Honor Code

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Task #1

For this task we decided to use mainly VARCHARS for names of publishers and addresses to give ample space for any address or name to be entered. Many inputs such as phone numbers that had a set number of characters we decided to limit the length of input and specify how many characters they are allowed. Numbers that are needed to identify someone and assigned by the database though are integers and have an auto increment constraint on them in order to automatically assign a value to that data entry. The tables that reference those numbers just have these identifying values as a foreign key.

```
CREATE TABLE PUBLISHER
(
   Publisher_Name    VARCHAR(30)        NOT NULL,
   Phone             CHAR(12),
   Address           VARCHAR(60),

   PRIMARY KEY (Publisher_Name)
);

CREATE TABLE LIBRARY_BRANCH
(
   Branch_Id         INT                AUTO_INCREMENT,
   Branch_Name       VARCHAR(30)        NOT NULL,
   Branch_Address    VARCHAR(60)        NOT NULL,

   PRIMARY KEY (Branch_Id)
);

CREATE TABLE BORROWER
(
   Card_No           INT                AUTO_INCREMENT,
   Name              VARCHAR(30)        NOT NULL,
   Address           VARCHAR(60)        NOT NULL,
   Phone             CHAR(12)           NOT NULL,

   PRIMARY KEY (Card_No)
```

```
);

CREATE TABLE BOOK
(
   Book_Id          INT             AUTO_INCREMENT,
   Title            VARCHAR(50)     NOT NULL,
   Publisher_Name   VARCHAR(30)     NOT NULL,

   PRIMARY KEY (Book_Id),
   FOREIGN KEY (Publisher_Name) REFERENCES PUBLISHER (Publisher_Name)
   ON UPDATE CASCADE
);

CREATE TABLE BOOK_LOANS
(
   Book_Id          INT             NOT NULL,
   Branch_Id        INT             NOT NULL,
   Card_No          INT             NOT NULL,
   Date_Out         DATE            NOT NULL,
   Due_Date         DATE            NOT NULL,
   Returned_date    DATE,

   PRIMARY KEY (Book_Id, Branch_Id, Card_No),

   FOREIGN KEY (Card_No) REFERENCES BORROWER (Card_No)
   ON DELETE CASCADE ON UPDATE CASCADE,

   FOREIGN KEY (Branch_Id) REFERENCES LIBRARY_BRANCH (Branch_Id)
   ON DELETE CASCADE ON UPDATE CASCADE,

   FOREIGN KEY (Book_Id) REFERENCES BOOK (Book_Id)
   ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE BOOK_COPIES
(
   Book_Id          INT             NOT NULL,
   Branch_Id        INT             NOT NULL,
   No_Of_Copies     INT             NOT NULL,

   PRIMARY KEY (Book_Id, Branch_Id),
   FOREIGN KEY (Book_Id) REFERENCES BOOK (Book_Id),
   FOREIGN KEY (Branch_Id) REFERENCES LIBRARY_BRANCH (Branch_Id)
   ON DELETE CASCADE ON UPDATE CASCADE
```

```
);

CREATE Table BOOK_AUTHORS
(
   Book_Id          INT                NOT NULL,
   Author_Name      VARCHAR(30)        NOT NULL,

   PRIMARY KEY (Book_Id, Author_Name),
   FOREIGN KEY (Book_Id) REFERENCES BOOK (Book_Id)
   ON DELETE CASCADE ON UPDATE CASCADE
);
```

Task #2

To insert all of the rows into my tables, I used a method that I found online for importing CSVs files into MySQL using the LOAD DATA command. The command first needs the file path for the CSV file that I wish to import into my tables and then the table I will insert the data into. After that i needed to specify delimiters for how the command will know when to move on to the next attribute/column. I needed to specify that not all commas would be delimiters for the columns so luckily there was a method for ignoring the commas that were surrounded by parentheses. I also needed to specify how each line would be terminated in the CSV file so I had to specify the new line (\n) symbol but also had to specify the carriage return symbol (\r) in the case that the system wouldn't work the same way with new lines. Lastly, I needed to specify for the command to skip the first row of each CSV file because it did not include data that would need to be inserted into the tables. The same template worked for every import so I only needed to change the file paths and the tables that they would be imported into.

Another important thing to note was that I needed to import the files in the correct order so that I would not have any trouble with the foreign key constraints of some of the tables.

CHALLENGES:

Some of the challenges that I faced when trying to find a method for importing the rows into my tables were that I needed to figure out how to set the correct delimiters so that the import wouldn't take into account the commas that were in the address. Another challenge was the whole carriage return situation because it did give me some trouble and i had to figure out how to go about that, i was made aware some time ago that the Mac does some weird stuff with the carriage return and new lines so i needed only to remember that when setting the new line delimiters. The last challenge I had was the order in which I chose to import each of the CSV files because the order did matter because of the foreign key constraints and I was trying to import the BOOKS rows last and it was not going to work in that way.

These are the LOAD DATA commands that I used to populate the rows in the tables. I used these commands after creating each of the tables in order to fill them in with the data from the CSV files.

```
load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Publisher.csv'
into table PUBLISHER
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
ignore 1 rows;

load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Library_Branch.csv'
into table LIBRARY_BRANCH
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
ignore 1 rows;

load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Borrower.csv'
into table BORROWER
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
ignore 1 rows;

load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Book.csv'
into table BOOK
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
ignore 1 rows;

load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Book_Loans.csv'
into table BOOK_LOANS
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
ignore 1 rows;

load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Book_Copies.csv'
into table BOOK_COPIES
fields terminated by ',' enclosed by ""
lines terminated by '\r\n'
```

ignore 1 rows;

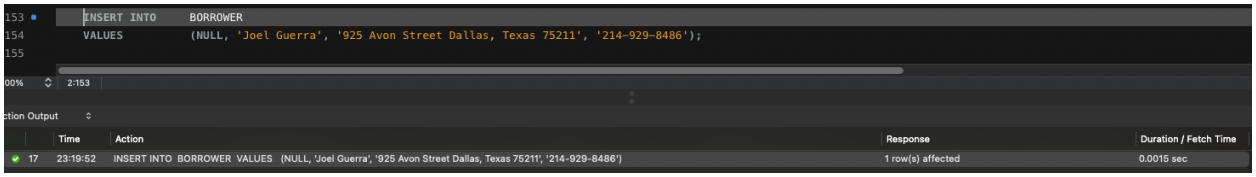
load data local infile '/Users/guerrajoel/Desktop/School/UTA Fall Semester 2023/CSE 3330 - Databases/Project 2/Project 2 Part 2/LMSDataset/Book_Authors.csv'
into table BOOK_AUTHORS
fields terminated by ',' enclosed by '"'
lines terminated by '\r\n'
ignore 1 rows;

```
SELECT COUNT(*) AS COUNT FROM PUBLISHER;          --17
SELECT COUNT(*) AS COUNT FROM LIBRARY_BRANCH;     --3
SELECT COUNT(*) AS COUNT FROM BORROWER;           --21
SELECT COUNT(*) AS COUNT FROM BOOK;               --21
SELECT COUNT(*) AS COUNT FROM BOOK_LOANS;         --21
SELECT COUNT(*) AS COUNT FROM BOOK_COPIES;        --21
SELECT COUNT(*) AS COUNT FROM BOOK_AUTHORS;       --21
```

Task #3

QUESTION (1): Insert yourself as a New Borrower. Do not provide the Card_no in your query.

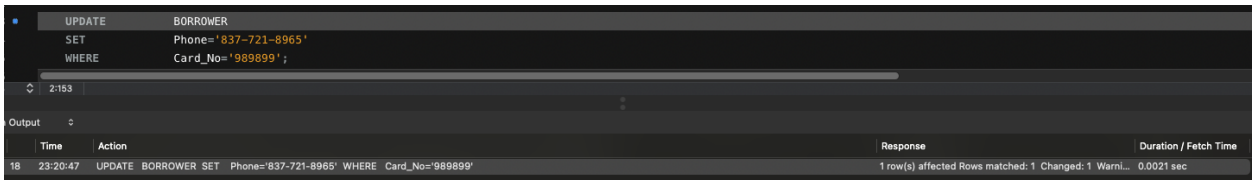
```
INSERT INTO     BORROWER
VALUES          (NULL, 'Joel Guerra', '925 Avon Street Dallas, Texas 75211',
                 '214-929-8486');
```



No output. 1 record inserted.

QUESTION (2): Update your phone number to (837) 721-8965

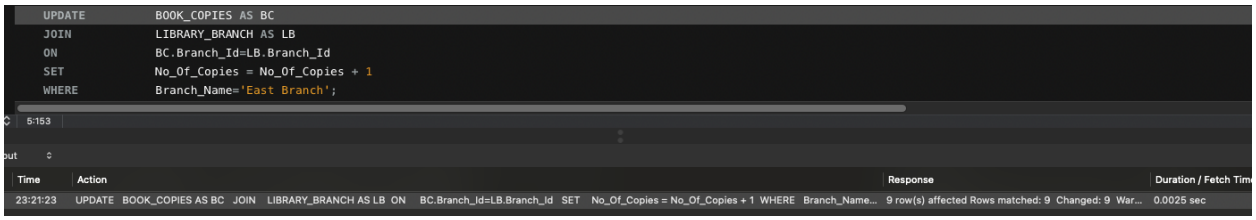
```
UPDATE          BORROWER
SET             Phone='837-721-8965'
WHERE           Card_No='989899';
```



No output. 1 row affected.

QUESTION (3): Increase the number of book_copies by 1 for the 'East Branch'

```
UPDATE      BOOK_COPIES AS BC
JOIN        LIBRARY_BRANCH AS LB
ON          BC.Branch_Id=LB.Branch_Id
SET         No_Of_Copies = No_Of_Copies + 1
WHERE       Branch_Name='East Branch';
```



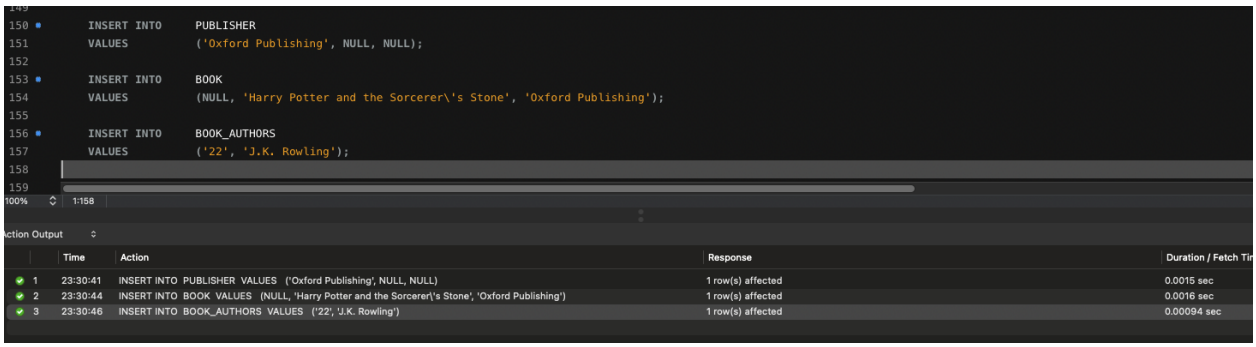
No output. 9 rows affected.

QUESTION (4a): Insert a new BOOK with the following info: Title: ‘Harry Potter and the Sorcerer's Stone’ ; Book_author: ‘J.K. Rowling’ ; Publisher_name: ‘Oxford Publishing'

```
INSERT INTO     PUBLISHER
VALUES          ('Oxford Publishing', NULL, NULL);

INSERT INTO     BOOK
VALUES          (NULL, 'Harry Potter and the Sorcerer\'s Stone', 'Oxford
                Publishing');

INSERT INTO     BOOK_AUTHORS
VALUES          ('22', 'J.K. Rowling');
```



No output. 3 rows inserted.

QUESTION (4b): You also need to insert the following branches:

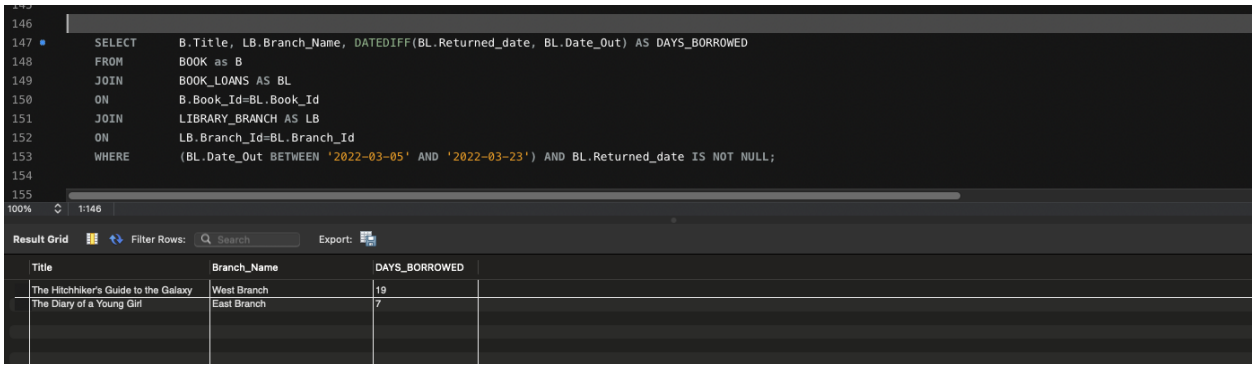
```
INSERT INTO     LIBRARY_BRANCH
VALUES          (NULL, 'North Branch', '456 NW, Irving, TX 76100');

INSERT INTO     LIBRARY_BRANCH
VALUES          (NULL, 'UTA Branch', '123 Cooper St, Arlington TX
                76101');
```

No output. 2 rows inserted.

QUESTION (5): Return all Books that were loaned between March 5, 2022 until March 23, 2022. List Book title and Branch name, and how many days it was borrowed for.

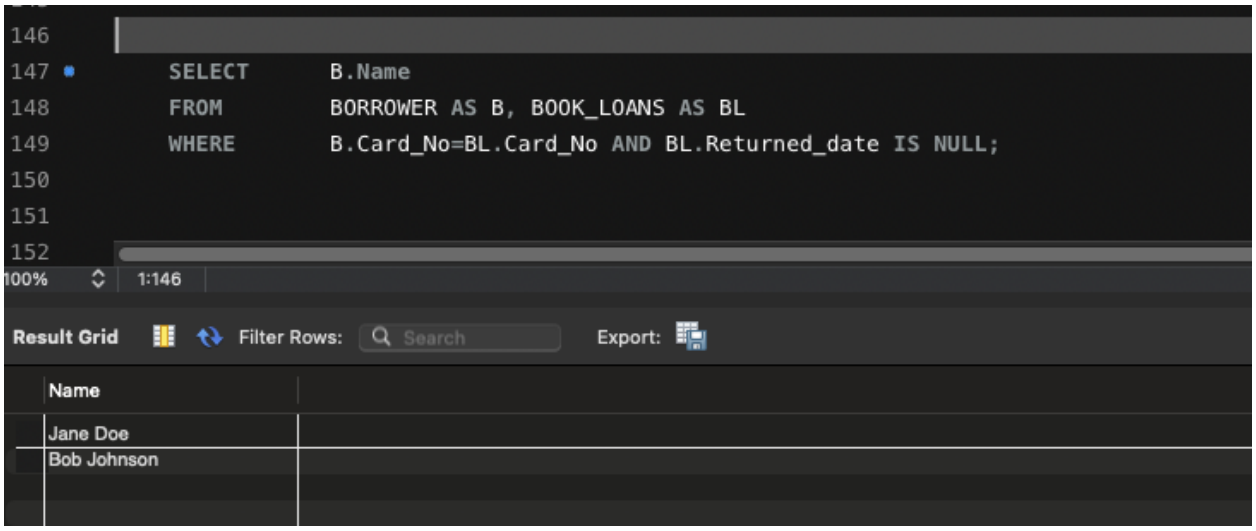
```
SELECT      B.Title, LB.Branch_Name, DATEDIFF(BL.Returned_date,
            BL.Date_Out) AS DAYS_BORROWED
FROM        BOOK as B
JOIN        BOOK_LOANS AS BL
ON          B.Book_Id=BL.Book_Id
JOIN        LIBRARY_BRANCH AS LB
ON          LB.Branch_Id=BL.Branch_Id
WHERE       (BL.Date_Out BETWEEN '2022-03-05' AND '2022-03-23') AND
            BL.Returned_date IS NOT NULL;
```



2 records returned.

QUESTION (6): Return a List borrower names that have books not returned.

```
SELECT      B.Name
FROM        BORROWER AS B, BOOK_LOANS AS BL
WHERE       B.Card_No=BL.Card_No AND BL.Returned_date IS NULL;
```



2 records returned.

QUESTION (7): Create a report that will return all branches with the number of books borrowed per branch separated by if they have been returned, still borrowed, or late.

```
SELECT      LB.Branch_Name, COUNT(BL.Book_Id) AS
            BOOKS_BORROWED,
CASE        WHEN BL.Returned_date IS NOT NULL THEN 'Returned'
            WHEN (BL.Returned_date IS NULL) AND
            (DATEDIFF(BL.Date_Out, BL.Due_Date) < 0) THEN 'Late'
            WHEN BL.Book_Id IS NULL THEN 'No loans'
            ELSE 'Still Borrowed'
END         AS BOOK_STATUS
FROM        LIBRARY_BRANCH AS LB
LEFT JOIN   BOOK_LOANS AS BL
ON          LB.Branch_Id=BL.Branch_Id
GROUP BY    LB.Branch_Name, BOOK_STATUS;
```

```
146
147     SELECT      LB.Branch_Name, COUNT(BL.Book_Id) AS BOOKS_BORROWED,
148     CASE        WHEN BL.Returned_date IS NOT NULL THEN 'Returned'
149                 WHEN (BL.Returned_date IS NULL) AND (DATEDIFF(BL.Date_Out, BL.Due_Date) < 0) THEN 'Late'
150                 WHEN BL.Book_Id IS NULL THEN 'No loans'
151                 ELSE 'Still Borrowed'
152     END         AS BOOK_STATUS
153     FROM        LIBRARY_BRANCH AS LB
154     LEFT JOIN   BOOK_LOANS AS BL
155     ON          LB.Branch_Id=BL.Branch_Id
156     GROUP BY    LB.Branch_Name, BOOK_STATUS;
157
158
```

100% 4:147

Result Grid Filter Rows: Search Export:

Branch_Name	BOOKS_BORROWED	BOOK_STATUS
Main Branch	6	Returned
Main Branch	1	Late
West Branch	1	Late
West Branch	4	Returned
East Branch	9	Returned
North Branch	0	No loans
UTA Branch	0	No loans

7 records returned.

QUESTION (8): List all the books (title) and the maximum number of days that they were borrowed.

```
SELECT      B.Title,
CASE        WHEN (BL.Returned_date IS NOT NULL) THEN
            DATEDIFF(BL.Returned_date, BL.Date_Out)
            ELSE '(Not borrowed)'
            END AS DAYS_BORROWED
FROM        BOOK AS B
LEFT JOIN   BOOK_LOANS AS BL
ON          B.Book_Id=BL.Book_Id;
```

```
146
147     SELECT      B.Title,
148     CASE        WHEN (BL.Returned_date IS NOT NULL) THEN DATEDIFF(BL.Returned_date, BL.Date_Out)
149                 ELSE '(Not borrowed)'
150                 END AS DAYS_BORROWED
151     FROM        BOOK AS B
152     LEFT JOIN   BOOK_LOANS AS BL
153     ON          B.Book_Id=BL.Book_Id;
154
```

100% 5:147

Result Grid Filter Rows: Search Export:

Title	DAYS_BORROWED
To Kill a Mockingbird	31
1984	(Not borrowed)
Pride and Prejudice	(Not borrowed)
The Great Gatsby	31
One Hundred Years of Solitude	35
Animal Farm	35
The Catcher in the Rye	60
Lord of the Flies	61
Brave New World	28
The Picture of Dorian Gray	28
The Alchemist	7
The God of Small Things	7
Wuthering Heights	15
The Hobbit	76
The Lord of the Rings	37
The Hitchhiker's Guide to the Galaxy	19
The Diary of a Young Girl	7
The Da Vinci Code	31
The Adventures of Huckleberry Finn	7
The Adventures of Tom Sawyer	31
A Tale of Two Cities	31
Harry Potter and the Sorcerer's Stone	(Not borrowed)

22 rows returned.

QUESTION (9): Create a report for Ethan Martinez with all the books they borrowed. List the book title and author. Also, calculate the number of days each book was borrowed for and if any book is late being returned. Order the results by the date_out.

```
SELECT          B.Title, BA.Author_Name, DATEDIFF(BL.Returned_date,
                BL.Date_Out) AS Days_Borrowed,
CASE            WHEN (BL.Returned_date IS NULL) AND
                (DATEDIFF(BL.Date_Out, BL.Due_Date) < 0) THEN 'Late'
                ELSE ' '
END AS LATE_OR_NOT
FROM            BOOK AS B
JOIN            BOOK_AUTHORS AS BA
ON              B.Book_Id=BA.Book_Id
JOIN            BOOK_LOANS AS BL
ON              BA.Book_Id=BL.Book_Id
JOIN            BORROWER AS BR
ON              BL.Card_No=BR.Card_No
WHERE           BR.Name='Ethan Martinez'
```

```
ORDER BY        BL.Date_Out ASC;
```

```
146
147     SELECT      B.Title, BA.Author_Name, DATEDIFF(BL.Returned_date, BL.Date_Out) AS Days_Borrowed,
148     CASE        WHEN (BL.Returned_date IS NULL) AND (DATEDIFF(BL.Date_Out, BL.Due_Date) < 0) THEN 'Late'
149                 ELSE ' '
150     END         AS LATE_OR_NOT
151     FROM        BOOK AS B
152     JOIN        BOOK_AUTHORS AS BA
153     ON          B.Book_Id=BA.Book_Id
154     JOIN        BOOK_LOANS AS BL
155     ON          BA.Book_Id=BL.Book_Id
156     JOIN        BORROWER AS BR
157     ON          BL.Card_No=BR.Card_No
158     WHERE       BR.Name='Ethan Martinez'
159     ORDER BY    BL.Date_Out ASC;
160
161
```

100% 3:147

Result Grid Filter Rows: Search Export:

Title	Author_Name	Days_Borrowed	LATE_OR_NOT
The God of Small Things	Arundhati Roy	7	

1 row returned.

QUESTION (10): Return the names of all borrowers that borrowed a book from the West Branch including their addresses.

```
SELECT      B.Name, B.Address
FROM        BORROWER AS B
JOIN        BOOK_LOANS AS BL
ON          B.Card_No=BL.Card_No
JOIN        LIBRARY_BRANCH AS LB
ON          BL.Branch_Id=LB.Branch_Id
WHERE       LB.Branch_Name='West Branch';
```

```
147    SELECT    B.Name, B.Address
148    FROM      BORROWER AS B
149    JOIN      BOOK_LOANS AS BL
150    ON        B.Card_No=BL.Card_No
151    JOIN      LIBRARY_BRANCH AS LB
152    ON        BL.Branch_Id=LB.Branch_Id
153    WHERE     LB.Branch_Name='West Branch';
154
155
```

100% 38:153

Result Grid Filter Rows: Search Export:

Name	Address
Bob Johnson	12 Elm St, Arizona, AR 70345
Emily Lee	389 Oaklay St, Arizona, AR 70986
Michael Park	123 Pinewood St, New Jersey, NJ 32954
Rachel Lee	999 Apple Ave, Arizona, AR 70671
Noah Thompson	189 GreenOak Ave, New Jersey, NJ 32453

5 rows returned.

Contribution List

Task #1 - Trenton Laule Editor: Joel Guerra
Task #2 - Joel Guerra
Task #3 - Joel Guerra Editor: Trenton Laule

Testing - Joel Guerra
Screenshots - Joel Guerra
Documentation - Trenton Laule