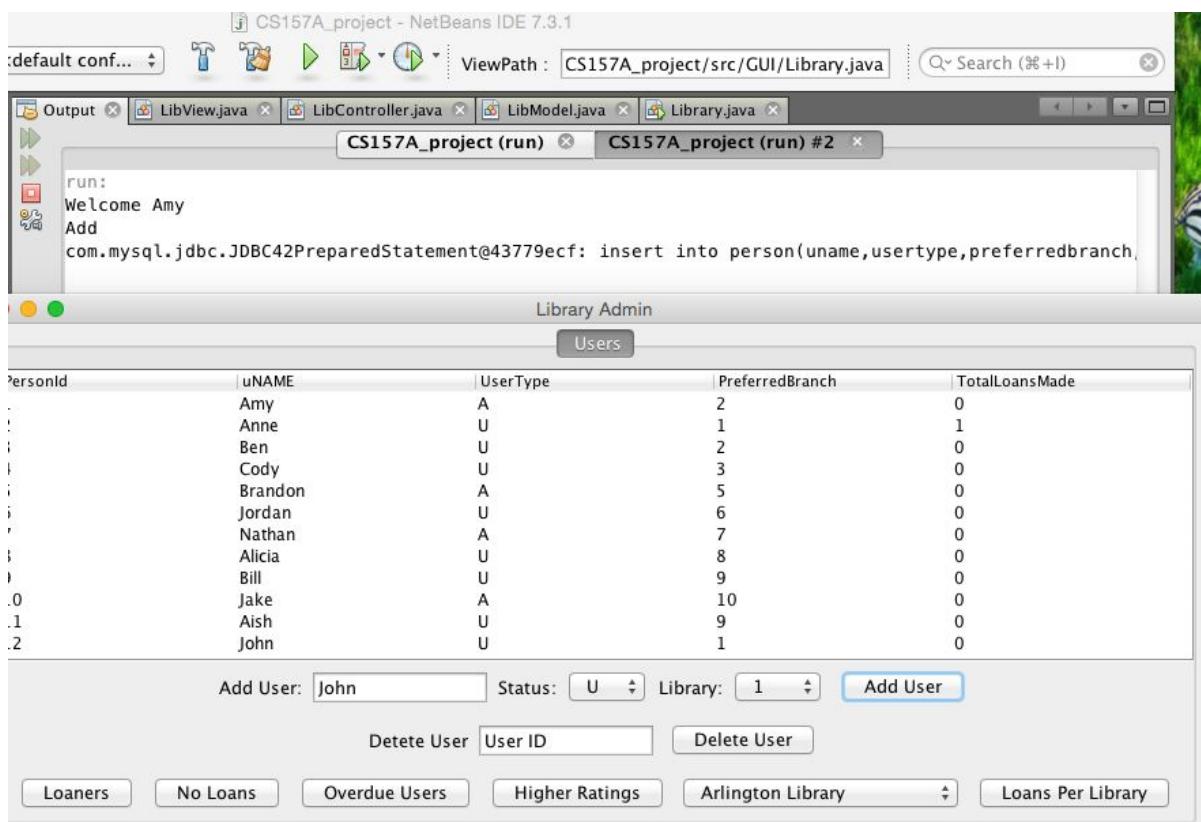
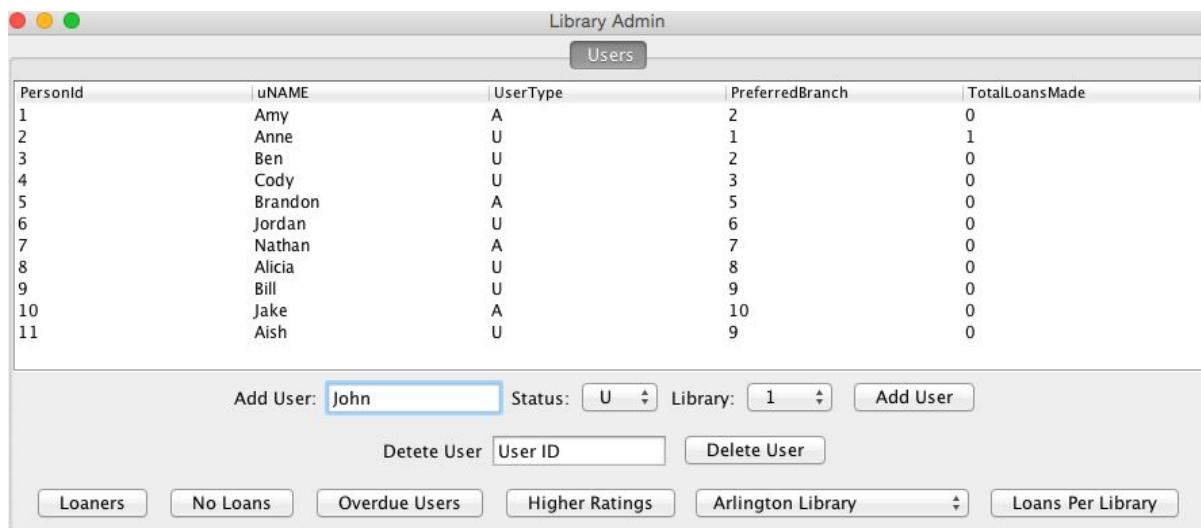
The screenshot shows a Mac OS X application window titled "Library User". The window has a standard title bar with red, yellow, and green close buttons. Below the title bar is a menu bar with four items: "Account", "Search", "Rating", and "Return".

The main content area contains several text fields and labels:

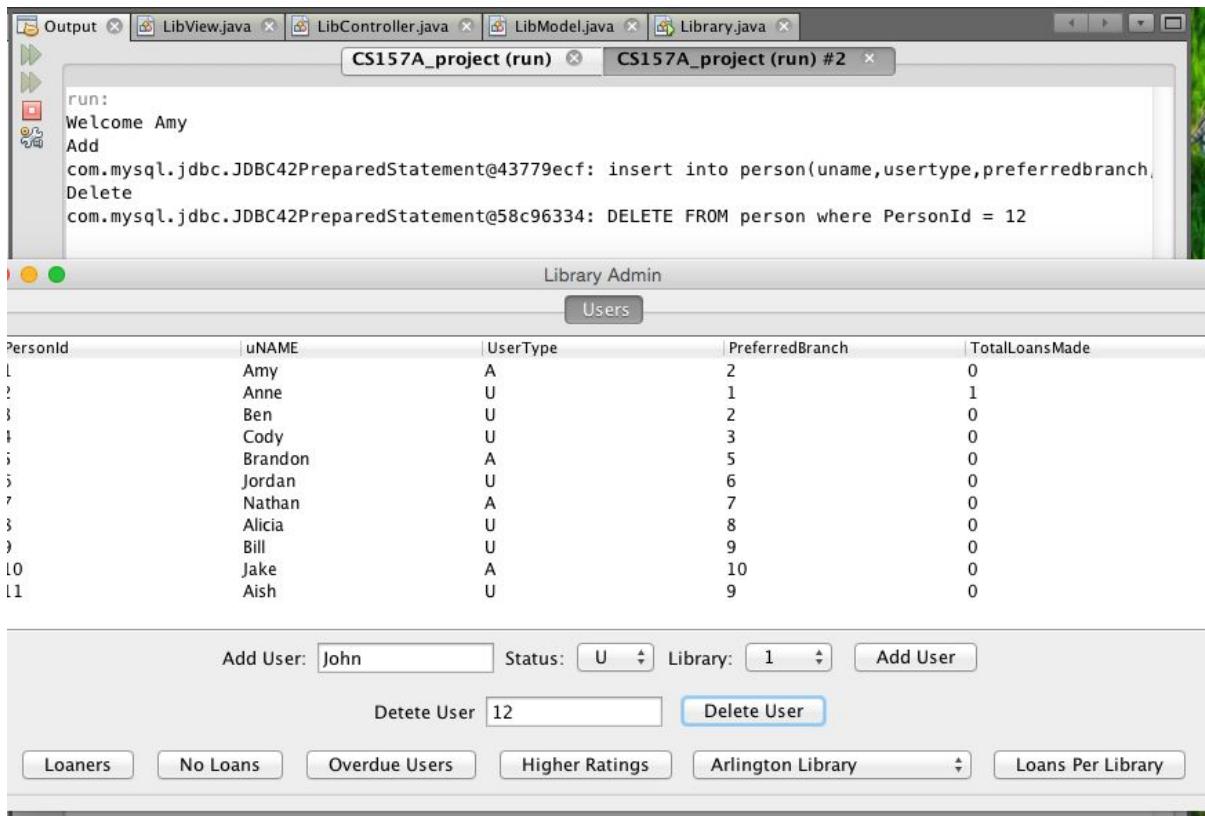
- A label "User ID:" followed by a text field containing the value "2".
- A label "Name:" followed by a text field containing the value "Anne".
- A label "Library" followed by a dropdown menu containing the value "Arlington Library".
- A large button at the bottom labeled "Update".

The window has scroll bars on the right side, indicating it is a scrollable view.

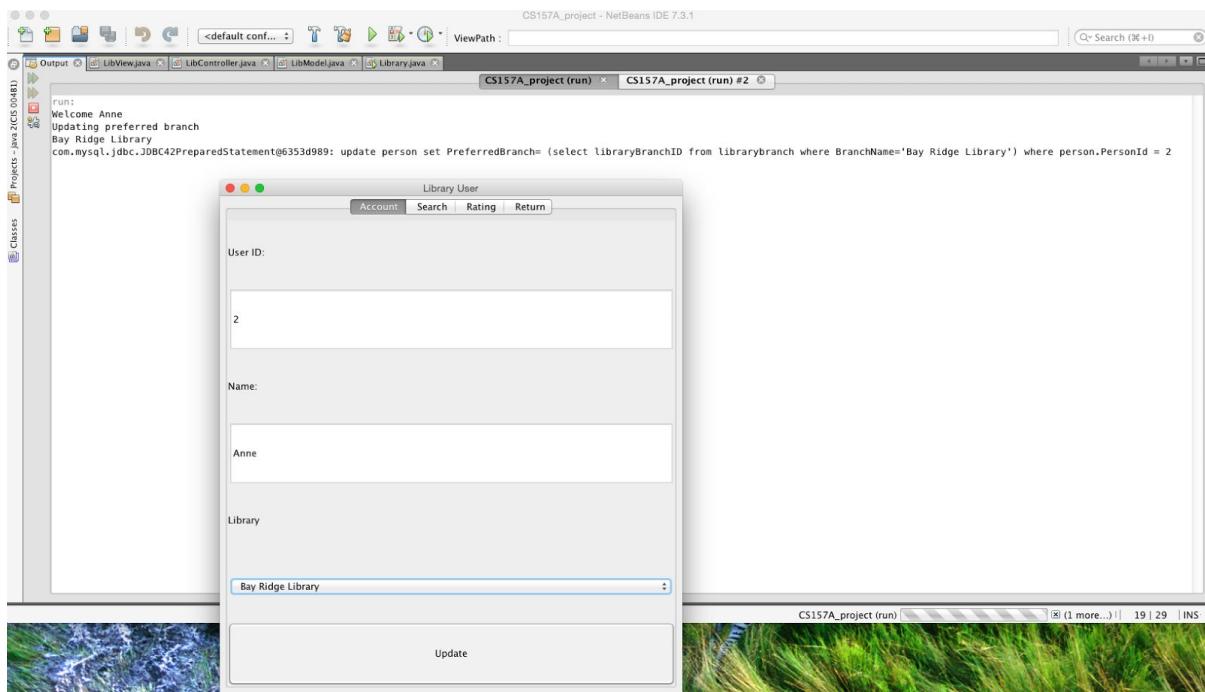
-- 2. Admin shall be able to add new user:



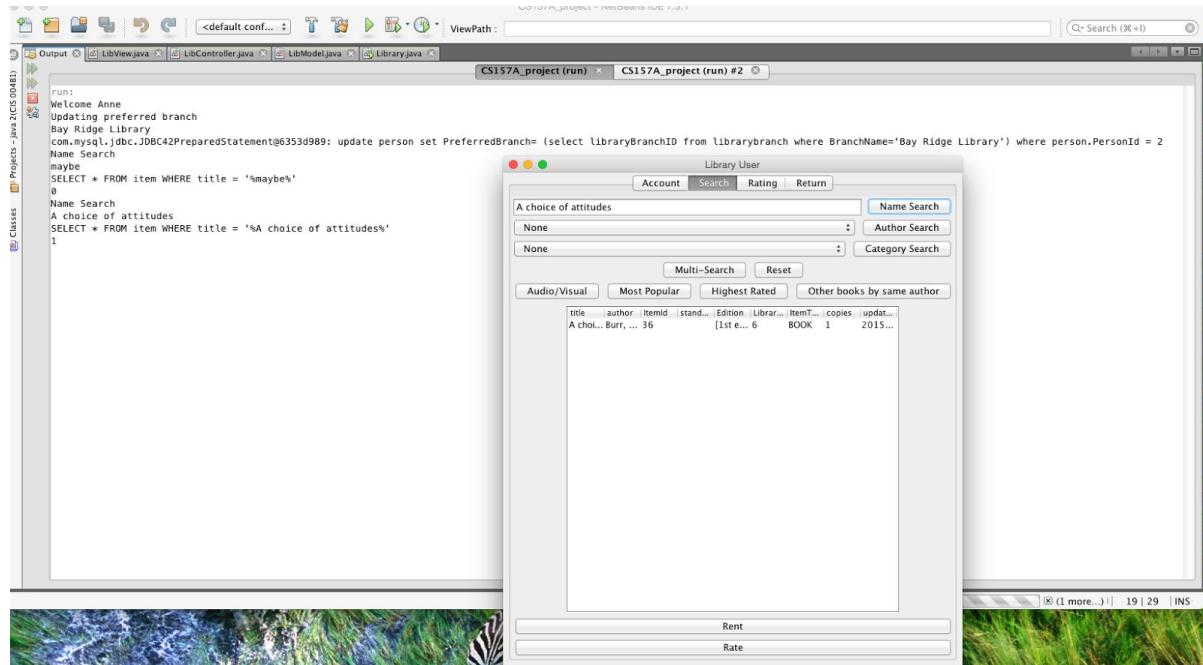
-- 3. Admin shall be able to delete a user:



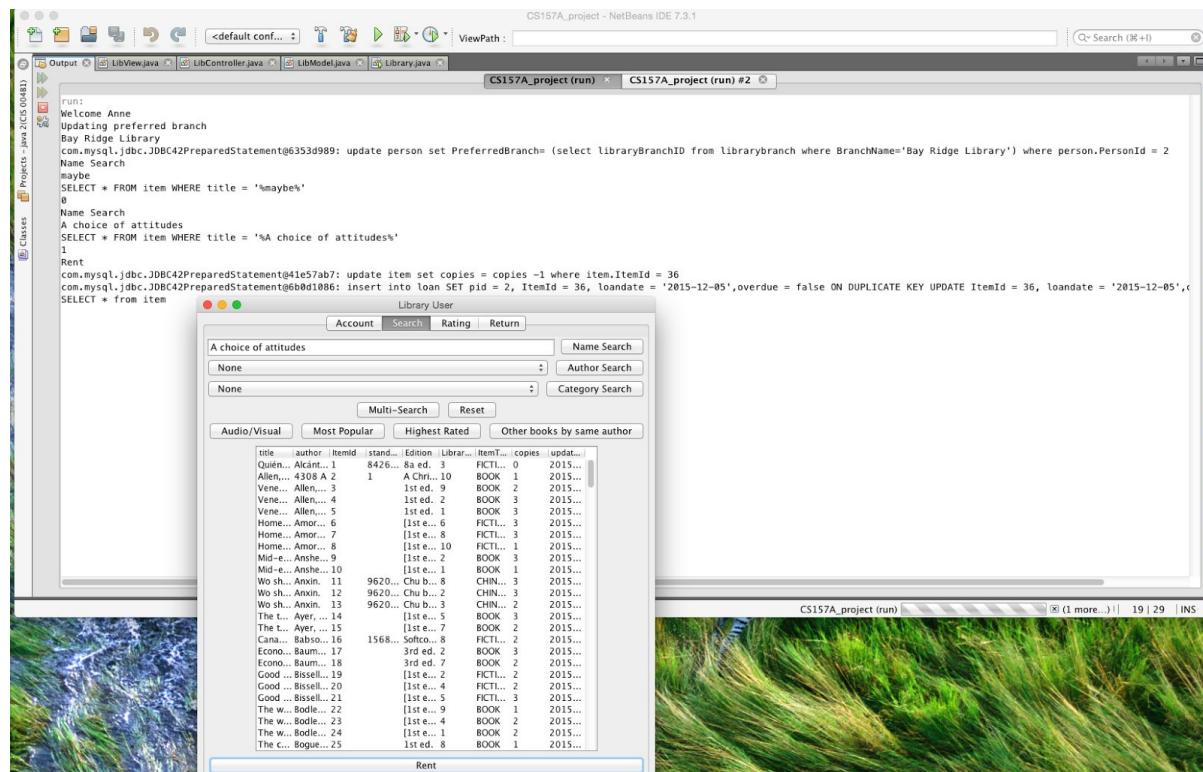
-- 4. Users shall be able to update information. [preferred branch]



-- 5. Users shall be able to search for a book using the title, author or category. Search can be done in any combination i.e. only one or two or three of these parameters can be given. Users should be able to search this in combination.



-- 6. User shall be able to rent a new book (Display books on for the library).

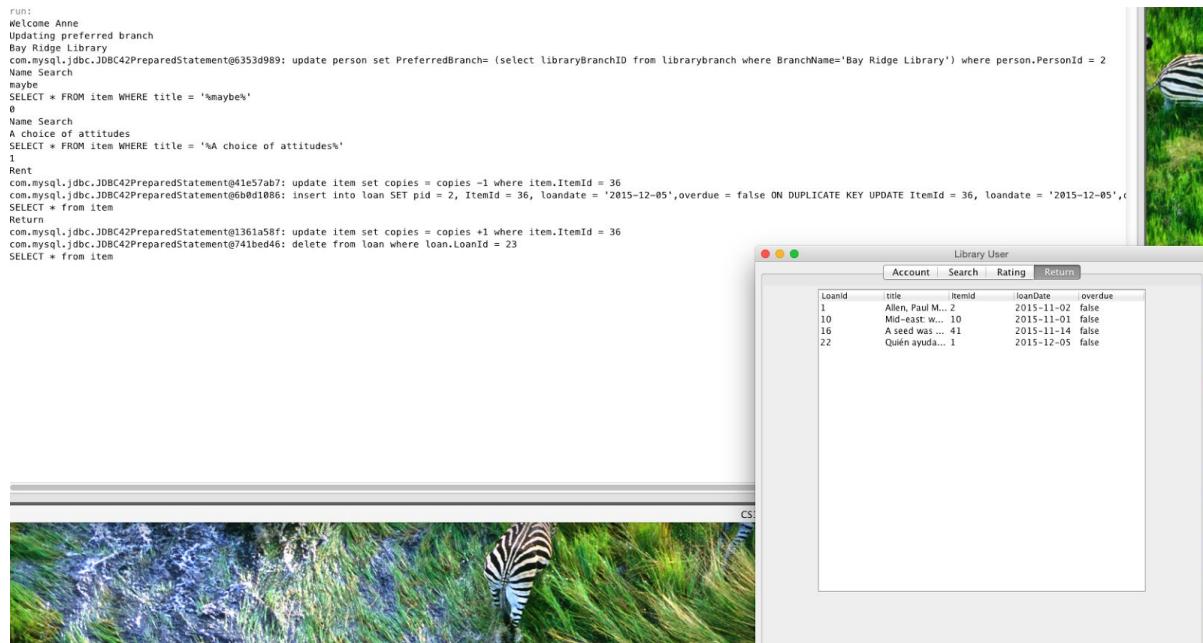


-- 7. Users shall be able to return a book.

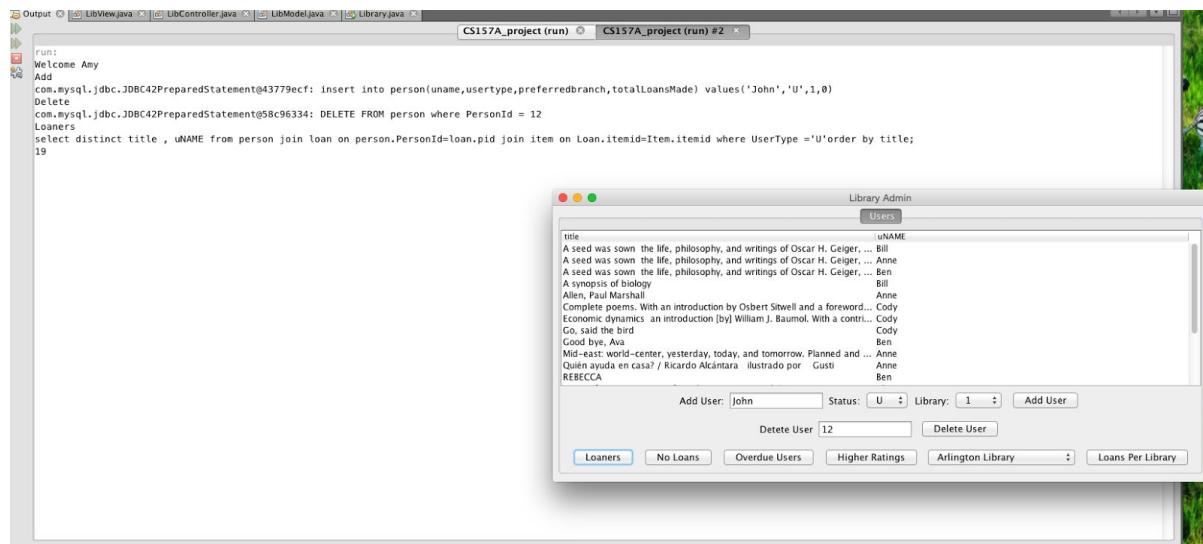
```

run:
Welcome Anne
Updating preferred branch
Bay Ridge Library
com.mysql.jdbc.JDBC42PreparedStatement@6353d989: update person set PreferredBranch= (select libraryBranchID from librarybranch where BranchName='Bay Ridge Library') where person.PersonId = 2
Name Search
maybe
SELECT * FROM item WHERE title = '%maybe%'
0
Name Search
A choice of attitudes
SELECT * FROM item WHERE title = '%A choice of attitudes%'
1
Rent
com.mysql.jdbc.JDBC42PreparedStatement@41e57ab7: update item set copies = copies -1 where item.ItemId = 36
com.mysql.jdbc.JDBC42PreparedStatement@6bd01086: insert into loan SET pid = 2, ItemId = 36, loanDate = '2015-12-05',overdue = false ON DUPLICATE KEY UPDATE ItemId = 36, loanDate = '2015-12-05',c
SELECT * FROM item
Return
com.mysql.jdbc.JDBC42PreparedStatement@1361a58f: update item set copies = copies +1 where item.ItemId = 36
com.mysql.jdbc.JDBC42PreparedStatement@741bed46: delete from loan where loan.LoanId = 23
SELECT * from item

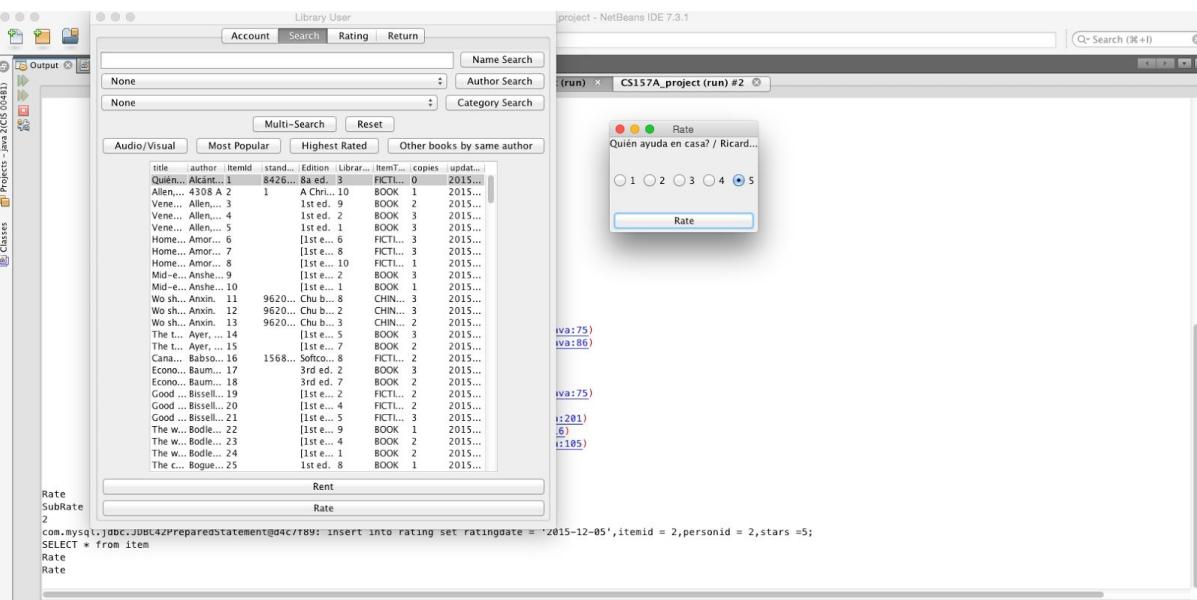
```



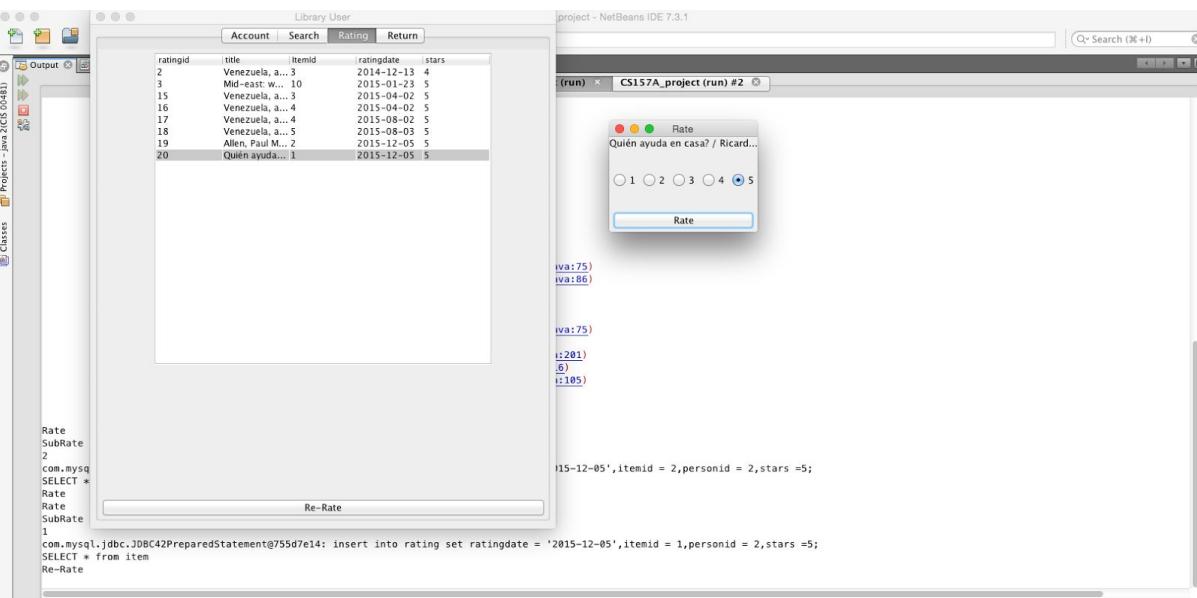
-- 8. Admins shall be able to see all distinct users who have rented a book.



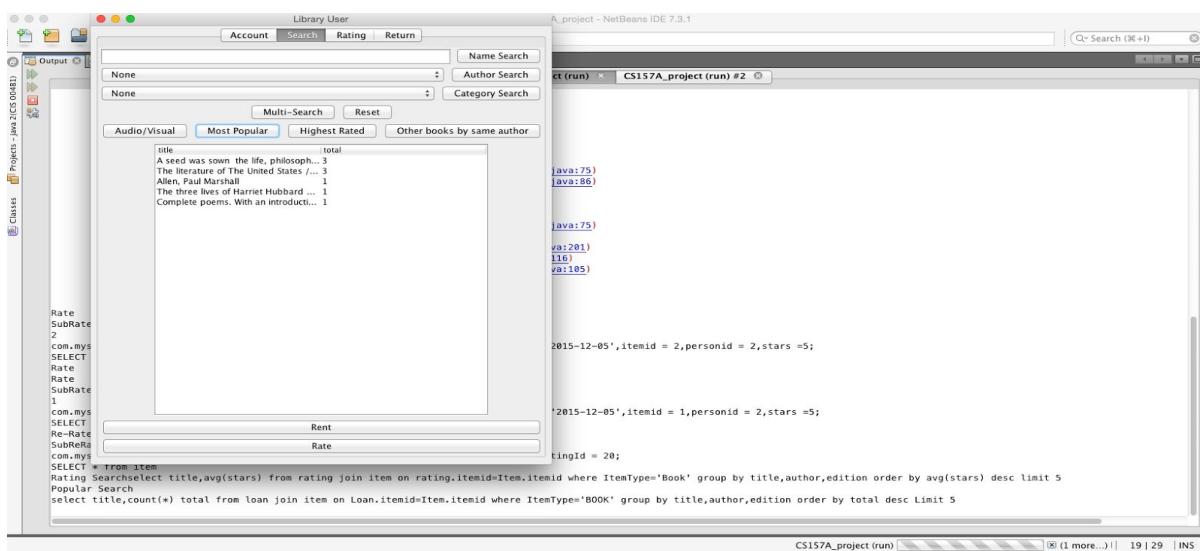
-- 9. Users can rate a book.



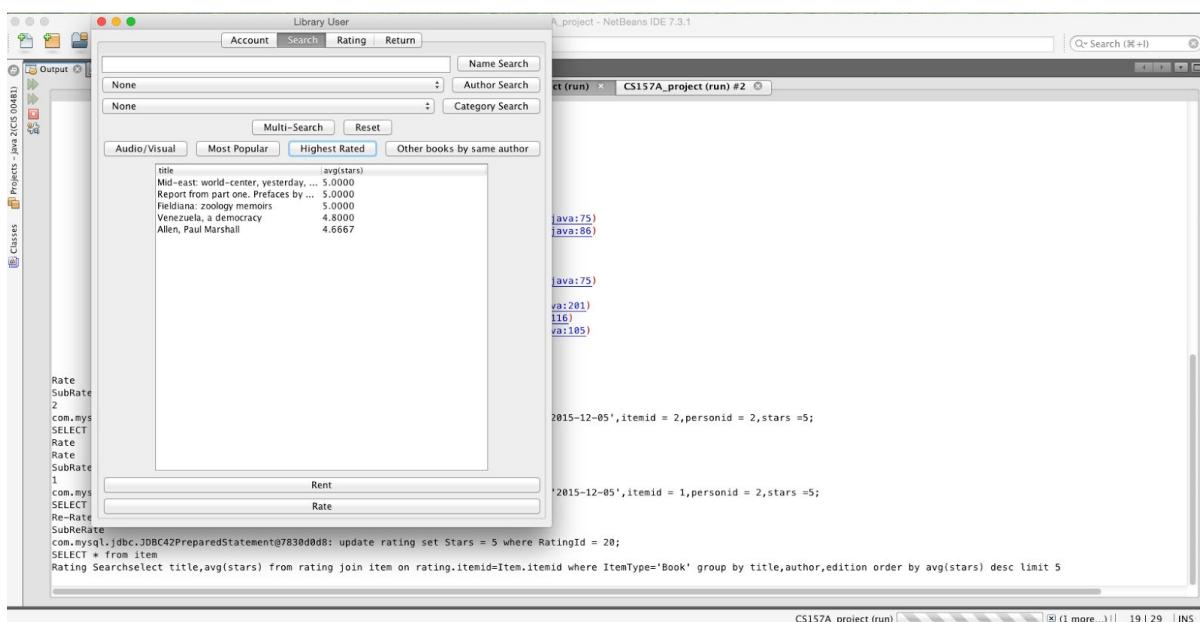
-- 10. Users can update the rating for a book later.



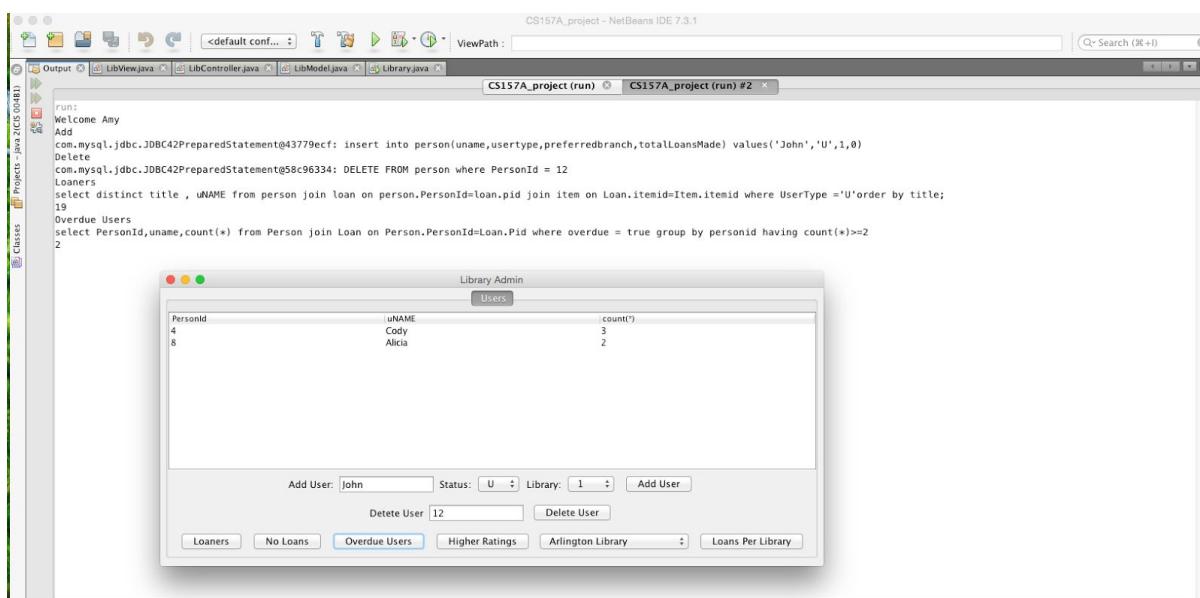
-- 11. Users shall be able to see the most popular books (top 5).



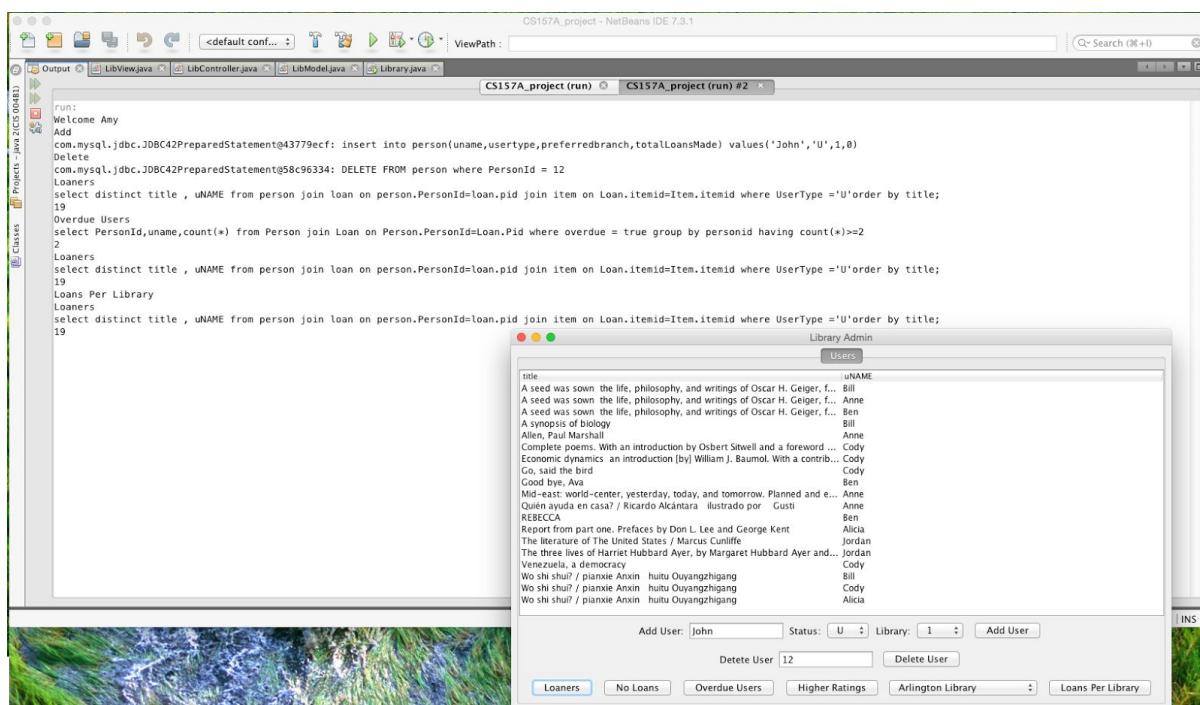
-- 12. Users shall be able to see the highest rated books (top 5).



-- 13. Admin shall be able to retrieve users who have several [two or more] overdue books.

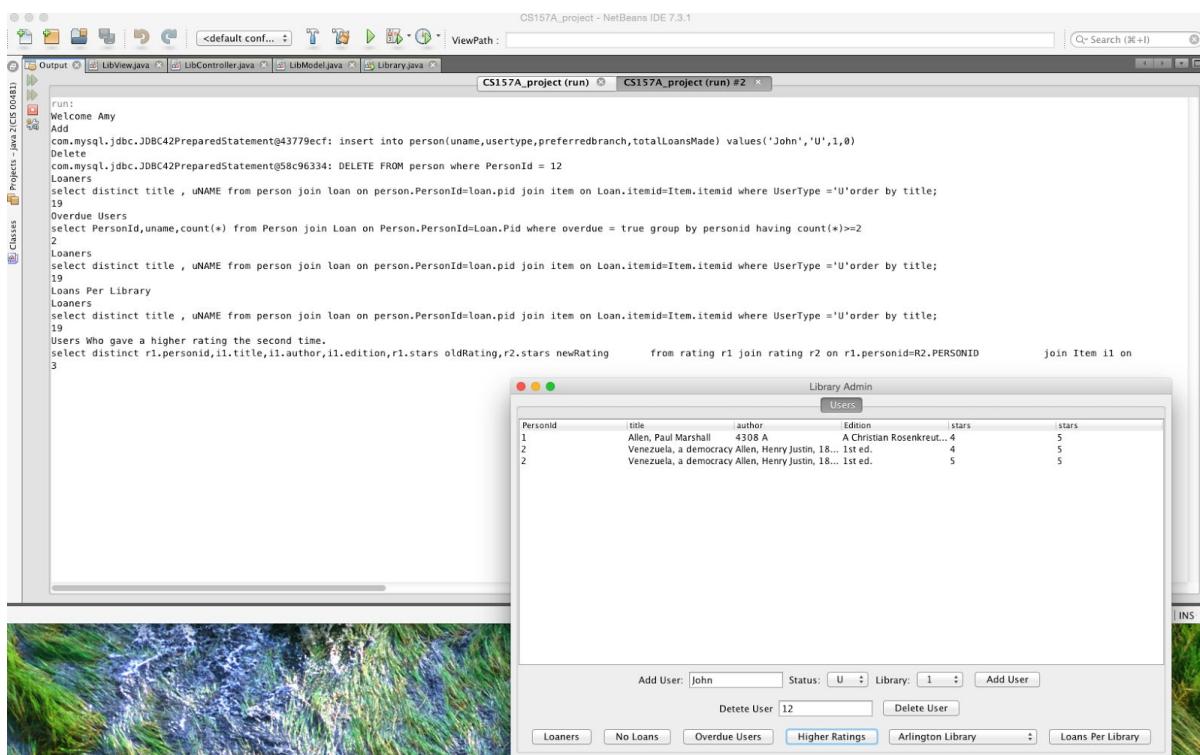


-- 14. Admins can find the total number of loans made at a library.



-- 15. Admin can find users who have given items a higher rating the second time he/she rated it.

CS 157A Final Report 24



-- 16. Users can find other books that have the same author as a given book.

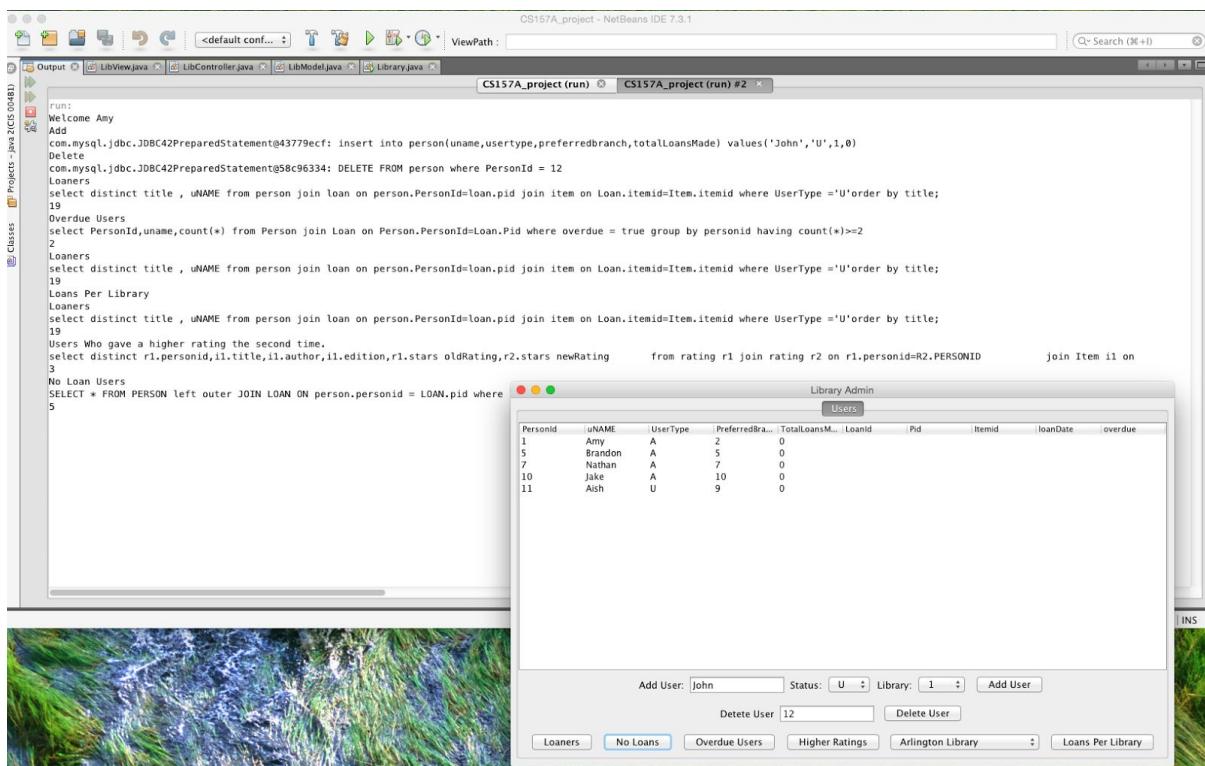
Tom Harris

title	author	itemid	stand...	Edition	Librar...	ItemT...	(copies)	update...
Tom ... Them...	Tom Harris	175	[1st A...	10	FICTI...	1	2015...	
Tom ... Them...	Tom Harris	176	[1st A...	9	FICTI...	3	2015...	

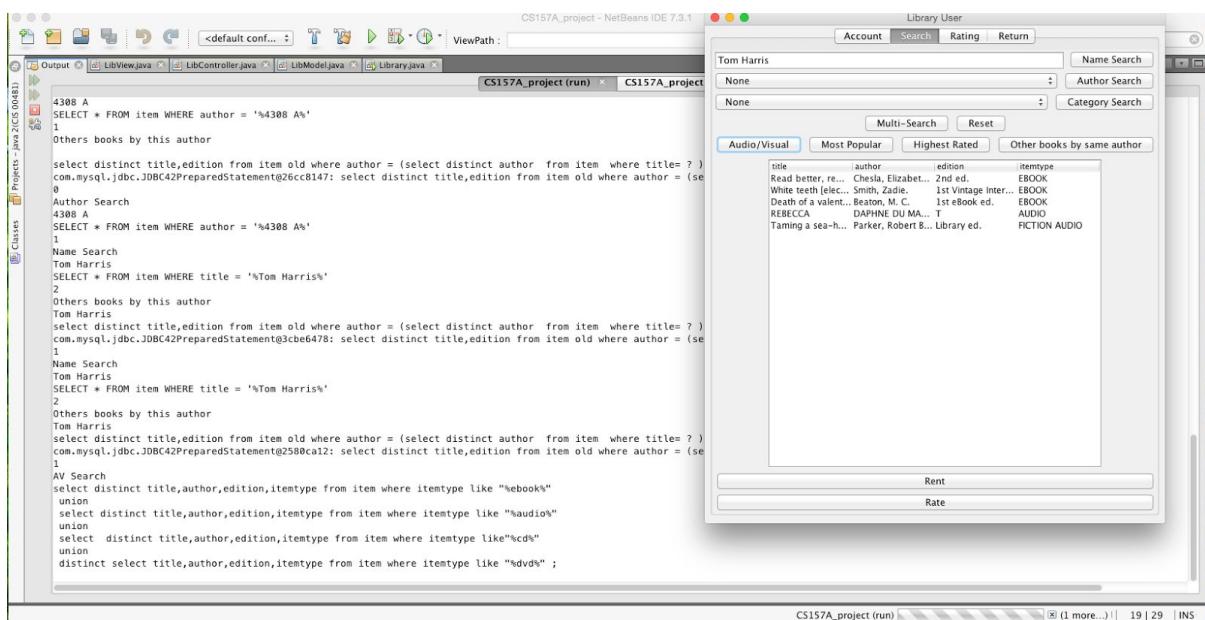
Tom Harris

title	Edition
Tom Harris	[1st American ed.]

-- 17. Admin can see users who have not loaned any books.

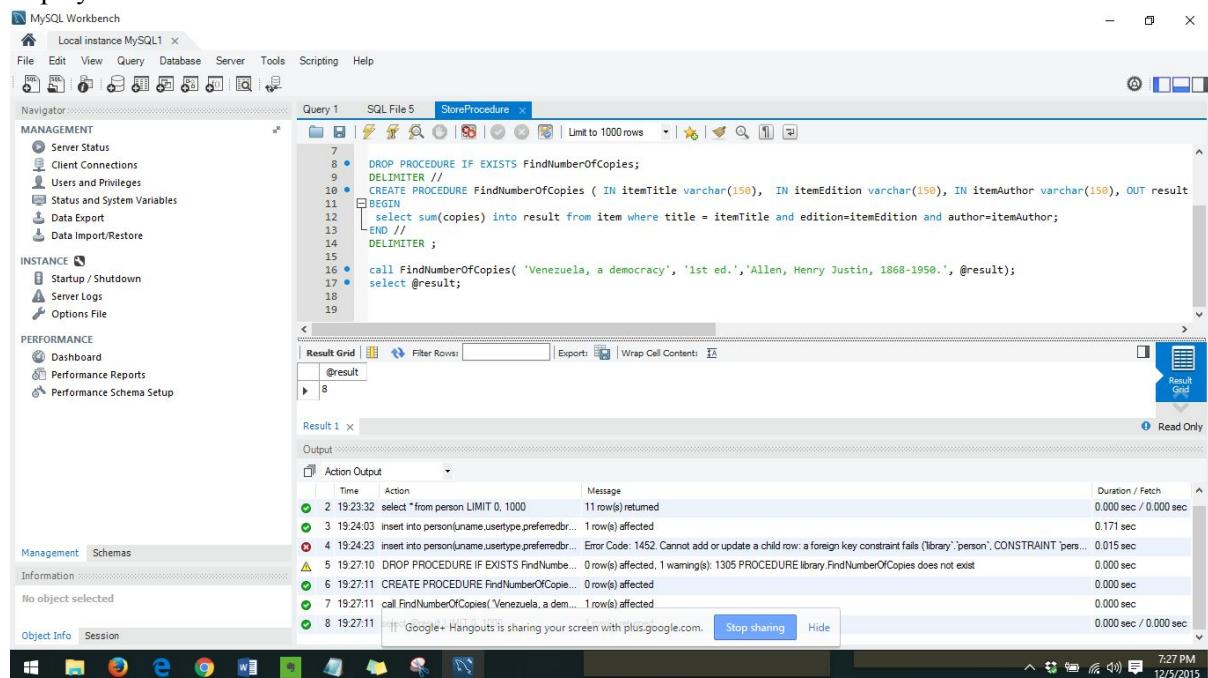


-- 18. Users can view all items that are of audio/video media [including e-books].



Stored Procedure Screenshots

After running FindNumberOfCopies procedure, the total number of copies of the given book is displayed in the result:



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL1, showing sections like MANAGEMENT, INSTANCE, PERFORMANCE, and Management.
- Query Editor:** Title: SQL File 5, Subtitle: StoredProcedure. Contains the following code:

```

7
8  DROP PROCEDURE IF EXISTS FindNumberOfCopies;
9
10 DELIMITER //
11 BEGIN
12   select sum(copies) into result from item where title = itemTitle and edition=itemEdition and author=itemAuthor;
13 END //
14 DELIMITER ;
15
16 call FindNumberOfCopies( 'Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result);
17 select @result;
18
19

```
- Result Grid:** Shows the output of the query: 8.
- Action Output:** Shows the history of actions taken during the session:

Time	Action	Message	Duration / Fetch
2 19:23:32	select *from person LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
3 19:24:03	insert into person(uname,userType,preferredbr...	1 row(s) affected	0.171 sec
4 19:24:23	insert into person(uname,userType,preferredbr... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (library`.`person`,CONSTRAINT `pers...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (library`.`person`,CONSTRAINT `pers...	0.015 sec
5 19:27:10	DROP PROCEDURE IF EXISTS FindNumber...	0 row(s) affected, 1 warning(s): 1305 PROCEDURE library.FindNumberOfCopies does not exist	0.000 sec
6 19:27:11	CREATE PROCEDURE FindNumberOfCopies(0 row(s) affected	0.000 sec
7 19:27:11	call FindNumberOfCopies('Venezuela, a dem...	1 row(s) affected	0.000 sec
8 19:27:11	Google+ Hangouts is sharing your screen with plus.google.com.		
- System Bar:** Shows various icons for system tasks like Task Manager, File Explorer, and Control Panel.
- Bottom Bar:** Shows the date and time: 7:27 PM, 12/5/2015.

Stored Procedure for Archiving Screenshot

After running the CopyToArchiveProcedure:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure.
- Code Editor:** Displays the stored procedure code:

```

CREATE PROCEDURE CopyToArchive ( IN cutoffDate timestamp )
BEGIN
    START TRANSACTION;
    insert into archiveitem(title,author,standardNumber,edition,libraryBranchID,itemtype,copies,updatedAt)
    select title,author,standardNumber,edition,libraryBranchID,itemtype,copies,updatedAt from item where date(updatedAt)<=date(cutoffdate);
    commit;
END //
DELIMITER ;
-- select * from archiveitem;
call copyToArchive(current_timestamp());
    
```
- Result Grid:** Shows the data copied from the 'item' table into the 'archiveitem' table. The data includes columns: title, author, ItemId, standardnumber, Edition, LibraryBranchID, ItemType. The ItemType column shows values like FICTION, SPANISH, BOOK, etc.
- Action Output:** Shows the execution log with the following entries:
 - Time: 7 19:27:11 Action: call findNumberOfCopies('Venezuela, a dem...') Message: 1 row(s) affected
 - Time: 8 19:27:11 Action: select @result LIMIT 0, 1000 Message: 1 row(s) returned
 - Time: 9 19:28:33 Action: use library Message: 0 row(s) affected
 - Time: 10 19:28:43 Action: DROP PROCEDURE IF EXISTS CopyToArc... Message: 0 row(s) affected, 1 warning(s): 1305 PROCEDURE library.CopyToArchive does not exist
 - Time: 11 19:28:43 Action: CREATE PROCEDURE CopyToArchive (IN ...) Message: 0 row(s) affected
 - Time: 12 19:28:54 Action: call copyToArchive(current_timestamp())+1) Message: 0 row(s) affected
 - Time: 13 19:28:55 Action: select * from archiveitem LIMIT 0, 1000 Message: 251 row(s) returned

This screenshot is nearly identical to the one above, showing the same interface and data. The key difference is in the Action Output section, where entry 10 shows a warning message:

```

Time: 10 19:28:43 Action: DROP PROCEDURE IF EXISTS CopyToArc... Message: 0 row(s) affected, 1 warning(s): 1305 PROCEDURE library.CopyToArchive does not exist

```

Key constraint and foreign key constraint violations Screenshots

Library Branch violations

-> Primary key violation [librarybranchid]

The screenshot shows the MySQL Workbench interface with multiple tabs open. The active tab, 'SQL File 13*', contains the following SQL query:

```
1 • INSERT INTO `librarybranch` VALUES (1,'Arlington Library','203 Arlington Ave. at Warwick St. Brooklyn, NY11207')
```

Below the query editor, the log window displays the following timeline of events:

Action	Time	Action	Response	Duration / Fetch Time
320	21:08:02	CREATE TRIGGER UpdateUserTotalLoans BEFORE INSERT ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoansMade =...	0 row(s) affected	0.023 sec
321	21:08:02	-- when a loan is closed, update the number of copies as well as the total loans made by user. DROP TRIGGER IF EXISTS UpdateUser...	0 row(s) affected, 1 warning(s): 1360 Trigger does no...	0.00014 sec
322	21:08:02	CREATE TRIGGER UpdateUserTotalLoansOnDelete BEFORE DELETE ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoa...	0 row(s) affected	0.022 sec
323	21:08:20	use library	0 row(s) affected	0.00017 sec
324	21:08:20	DROP PROCEDURE IF EXISTS FindNumberOfCopies	0 row(s) affected, 1 warning(s): 1305 PROCEDURE li...	0.00019 sec
325	21:08:20	CREATE PROCEDURE FindNumberOfCopies (IN itemTitle varchar(150), IN itemEdition varchar(150), IN itemAuthor varchar(150), OUT...	0 row(s) affected	0.00077 sec
326	21:08:20	call FindNumberOfCopies('Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result)	1 row(s) affected	0.00097 sec
327	21:08:20	select @result LIMIT 0, 1000	1 row(s) returned	0.00016 sec / 0.000...
328	21:47:58	INSERT INTO `librarybranch` VALUES (1,'Arlington Library','203 Arlington Ave. at Warwick St. Brooklyn, NY11207')	Error Code: 1062: Duplicate entry '1' for key 'PRIMARY'	0.00031 sec
329	21:49:33	insert into person(uname,userType,preferredBranch,totalLoansMade) values('Amy','A',2,0)	1 row(s) affected	0.00053 sec
330	21:52:52	insert into item(title,author,edition,libraryBranchID,itemType,standardNumber,copies) values('Quién ayuda en casa? / Ricardo Alcántar...',1)	1 row(s) affected	0.00055 sec
331	21:55:10	INSERT INTO `librarybranch` VALUES (1,'Arlington Library','203 Arlington Ave. at Warwick St. Brooklyn, NY11207')	Error Code: 1062: Duplicate entry '1' for key 'PRIMARY'	0.00039 sec

Query interrupted

-> No foreign keys associated with this table.

Person violations

-> Primary key violation [personid]

```
insert into person(uname,usertype,preferredbranch,totalLoansMade) values('Amy','A',2,0);
```

The screenshot shows the MySQL Workbench interface. In the main query editor window, there is a single line of SQL code: `insert into person(uname,usertype,preferredbranch,totalLoansMade) values('Amy','A',2,0);`. Below the editor is a history window titled "Action Output" which lists 28 rows of log entries. The log includes various database operations such as triggers, stored procedures, and insertions. The last entry in the log is a failed insert attempt:

Action	Time	Message	Duration / Fetch Time
insert into person(uname,usertype,preferredbranch,totalLoansMade) values('Amy','A',2,0);	14:36:52	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`person', CONSTRAINT `person_ibfk_1` FOREIGN KEY (PreferredBranch) REFERENCES `library` (BranchID))	0.000 sec

-> Foreign key violation [on personid]

Violate person LibraryBranchID foreign key:

```
insert into person(uname,usertype,preferredbranch,totalLoansMade)
values('Alice Liddel','U',11,0);
```

The screenshot shows the MySQL Workbench interface with the "Output" tab selected. It displays the same log from the history window, but the last entry is highlighted in red to indicate the error:

Action	Time	Message	Duration / Fetch Time
insert into person(uname,usertype,preferredbranch,totalLoansMade) values('Alice Liddel','U',11,0);	14:36:52	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`person', CONSTRAINT `person_ibfk_1` FOREIGN KEY (PreferredBranch) REFERENCES `library` (BranchID))	0.000 sec

Rating table violations

-> Primary key violation on ratingid:

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, the following SQL code is entered:

```

use library;
select * from rating;
insert into rating values("51",current_date(),200,1,1);

```

In the 'Output' pane, the results of the query are shown:

Action	Time	Message	Duration / Fetch
11 insert into loan values("51",15,"31",current_date(),"1")	10:34:06	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`loan', CONSTRAINT `loan_ibfk_1` FOREIGN KEY (`PersonId`) REFERENCES `person`(`PersonId`))	0.000 sec
12 insert into loan values("51",11,900,current_date(),"1")	10:35:50	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`loan', CONSTRAINT `loan_ibfk_1` FOREIGN KEY (`PersonId`) REFERENCES `person`(`PersonId`))	0.188 sec
13 select * from rating insert into loan values("51",11,900,cu...)	10:36:38	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'insert into loan values("51",11,900,cu...' at line 1	0.000 sec
14 select * from rating LIMIT 0, 1000	10:36:41	769 row(s) returned	0.000 sec / 0.000 sec
15 insert into rating values("51",current_date(),200,1,1)	10:37:14	Error Code: 1062. Duplicate entry '51' for key 'PRIMARY'	0.000 sec

-> Violate rating PersonId foreign key:

```

insert into rating(ratingdate,itemid,personid,stars)
values('2015-04-03',2,12,5);

```

The screenshot shows the MySQL Workbench interface. In the 'Output' pane, the results of the query are shown:

Action	Time	Message	Duration / Fetch
1 insert into rating(ratingdate,itemid,personid,stars) values(...)	14:43:35	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`rating', CONSTRAINT `rating_ibfk_1` FOREIGN KEY (`PersonId`) REFERENCES `person`(`PersonId`))	0.015 sec

-> Violate rating ItemId foreign key:

```

insert into rating(ratingdate,itemid,personid,stars)
values('2015-04-03',256,1,5);

```

The screenshot shows the MySQL Workbench interface. In the 'Output' pane, the results of the query are shown:

Action	Time	Message	Duration / Fetch
1 insert into rating(ratingdate,itemid,personid,stars) values(...)	14:44:53	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library`.`rating', CONSTRAINT `rating_ibfk_2` FOREIGN KEY (`ItemId`) REFERENCES `item`(`ItemId`))	0.000 sec

Loan table violations

-> Primary key violation [See error on query 26]

The screenshot shows the MySQL Workbench interface. In the top pane, there is a single query:

```
insert into loan(pid,itemid,loanDate,overdue) values(9,41,'2015-11-14',0);
```

In the bottom pane, the 'Action Output' tab displays the execution log for this query. The log shows various database operations and their results. The final entry for the insert query is highlighted in red, indicating an error:

Action	Time	Action	Response	Duration / Fetch Time
15	10:22:33	DROP TRIGGER IF EXISTS SuperReviewer	0 row(s) affected, 1 warning(s): 1360 Trigger does no...	0.00017 sec
16	10:22:33	CREATE TRIGGER SuperReviewer AFTER UPDATE ON person FOR EACH ROW BEGIN IF (new.uname='PaidGuy') then insert into...	0 row(s) affected	0.024 sec
17	10:22:33	-- select * from person join rating using(personid) where personid=8; -- update person set uname='PaidGuy' where personid=8; -- s...	0 row(s) affected, 1 warning(s): 1360 Trigger does no...	0.00022 sec
18	10:22:33	CREATE TRIGGER UpdateUserTotalLoans BEFORE INSERT ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoansMade =...	0 row(s) affected	0.022 sec
19	10:22:33	-- when a loan is closed, update the number of copies as well as the total loans made by user. DROP TRIGGER IF EXISTS UpdateUse...	0 row(s) affected, 1 warning(s): 1360 Trigger does no...	0.00012 sec
20	10:22:33	CREATE TRIGGER UpdateUserTotalLoansOnDelete BEFORE DELETE ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoansMade =...	0 row(s) affected	0.024 sec
21	10:22:39	use library	0 row(s) affected	0.00021 sec
22	10:22:39	DROP PROCEDURE IF EXISTS FindNumberOfCopies	0 row(s) affected	0.00018 sec
23	10:22:39	CREATE PROCEDURE FindNumberOfCopies (IN ItemTitle varchar(150), IN itemEdition varchar(150), IN itemAuthor varchar(150), OUT...	0 row(s) affected	0.00030 sec
24	10:22:39	call FindNumberOfCopies('Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result)	1 row(s) affected	0.00049 sec
25	10:22:39	select @result LIMIT 0, 1000	1 row(s) returned	0.00014 sec / 0.000...
26	10:22:44	insert into loan(pid,itemid,loanDate,overdue) values(9,41,'2015-11-14',0)	call FindNumberOfCopies('Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result)	update a child row:... 0.0010 sec

Query interrupted

Foreign key violation

-> trying to add a loan for a personid that does not exist.

Violate loan PersonId foreign key:

```
insert into loan(pid,itemid,loanDate,overdue) values
(12,2,'2015-11-02',false);
```

The screenshot shows the MySQL Workbench interface. The 'Output' tab displays the following error message:

```
1 14:39:06 insert into loan(pid,itemid,loanDate,overdue) values (12,2,... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.loan', CONSTRAINT loan_ibfk_1 FOREIGN KEY (Pid) REFERENCES 'person' (P... 0.000 sec
```

Violate loan ItemId foreign key:

```
insert into loan(pid,itemid,loanDate,overdue) values
(2,252,'2015-11-02',false);
```

The screenshot shows the MySQL Workbench interface. The 'Output' tab displays the following error message:

```
1 14:41:58 insert into loan(pid,itemid,loanDate,overdue) values (2,2,... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.loan', CONSTRAINT loan_ibfk_2 FOREIGN KEY (itemid) REFERENCES 'item' (I... 0.000 sec
```

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is run:

```
use library;
insert into loan values("51",15,"31",current_date(),"1");
```

In the Output window, the results of the execution are shown:

Action	Time	Message	Duration / Fetch
7 10:33:34 Insert into loan values("2","9","31",current_date(),"1")		Error Code: 1062. Duplicate entry '2' for key 'PRIMARY'	0.094 sec
8 10:33:46 Insert into loan values("2",11,"31",current_date(),"1")		Error Code: 1062. Duplicate entry '2' for key 'PRIMARY'	0.000 sec
9 10:33:52 Insert into loan values("50",11,"31",current_date(),"1")		1 row(s) affected	0.094 sec
10 10:34:01 Insert into loan values("50",15,"31",current_date(),"1")		Error Code: 1062. Duplicate entry '50' for key 'PRIMARY'	0.000 sec
11 10:34:06 Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.'loan', CONSTRAINT 'loan_ibfk_1' FOREIGN KEY ('Pid') REFERENCES 'person' ('PersonId'))			

The status bar at the bottom right indicates the time as 10:34 AM and the date as 12/6/2015.

-> Trying to add a loan on a itemid that does not exist:

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is run:

```
use library;
insert into loan values("51",11,900,current_date(),"1");
```

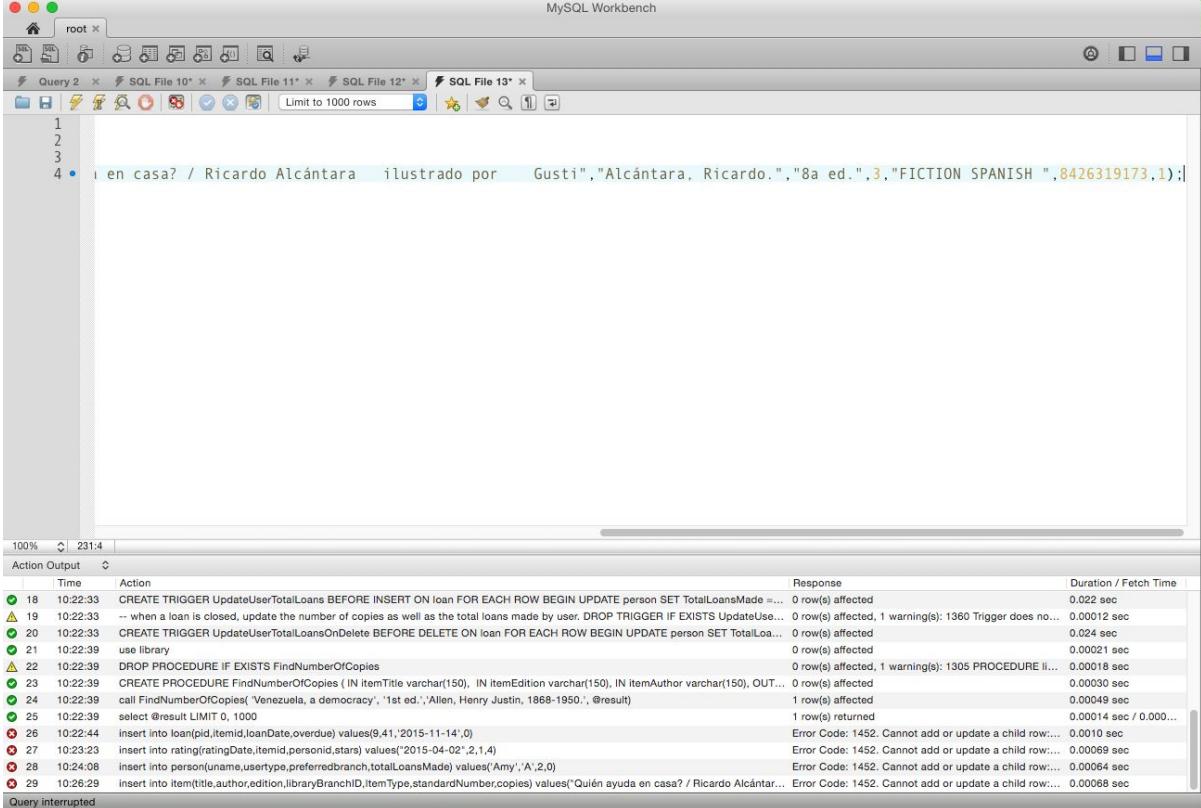
In the Output window, the results of the execution are shown:

Action	Time	Message	Duration / Fetch
8 10:33:46 Insert into loan values("2",11,"31",current_date(),"1")		Error Code: 1062. Duplicate entry '2' for key 'PRIMARY'	0.000 sec
9 10:33:52 Insert into loan values("50",11,"31",current_date(),"1")		1 row(s) affected	0.094 sec
10 10:34:01 Insert into loan values("50",15,"31",current_date(),"1")		Error Code: 1062. Duplicate entry '50' for key 'PRIMARY'	0.000 sec
11 10:34:06 Insert into loan values("51",15,"31",current_date(),"1")		Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.'loan', CONSTRAINT 'loan_ibfk_2' FOREIGN KEY ('Itemid') REFERENCES 'item' ('Itemid'))	0.000 sec
12 10:35:50 Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.'loan', CONSTRAINT 'loan_ibfk_2' FOREIGN KEY ('Itemid') REFERENCES 'item' ('Itemid'))			

The status bar at the bottom right indicates the time as 10:35 AM and the date as 12/6/2015.

Item table violations

-> Primary key violation [on itemid]



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

1
2
3
4 • "Quién ayuda en casa? / Ricardo Alcántara ilustrado por Gusti","Alcántara, Ricardo.", "8a ed.", 3, "FICTION SPANISH ", 8426319173, 1);

```

The status bar at the bottom indicates "Query 2" and "root". Below the editor is a log window titled "Action Output" showing the execution of various database statements, including triggers, procedures, and insertions. The log ends with a message: "Query interrupted".

Action	Time	Response	Duration / Fetch Time
CREATE TRIGGER UpdateUserTotalLoans BEFORE INSERT ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoansMade =...	10:22:33	0 row(s) affected	0.022 sec
-- when a loan is closed, update the number of copies as well as the total loans made by user. DROP TRIGGER IF EXISTS UpdateUser...	10:22:33	0 row(s) affected, 1 warning(s): 1360 Trigger does no...	0.00012 sec
CREATE TRIGGER UpdateUserTotalLoansOnDelete BEFORE DELETE ON loan FOR EACH ROW BEGIN UPDATE person SET TotalLoa...	10:22:33	0 row(s) affected	0.024 sec
use library	10:22:39	0 row(s) affected	0.00021 sec
DROP PROCEDURE IF EXISTS FindNumberOfCopies	10:22:39	0 row(s) affected, 1 warning(s): 1305 PROCEDURE li...	0.00018 sec
CREATE PROCEDURE FindNumberOfCopies (IN ItemTitle varchar(150), IN ItemEdition varchar(150), IN ItemAuthor varchar(150), OUT...	10:22:39	0 row(s) affected	0.00030 sec
call FindNumberOfCopies('Venezuela, a democracy', '1st ed.', 'Allen, Henry Justin, 1868-1950.', @result)	10:22:39	1 row(s) affected	0.00049 sec
select @result LIMIT 0, 1000	10:22:39	1 row(s) returned	0.00014 sec / 0.000...
insert into loan(pid, itemid, loanDate, overdue) values(9,41, '2015-11-14', 0)	10:22:44	Error Code: 1452. Cannot add or update a child row...	0.00010 sec
insert into rating(ratingDate, itemid, personid, stars) values('2015-04-02', 2, 1, 4)	10:23:23	Error Code: 1452. Cannot add or update a child row...	0.00069 sec
insert into person(uname, userType, preferredBranch, totalLoansMade) values('Amy', 'A', 2, 0)	10:24:08	Error Code: 1452. Cannot add or update a child row...	0.00064 sec
insert into item(title, author, edition, libraryBranchID, itemType, standardNumber, copies) values("Quién ayuda en casa? / Ricardo Alcántar...	10:26:29	Error Code: 1452. Cannot add or update a child row...	0.00068 sec

-> Foreign key violation [LibraryBranchID constraint]

insert into item (title,author,edition,libraryBranchID,ItemType, standardNumber, copies) values("Alice's Adventures in Wonderland," Lewis Carroll", "1a", 11, "FICTION ", 1, 1);



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

insert into item (title,author,edition,libraryBranchID,ItemType, standardNumber, copies) values("Alice's Adventures in Wonderland," Lewis Carroll", "1a", 11, "FICTION ", 1, 1);

```

The status bar at the bottom indicates "Query 2" and "root". Below the editor is a log window titled "Output" showing the execution of the insertion statement. The log ends with a message: "Query interrupted".

Action	Time	Message	Duration / Fetch
insert into item(title,author,edition,libraryBranchID,ItemType,s...	14:34:05	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('library'.'item', CONSTRAINT 'item_ibfk_1' FOREIGN KEY ('LibraryBranchID') REFERENCE...	0.000 sec

Trigger Screenshots

If a user's name has been changed to paid guy, add "5" star reviews from that person for all items. The logic is that if this person has been paid to write good reviews, we just add his/her "5 star" reviews to all books.

Before:

	uName	ratingid	ratingdate	ItemId	PersonId	stars
▶	Amy	1	2015-04-03	2	1	5
	Amy	14	2015-04-02	2	1	4
	Anne	2	2014-12-13	3	2	4
	Anne	3	2015-01-23	10	2	5
	Anne	15	2015-04-02	3	2	5
	Anne	16	2015-04-02	4	2	5
	Anne	17	2015-08-02	4	2	5
	Anne	18	2015-08-03	5	2	5
	Cody	4	2013-04-05	12	4	2
	Cody	5	2015-02-13	40	4	5
	Jordan	7	2015-05-03	71	6	3
	Jordan	8	2014-01-10	81	6	1
	Alicia	9	2011-03-09	41	8	2
	Alicia	11	2015-08-27	51	8	4
	Alicia	13	2012-06-09	50	8	3
	Bill	6	2012-08-23	50	9	4
	Bill	10	2010-04-07	31	9	5
	Bill	12	2013-07-13	61	9	4

After:

```
update person set uname="PaidGuy" where personid=9;
```

	uName	ratingid	ratingdate	ItemId	PersonId	stars
▶	Amy	1	2015-04-03	2	1	5
	Amy	14	2015-04-02	2	1	4
	Anne	2	2014-12-13	3	2	4
	Anne	3	2015-01-23	10	2	5
	Anne	15	2015-04-02	3	2	5
	Anne	16	2015-04-02	4	2	5
	Anne	17	2015-08-02	4	2	5
	Anne	18	2015-08-03	5	2	5
	Cody	4	2013-04-05	12	4	2
	Cody	5	2015-02-13	40	4	5
	Jordan	7	2015-05-03	71	6	3
	Jordan	8	2014-01-10	81	6	1
	Alicia	9	2011-03-09	41	8	2
	Alicia	11	2015-08-27	51	8	4
	Alicia	13	2012-06-09	50	8	3
	PaidGuy	6	2012-08-23	50	9	4
	PaidGuy	10	2010-04-07	31	9	5
	PaidGuy	12	2013-07-13	61	9	4
	PaidGuy	19	2015-12-06	5	9	5
	PaidGuy	20	2015-12-06	10	9	5
	PaidGuy	21	2015-12-06	24	9	5
	PaidGuy	22	2015-12-06	28	9	5
	PaidGuy	23	2015-12-06	38	9	5
	PaidGuy	24	2015-12-06	48	9	5
	PaidGuy	25	2015-12-06	51	9	5
	PaidGuy	26	2015-12-06	67	9	5
	PaidGuy	27	2015-12-06	69	9	5

	uName	ratingid	ratingdate	ItemId	PersonId	stars
	PaidGuy	27	2015-12-06	69	9	5
	PaidGuy	28	2015-12-06	74	9	5
	PaidGuy	29	2015-12-06	78	9	5
	PaidGuy	30	2015-12-06	82	9	5
	PaidGuy	31	2015-12-06	88	9	5
	PaidGuy	32	2015-12-06	100	9	5
	PaidGuy	33	2015-12-06	114	9	5
	PaidGuy	34	2015-12-06	141	9	5
	PaidGuy	35	2015-12-06	143	9	5
	PaidGuy	36	2015-12-06	147	9	5
	PaidGuy	37	2015-12-06	150	9	5
	PaidGuy	38	2015-12-06	153	9	5
	PaidGuy	39	2015-12-06	155	9	5
	PaidGuy	40	2015-12-06	168	9	5
	PaidGuy	41	2015-12-06	180	9	5
	PaidGuy	42	2015-12-06	196	9	5
	PaidGuy	43	2015-12-06	205	9	5
	PaidGuy	44	2015-12-06	208	9	5
	PaidGuy	45	2015-12-06	210	9	5
	PaidGuy	46	2015-12-06	214	9	5
	PaidGuy	47	2015-12-06	218	9	5
	PaidGuy	48	2015-12-06	222	9	5
	PaidGuy	49	2015-12-06	228	9	5
	PaidGuy	50	2015-12-06	241	9	5
	PaidGuy	51	2015-12-06	246	9	5
	PaidGuy	52	2015-12-06	4	9	5
	PaidGuy	53	2015-12-06	9	9	5
	PaidGuy	54	2015-12-06	12	9	5

When a loan is made, update the number of copies as well as the total loans made by user.

Before:

	uName	TotalLoansMade	LoanId	Pid	Itemid	loanDate	overdue
▶	Anne	3	1	2	2	2015-11-02	0
	Anne	3	10	2	10	2015-11-01	0
	Anne	3	16	2	41	2015-11-14	0
	Ben	4	3	3	20	2015-09-03	1
	Ben	4	4	3	19	2015-11-27	0
	Ben	4	15	3	41	2015-11-14	0
	Ben	4	21	3	61	2015-11-14	0
	Cody	5	5	4	17	2015-10-19	1
	Cody	5	9	4	12	2015-09-12	1
	Cody	5	11	4	3	2015-10-10	1
	Cody	5	13	4	44	2015-11-14	0
	Cody	5	20	4	50	2015-11-14	0
	Jordan	4	6	6	14	2015-09-18	0
	Jordan	4	17	6	47	2015-11-14	0
	Jordan	4	18	6	48	2015-11-14	0
	Jordan	4	19	6	49	2015-11-14	0
	Alicia	2	2	8	31	2015-10-05	1
	Alicia	2	7	8	13	2015-11-17	1
	PaidGuy	3	8	9	12	2015-10-05	0
	PaidGuy	3	12	9	41	2015-11-14	0
	PaidGuy	3	14	9	45	2015-11-14	0

After:

```

Alice (Pid = 10) Loaned Book num 10
insert into loan(pid,itemid,loanDate,overdue) values
(8,10,'2015-11-02',false);

```

	uName	TotalLoansMade	LoanId	Pid	Itemid	loanDate	overdue
▶	Anne	3	1	2	2	2015-11-02	0
	Anne	3	10	2	10	2015-11-01	0
	Anne	3	16	2	41	2015-11-14	0
	Ben	4	3	3	20	2015-09-03	1
	Ben	4	4	3	19	2015-11-27	0
	Ben	4	15	3	41	2015-11-14	0
	Ben	4	21	3	61	2015-11-14	0
	Cody	5	5	4	17	2015-10-19	1
	Cody	5	9	4	12	2015-09-12	1
	Cody	5	11	4	3	2015-10-10	1
	Cody	5	13	4	44	2015-11-14	0
	Cody	5	20	4	50	2015-11-14	0
	Jordan	4	6	6	14	2015-09-18	0
	Jordan	4	17	6	47	2015-11-14	0
	Jordan	4	18	6	48	2015-11-14	0
	Jordan	4	19	6	49	2015-11-14	0
	Alicia	3	2	8	31	2015-10-05	1
	Alicia	3	7	8	13	2015-11-17	1
	Alicia	3	22	8	10	2015-11-02	0
	PaidGuy	3	8	9	12	2015-10-05	0
	PaidGuy	3	12	9	41	2015-11-14	0
	PaidGuy	3	14	9	45	2015-11-14	0

When a loan is closed, update the number of copies as well as the total loans made by user.

Before:

	uName	TotalLoansMade	LoanId	Pid	Itemid	loanDate	overdue
▶	Anne	3	1	2	2	2015-11-02	0
	Anne	3	10	2	10	2015-11-01	0
	Anne	3	16	2	41	2015-11-14	0
	Ben	4	3	3	20	2015-09-03	1
	Ben	4	4	3	19	2015-11-27	0
	Ben	4	15	3	41	2015-11-14	0
	Ben	4	21	3	61	2015-11-14	0
	Cody	5	5	4	17	2015-10-19	1
	Cody	5	9	4	12	2015-09-12	1
	Cody	5	11	4	3	2015-10-10	1
	Cody	5	13	4	44	2015-11-14	0
	Cody	5	20	4	50	2015-11-14	0
	Jordan	4	6	6	14	2015-09-18	0
	Jordan	4	17	6	47	2015-11-14	0
	Jordan	4	18	6	48	2015-11-14	0
	Jordan	4	19	6	49	2015-11-14	0
	Alicia	3	2	8	31	2015-10-05	1
	Alicia	3	7	8	13	2015-11-17	1
	Alicia	3	22	8	10	2015-11-02	0
	PaidGuy	3	8	9	12	2015-10-05	0
	PaidGuy	3	12	9	41	2015-11-14	0
	PaidGuy	3	14	9	45	2015-11-14	0

After:

```
Alice (Pid = 10) Returned Book num 10 (LoanID = 22)
delete from loan where LoanId = 22;
```

	uName	TotalLoansMade	LoanId	Pid	Itemid	loanDate	overdue
▶	Anne	3	1	2	2	2015-11-02	0
	Anne	3	10	2	10	2015-11-01	0
	Anne	3	16	2	41	2015-11-14	0
	Ben	4	3	3	20	2015-09-03	1
	Ben	4	4	3	19	2015-11-27	0
	Ben	4	15	3	41	2015-11-14	0
	Ben	4	21	3	61	2015-11-14	0
	Cody	5	5	4	17	2015-10-19	1
	Cody	5	9	4	12	2015-09-12	1
	Cody	5	11	4	3	2015-10-10	1
	Cody	5	13	4	44	2015-11-14	0
	Cody	5	20	4	50	2015-11-14	0
	Jordan	4	6	6	14	2015-09-18	0
	Jordan	4	17	6	47	2015-11-14	0
	Jordan	4	18	6	48	2015-11-14	0
	Jordan	4	19	6	49	2015-11-14	0
	Alicia	2	2	8	31	2015-10-05	1
	Alicia	2	7	8	13	2015-11-17	1
	PaidGuy	3	8	9	12	2015-10-05	0
	PaidGuy	3	12	9	41	2015-11-14	0
	PaidGuy	3	14	9	45	2015-11-14	0