**CSE 3330 – 004 (Project 2 – Part 2)**

**GROUP 14 (Araohat Kokate, Inshaad Merchant , Athrva Arora)**

**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

**TABLE OF CONTENTS**

**1) TASK 1 : CREATE TABLE………………………………………………………………….3**

**2) TASK 2 : IMPORT DATA………………………………………………………………….4**

**3) TASK 3 : QUERIES………………………………………………………………………….5**

**4) CONTRIBUTION……………………………………………………………………………18**

**TASK 1: (CREATE TABLE)**

DROP TABLE IF EXISTS LIBRARY\_BRANCH;

CREATE TABLE LIBRARY\_BRANCH(

        Branch\_id           INTEGER             PRIMARY KEY AUTOINCREMENT,

        Branch\_name         VARCHAR(20)         NOT NULL,

        Address             TEXT                NOT NULL

        );

DROP TABLE IF EXISTS PUBLISHER;

CREATE TABLE PUBLISHER(

        Publisher\_name      CHAR(20)            NOT NULL,

        Phone\_no            VARCHAR(13)         NOT NULL,

        Address             TEXT                NOT NULL,

        PRIMARY KEY (Publisher\_name));

DROP TABLE IF EXISTS BOOK;

CREATE TABLE BOOK(

        Book\_id             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Title               VARCHAR(20)         NOT NULL,

        Publisher\_name      VARCHAR(50)         NOT NULL,

        FOREIGN KEY (Publisher\_name) REFERENCES PUBLISHER(Publisher\_name)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

DROP TABLE IF EXISTS BOOK\_AUTHORS;

CREATE TABLE BOOK\_AUTHORS(

        Book\_id             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Author\_name         VARCHAR(50)         NOT NULL

        );

DROP TABLE IF EXISTS BOOK\_COPIES;

CREATE TABLE BOOK\_COPIES(

        Book\_id             INT                 NOT NULL,

        Branch\_id           INT                 NOT NULL,

        No\_of\_Copies        INT                 NOT NULL,

        FOREIGN KEY (Book\_id) REFERENCES BOOK(Book\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Branch\_id) REFERENCES LIBRARY\_BRANCH(Branch\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

DROP TABLE IF EXISTS BORROWER;

CREATE TABLE BORROWER(

        Card\_no             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Name                VARCHAR(30)         NOT NULL,

        Address             TEXT                NOT NULL,

        Phone               VARCHAR(13)         NOT NULL

        );

DROP TABLE IF EXISTS BOOK\_LOANS;

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                NULL,

        FOREIGN KEY (Book\_id)   REFERENCES BOOK(Book\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Branch\_id) REFERENCES LIBRARY\_BRANCH(Branch\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Card\_no)   REFERENCES BORROWER(Card\_no)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

Comments:

* We gave the PRIMARY KEY AUTOINCREMENT for Branch\_id in LIBRARY BRANCH, Book\_id in BOOK and BOOK\_AUTHORS, Card\_no in BORROWER since every entry must have a primary key and we were inserting values that did not have key primary specified
* We have added extra attributes to BOOK\_COPIES as “Branch\_id” and “Id\_no” in the relational schema table above as BOOK\_COPIES is a weak entity and needed to be linked to the LIBRARY\_BRANCHES AND BOOKS table in the relational schema and ER Diagram.
* Card\_no is the primary key for the borrowers as no two different borrowers can have the same Card\_no
* We are assuming Publisher\_name as the primary key for PUBLSIHERS as no two publishers can have same name

**TASK 2: (IMPORTING DATA)**

All the csv files given in the LMSDataset were imported using sqlite3. We used .import method to import the data from the given dataset to their respective tables.

import.sql – File used to import all the dataset into their corresponding tables that we have created in the last step. We are also using ‘ - - skip 1’ which is used to skip the first row of each dataset file which contains column names which are not part of dataset we need to import.

.mode csv

.import --skip 1 Book.csv BOOK

.import --skip 1 Book\_Authors.csv BOOK\_AUTHORS

.import --skip 1 Book\_Copies.csv BOOK\_COPIES

.import --skip 1 Book\_Loans.csv BOOK\_LOANS

.import --skip 1 Borrower.csv BORROWER

.import --skip 1 Library\_Branch.csv LIBRARY\_BRANCH

.import --skip 1 Publisher.csv PUBLISHER

count.sql – File contains all the queries required to find number of records in all the tables in the LMSDataset

SELECT 'BOOK: ', COUNT(\*) FROM BOOK

SELECT 'BOOK\_AUTHORS: ', COUNT(\*) FROM BOOK\_AUTHORS

SELECT 'BOOK\_COPIES: ', COUNT(\*) FROM BOOK\_COPIES

SELECT 'BOOK\_LOANS: ', COUNT(\*) FROM BOOK\_LOANS

SELECT 'BORROWER: ', COUNT(\*) FROM BORROWER

SELECT 'PUBLISHER: ', COUNT(\*) FROM PUBLISHER

SELECT 'LIBRARY\_BRANCH: ', COUNT(\*) FROM LIBRARY\_BRANCH

A computer screen with white text

Description automatically generated

**TASK 3: (QUERIES)**

We are using view.sql to give the output in the respective format with number of entries in the table.

view.sql –

.headers on

.mode column

SELECT \* FROM table\_name;

.headers off

SELECT 'No. of Entries in table\_name:', COUNT(\*) FROM table\_name;

1. Insert yourself as a New Borrower. Do not provide the Card\_no in your query.

query1.sql –

INSERT INTO BORROWER(Name, Address, Phone)

VALUES('Araohat Kokate','587 Spaniolo Dr, Arlington TX, 76010','682-340-0275');

Output – used terminal to run the respective query1.sql

A screenshot of a computer

Description automatically generated

**There were 22 records output to the screen after we added a new record of Araohat Kokate.**

1. Update your phone number to (837) 721-8965

query2.sql-

UPDATE BORROWER

SET Phone = '837-721-8965'

WHERE Name like 'Araohat Kokate';

Output – The Phone number was updated but number of entries remains same as we didn’t add anything to the table.

A screenshot of a computer

Description automatically generated

**There were 22 records output to the screen after we added a new record of Araohat Kokate.**

1. Increase the number of book\_copies by 1 for the ‘East Branch’

Query3.sql –

UPDATE BOOK\_COPIES

SET No\_of\_Copies = No\_of\_Copies + 1

WHERE Branch\_id = (SELECT LB.Branch\_id FROM LIBRARY\_BRANCH AS LB

WHERE LB.Branch\_name = 'East Branch');

Output – Number of Copies were incremented from 4 -> 5 for Branch ”East Branch” with Branch\_id = ‘3’

A screenshot of a computer

Description automatically generated

**The number of records output to the screen were 21 but the variable No\_of\_Copies was increased by 1 where Branch\_id was 3.**

1. A) Question 4-a: Insert a new BOOK with the following info: Title: ‘Harry Potter and the Sorcerer's Stone’ ;Book\_author: ‘J.K. Rowling’ ; Publisher\_name: ‘Oxford Publisheing’

query4a.sql -

INSERT INTO BOOK(Title, Publisher\_name)

VALUES('Harry Potter and the Sorcerer"s Stone','Oxford Publishing');

INSERT INTO BOOK\_AUTHORS(Author\_Name)

VALUES('J.K.Rowling');

Output –

Book Authors After insertion:

A screenshot of a computer

Description automatically generated

**The total number of records increased to 22 after we added a new record for the book “Harry Potter and the Sorcerer’s Stone” whose author was “J.K.Rowling”**

BOOK After insertion –

A screenshot of a computer program

Description automatically generated

4 B) You also need to insert the following branches:

North Branch 456 NW, Irving, TX 76100

UTA Branch 123 Cooper St, Arlington TX 76101

query4b.sql –

INSERT INTO LIBRARY\_BRANCH(Branch\_name, Address)

VALUES('North Branch','456 NW, Irving, TX 76100');

INSERT INTO LIBRARY\_BRANCH(Branch\_name, Address)

VALUES('UTA Branch','123 Cooper St, Arlington TX 76101');

A screenshot of a computer

Description automatically generatedOutput – We have autoincremented the branch\_id for the following new records added –

**The number of records increased from 3 to 5 in LIBRARY\_BRANCH after we inserted two new branches.**

1. 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.

query5.sql –

SELECT B.Title, LB.Branch\_Name,

CAST(JULIANDAY(BL.Returned\_date) AS INTEGER) - CAST(JULIANDAY(BL.Date\_out) AS INTEGER) AS Borrowed\_Days

FROM    BOOK AS B, LIBRARY\_BRANCH  AS LB, BOOK\_LOANS AS BL

WHERE   B.Book\_Id = BL.Book\_Id AND

        BL.Branch\_Id = LB.Branch\_Id AND

        BL.Date\_Out BETWEEN '2022-03-05' AND '2022-03-23' AND

        BL.Returned\_date IS NOT NULL;

Output – We have Changed the Dates to the following format (YYYY– MM- DD) in the file BOOK\_LOANS.csv

A black background with white text

Description automatically generated

No. of Books that were loaned between March 5, 2022, until March 23, 2022.: 2

**The number of records output were 2 books that were loaned out between March 5, 2022 until March 23, 2022.**

1. Return a List borrower names, that have books not returned.

query6.sql

SELECT B.Name

FROM   BORROWER AS B

JOIN   BOOK\_LOANS AS BL ON BL.Returned\_date = 'NULL'

WHERE  B.Card\_no = BL.Card\_no;

Output-

A black screen with a black background

Description automatically generated

**The number of records output to the screen were 2 borrowers who have not returned their books yet.**

1. 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.

query7.sql –

SELECT LIBRARY\_BRANCH.Branch\_Name,

    COUNT(BOOK\_LOANS.Book\_id) AS Num\_of\_Book\_Borrowed,

    COUNT(CASE WHEN DATE('now') >= BOOK\_LOANS.Returned\_date THEN BOOK\_LOANS.Book\_id END) AS Num\_of\_Books\_Returned,

    COUNT(CASE WHEN BOOK\_LOANS.Returned\_date IS NULL OR BOOK\_LOANS.Returned\_date = 'NULL' THEN BOOK\_LOANS.Book\_id END) AS Num\_of\_Books\_Still\_Borrowed,

    COUNT(CASE WHEN BOOK\_LOANS.Returned\_date > BOOK\_LOANS.Due\_Date AND

    BOOK\_LOANS.Returned\_date IS NOT 'NULL' THEN BOOK\_LOANS.Book\_id END) AS

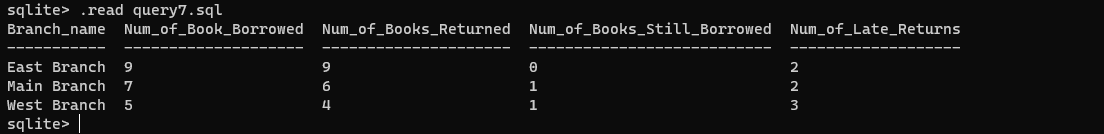
    Num\_of\_Late\_Returns

FROM LIBRARY\_BRANCH, BOOK\_LOANS

WHERE LIBRARY\_BRANCH.Branch\_Id = BOOK\_LOANS.Branch\_Id

GROUP BY LIBRARY\_BRANCH.Branch\_Name;

Output-



**The number of records output to the screen were 3 library branches and it shows all the books for that particular branch that have been returned on-time, returned late, and have not been returned yet.**

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

query8.sql-

SELECT B.Title,

    IFNULL(CAST(JULIANDAY(BL.Returned\_date) AS INTEGER) - CAST(JULIANDAY(BL.Date\_out) AS INTEGER), 0) AS Borrowed\_Days

FROM BOOK AS B

JOIN BOOK\_LOANS AS BL ON B.Book\_Id = BL.Book\_Id;

We used This for reference , <https://www.w3schools.com/sql/sql_isnull.asp> (IFNULL)

Output –

A screenshot of a computer

Description automatically generated

**The number of records output to the screen were 21.**

1. 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.

query9.sql-

SELECT BOOK.Title, BOOK\_AUTHORS.Author\_Name, CAST(JULIANDAY(Returned\_date) AS INTEGER) - CAST(JULIANDAY(Date\_out) AS INTEGER) as Days\_Borrowed,

CASE WHEN BOOK\_LOANS.Returned\_date IS NULL THEN 'YES' WHEN CAST(JULIANDAY(Returned\_date) AS INTEGER) - CAST(JULIANDAY(Due\_Date)AS INTEGER) > 0 THEN 'YES' ELSE 'NO'

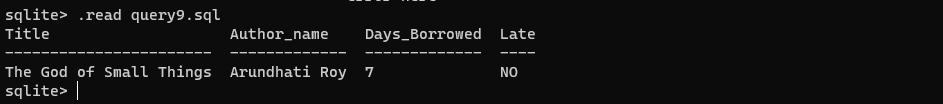
END AS Late

FROM  BORROWER ,BOOK , BOOK\_AUTHORS

JOIN  BOOK\_LOANS ON BORROWER.Card\_no=BOOK\_LOANS.Card\_No

WHERE BORROWER.Name = 'Ethan Martinez' AND BOOK\_LOANS.Book\_id=BOOK.Book\_id AND BOOK.Book\_id=BOOK\_AUTHORS.Book\_Id

ORDER BY BOOK\_LOANS.Date\_Out;

Output – Thre’s only 1 book that Ethan Martinez have borrowed with Card\_no = 444444 and Book\_id = 12 which was returned on the due date.

**The number of records output to the screen was 1 book borrowed by Ethan Martinez.**

1. Return the names of all borrowers that borrowed a book from the West Branch include their addresses.

query10.sql –

SELECT B.Name, B.Address

FROM BORROWER AS B, LIBRARY\_BRANCH AS LB,  BOOK\_LOANS AS BL

WHERE BL.branch\_id = LB.branch\_id AND B.card\_no = BL.card\_no AND

LB.branch\_name = 'West Branch';

Output –

A screenshot of a computer program

Description automatically generated

**The number of records output to the screen was 5 borrowers that borrowed a book from West Branch.**

**CONTRIBUTIONS**

**TASK 1:** Inshaad Merchant

**TASK 2:** Athrva Arora

**TASK 3:** Queries

1) Inshaad Merchant

2) Athrva Arora

3) Araohat Kokate

4a) Inshaad Merchant

4b) Inshaad Merchant

5) Athrva Arora

6) Araohat Kokate

7) Athrva Arora & Araohat Kokate

8) Inshaad Merchant

9) Athrva Arora

10) Araohat Kokate

**Documentation:** Araohat Kokate