**Justin Aegtema**

**CS 6360 Database Design**

**Assignment #3**

1. SQL statement to create the Book\_Loans table:

CREATE TABLE Book\_Loans (

Book\_id INT,

Branch\_id INT,

Card\_no INT,

Date\_out DATE,

Due\_date DATE,

PRIMARY KEY (Book\_id, Branch\_id, Card\_no),

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

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (Branch\_id) REFERENCES Library\_Branch(Branch\_id)

ON DELETE SET NULL

ON UPDATE CASCADE,

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

ON DELETE RESTRICT

ON UPDATE CASCADE

);

Referential integrity constraing justifications:

* If the book, branch, or borrower associated with a loan has their ID updated, it makes sense that the loan should be updated to reflect that. Hence, ON UPDATE CASCADE for all three
* If the book is deleted, it makes sense to delete the loans associated with it. Hence, ON DELETE CASCADE for the book
* If the branch is deleted, it’s loans might not necessarily be invalidated and the records of them still might be valuable. But the branch itself is no longer relevant. Hence ON DELETE SET NULL so that record of the loans remain, but the branch is no longer relevant so it’s part of the record is removed
* If the borrower wishes to be removed from the system, it would not make sense to close out their account while they have outstanding loans. After their loans are closed out and deleted, it is fine for the borrower to be deleted, but not before. Hence, ON DELETE RESTRICT for the borrower

1. Retrieve each book title that has been borrowed by any borrower who lives in ‘Dallas’

SELECT DISTINCT Book.title

FROM Book

JOIN Book\_Loans ON Book.Book\_id = Book\_Loans.Book\_id

JOIN Borrower ON Book\_Loans.Card\_no = Borrower.Card\_no

WHERE Borrower.address LIKE '%Dallas%';

1. Retrieve the name of each publisher that has not published any book that was authored by ‘John Smith’

SELECT DISTINCT Publisher.Name

FROM Publisher

WHERE Publisher.Name NOT IN (

SELECT DISTINCT Book.publisher\_name

FROM Book

JOIN Book\_Authors ON Book.Book\_id = Book\_Authors.Book\_id

WHERE Book\_Authors.Author\_name = 'John Smith'

);

1. Write a SQL statement to retrieve each library branch ID, its name, and the total number of copies (of all of its book titles) it has

SELECT Library\_Branch.Branch\_id, Library\_Branch.Branch\_name, SUM(Book\_Copies.No\_of\_copies) AS Total\_Copies

FROM Library\_Branch

JOIN Book\_Copies ON Library\_Branch.Branch\_id = Book\_Copies.Branch\_id

GROUP BY Library\_Branch.Branch\_id, Library\_Branch.Branch\_name;

1. Write a SQL command to increase the number of copies by 10 for each book that was loaned to more than 9 borrowers

UPDATE Book\_Copies

SET No\_of\_copies = No\_of\_copies + 10

WHERE Book\_id IN (

SELECT Book\_id

FROM Book\_Loans

GROUP BY Book\_id

HAVING COUNT(DISTINCT Card\_no) > 9

);

1. Write a SQL assertion to prevent ny borrower from borrowing more than 10 books at any time.

CREATE ASSERTION Borrower\_Book\_Limit

CHECK (

NOT EXISTS (

SELECT 1

FROM Book\_Loans

GROUP BY Card\_no

HAVING COUNT(\*) > 10

)

);