CREATE DATABASE db\_library;

CREATE TABLE Library\_Branch (

BranchID INT PRIMARY KEY NOT NULL IDENTITY (100,1),

BranchName VARCHAR(100) not null,

Address VARCHAR(200) not null

);

INSERT INTO Library\_Branch (BranchName, Address)

VALUES

('Sharpstown', '9887 St.'),

('Central', '300 Star Ave.'),

('Southern', '500 Moon St.'),

('Northern', '350 Disney Ave.'),

('Eastern', '900 Madrid St.'),

('Sky Lab', '111 Sky St.')

;

SELECT \* FROM Library\_Branch;

------------------------------------------------------------------

CREATE TABLE Publisher (

PublisherName VARCHAR(50) PRIMARY KEY NOT NULL,

Address VARCHAR(200) NOT NULL,

Phone VARCHAR(15) NOT NULL,

);

INSERT INTO Publisher(PublisherName, Address, Phone)

VALUES

('Harvard', 'Massachusets, USA', '598-478-9811'),

('Cambridge Press', 'Cambridge, UK', '400-455-8755'),

('Calgary Press', 'Calgary, Canada', '579-987-0987'),

('UBC Press', 'Vancouver, Canada', '214-879-8900'),

('McGregor Book', 'Dublin, Ireland', '598-478-9811'),

('Lemonde', 'Paris, France', '400-455-8755'),

('Xinhua', 'Beijing, China', '579-987-0987'),

('Vodka Books', 'Moscow, Russia', '214-879-8900'),

('Amigo', 'Madrid, Spain', '598-478-9811'),

('Terminator Press', 'Berlin, Germany', '400-455-8755')

;

SELECT \* FROM Publisher;

---------------------------------------------------------------------

CREATE TABLE Books (

BookID INT PRIMARY KEY NOT NULL IDENTITY (1,1),

Title VARCHAR(50) NOT NULL,

PublisherName VARCHAR(50) not null constraint fk\_PublisherName foreign key references Publisher(PublisherName) on update cascade on delete cascade,

);

INSERT INTO Books (Title, PublisherName)

VALUES

('The Lost Tribe', 'Amigo'),

('Hello World', 'Harvard'),

('How to run the World', 'Harvard'),

('How to make Cheese', 'Lemonde'),

('How to eat Frogs', 'Lemonde'),

('Politics of Russia', 'Vodka Books'),

('AI for Dummies', 'Vodka Books'),

('Big Data', 'McGregor Book'),

('How to Fight', 'McGregor Book'),

('Hello SQL', 'McGregor Book'),

('Red Whine', 'Amigo'),

('Red Beans', 'Amigo'),

('The Last Emperror', 'Xinhua'),

('Pekin Duck', 'Xinhua'),

('Python for Everybody', 'Calgary Press'),

('Judo', 'Calgary Press'),

('How to never Age?', 'UBC Press'),

('How to Make Cake', 'Terminator Press'),

('World History', 'Cambridge Press'),

('How to get Rich?', 'Cambridge Press')

;

SELECT \* FROM Books;

--------------------------------------------------------------------------------

CREATE TABLE Book\_Authors (

BookID int not null constraint fk\_BookID\_Authors foreign key references Books(BookID) on update cascade on delete cascade,

AuthorName varchar(60) not null

);

INSERT INTO Book\_Authors(BookID, AuthorName)

VALUES

(1, 'Wilbur Smith'),

(2, 'Wilbur Smith'),

(3, 'Jack London'),

(4, 'Ernest Hemingway'),

(5, 'Leo Dostoesky'),

(6, 'Leo Dostoevsky'),

(7, 'Alexander Pushkin'),

(8, 'Stephen King'),

(9, 'William Shakespear'),

(10, 'Katano Arigato')

;

SELECT \* FROM Book\_Authors;

---------------------------------------------------------------------------------------------

CREATE TABLE Book\_Copies (

BookID int not null constraint fk\_BookID\_Copies foreign key references Books(BookID) on update cascade on delete cascade,

BranchID int not null constraint fk\_BranchID\_Copies foreign key references Library\_Branch(BranchID) on update cascade on delete cascade,

Number\_Of\_Copies int not null

);

INSERT INTO Book\_Copies(BookID, BranchID, Number\_Of\_Copies)

VALUES

(1, 100, 2),

(2, 100, 6),

(3, 100, 3),

(4, 100, 5),

(7, 100, 2),

(17, 100, 15),

(8, 100, 3),

(9, 100, 9),

(20, 100, 7),

(5, 100, 2),

(1, 101, 5),

(3, 101, 6),

(5, 101, 8),

(7, 101, 6),

(9, 101, 7),

(11, 101, 8),

(13, 101, 2),

(15, 101, 4),

(17, 101, 2),

(19, 101, 3),

(3, 102, 6),

(8, 102, 20),

(5, 102, 4),

(9, 102, 2),

(10, 102, 2),

(13, 102, 2),

(18, 102, 2),

(1, 102, 2),

(2, 102, 2),

(12, 102, 2),

(11, 102, 2),

(4, 102, 2),

(20, 102, 2),

(1, 103, 2),

(2, 103, 7),

(3, 103, 8),

(4, 103, 2),

(5, 103, 2),

(6, 103, 2),

(7, 103, 3),

(8, 103, 6),

(9, 103, 2),

(10, 103, 2),

(11, 103, 2),

(12, 103, 2),

(13, 103, 4),

(14, 103, 3),

(15, 103, 9),

(16, 103, 20),

(17, 103, 2),

(18, 103, 3)

;

SELECT \* FROM Book\_Copies;

--------------------------------------------------------------------------

CREATE TABLE Borrower (

CardNo int primary key not null identity (1000,1),

Name varchar(50) not null,

Address varchar(50) not null,

Phone varchar(15) not null

);

INSERT INTO Borrower(Name, Address, Phone)

VALUES

('Bruce Lee', '19 London St.', '508-159-0000'),

('Jacky Chan', '876 Rich Ave.', '504-134-0989'),

('Bruce Willis', '987 Italian St.', '133-986-7890'),

('Keanu Reeves', '98 Aussi Way', '300-599-7296'),

('Kate Middleton', '23 New Way', '420-342-1232'),

('Julia Roberts', '098 Lemon St.', '434-509-0001'),

('Nicole Kidman', '493 Kangaroo St.', '410-917-8177'),

('Pamela Anderson', '78 East Coast Av.', '098-589-8900')

;

SELECT \* FROM Borrower;

-------------------------------------------------------------------------------------

CREATE TABLE Book\_Loans (

BookID int not null constraint fk\_BookID\_Loans foreign key references Books(BookID) on update cascade on delete cascade,

BranchID int not null constraint fk\_BranchID\_Loans foreign key references Library\_Branch(BranchID) on update cascade on delete cascade,

CardNo int not null constraint fk\_CardNo foreign key references Borrower(CardNo) on update cascade on delete cascade,

DateOut date not null,

DateDue date not null

);

INSERT INTO Book\_Loans

(BookID, BranchID, CardNo, DateOut, DateDue)

VALUES

(1, 100, 1000, '2019-03-03', '2019-06-03'),

(2, 100, 1000, '2019-03-03', '2019-06-03'),

(3, 100, 1000, '2019-03-03', '2019-06-03'),

(4, 100, 1000, '2019-03-03', '2019-06-03'),

(7, 100, 1000, '2019-03-03', '2019-06-03'),

(17, 100, 1000, '2019-03-03', '2019-06-03'),

(1, 101, 1001, '2019-03-03', '2019-06-03'),

(3, 101, 1001, '2019-03-03', '2019-06-03'),

(5, 101, 1001, '2019-03-03', '2019-06-03'),

(7, 101, 1001, '2019-03-03', '2019-06-03'),

(9, 101, 1001, '2019-03-03', '2019-06-03'),

(11, 101, 1001, '2019-03-03', '2019-06-03'),

(13, 101, 1001, '2019-03-03', '2019-06-03'),

(15, 101, 1001, '2019-03-03', '2019-06-03'),

(17, 101, 1001, '2019-03-03', '2019-06-03'),

(3, 102, 1002, '2019-03-03', '2019-06-03'),

(8, 102, 1002, '2019-03-03', '2019-06-03'),

(5, 102, 1002, '2019-03-03', '2019-06-03'),

(9, 102, 1002, '2019-03-03', '2019-06-03'),

(10, 102, 1002, '2019-03-03', '2019-06-03'),

(13, 102, 1002, '2019-03-03', '2019-06-03'),

(18, 103, 1003, '2019-03-03', '2019-06-03'),

(3, 103, 1003, '2019-03-03', '2019-06-03'),

(1, 103, 1003, '2019-03-03', '2019-06-03'),

(1, 100, 1007, '2019-03-03', '2019-06-03'),

(19, 101, 1005, '2019-03-03', '2019-06-03'),

(1, 101, 1005, '2019-03-03', '2019-06-03'),

(18, 102, 1006, '2019-03-03', '2019-06-03'),

(1, 102, 1006, '2019-03-03', '2019-06-03'),

(2, 102, 1006, '2019-03-03', '2019-06-03'),

(12, 102, 1006, '2019-03-03', '2019-06-03'),

(11, 102, 1006, '2019-03-03', '2019-06-03'),

(2, 103, 1007, '2019-03-03', '2019-06-03'),

(3, 103, 1007, '2019-03-03', '2019-06-03'),

(8, 100, 1000, '2019-03-09', '2019-06-09'),

(3, 101, 1001, '2019-03-09', '2019-06-09'),

(5, 101, 1007, '2019-03-09', '2019-06-09'),

(4, 102, 1006, '2019-03-09', '2019-06-09'),

(20, 102, 1002, '2019-03-09', '2019-06-09'),

(8, 102, 1002, '2019-03-09', '2019-06-09'),

(5, 102, 1002, '2019-03-09', '2019-06-09'),

(7, 101, 1007, '2019-03-05', '2019-06-05'),

(3, 102, 1006, '2019-03-05', '2019-06-05'),

(4, 103, 1007, '2019-03-05', '2019-06-05'),

(8, 102, 1002, '2019-03-05', '2019-06-05'),

(5, 103, 1003, '2019-03-05', '2019-06-05'),

(9, 100, 1005, '2019-03-05', '2019-06-05'),

(20, 100, 1005, '2019-03-05', '2019-06-05'),

(5, 100, 1005, '2019-03-04', '2019-06-04'),

(2, 100, 1000, '2019-03-04', '2019-06-04'),

(17, 100, 1000, '2019-03-04', '2019-06-04')

;

SELECT \* FROM Book\_Loans;

-------------------------------------------------------------------------------------------

SELECT \*

FROM ((Book\_Loans FULL OUTER JOIN Borrower ON Book\_Loans.CardNo = Borrower.CardNo)

FULL OUTER JOIN Books ON Book\_Loans.BookID = Books.BookID);

----------------------------------------------------------------------------------------------

go

-- Stored Procedure--Number of books titled 'The Lost Tribe' owned by 'Sharpstown' library branch

create proc dbo.BookCount @Title varchar(50), @Branch varchar(50)=null

as

select

a1.Title, a2.BranchName 'Branch', a3.Number\_Of\_Copies 'Copies'

from Book\_Copies a3

inner join Books a1 on a1.BookID = a3.BookID

inner join Library\_Branch a2 on a2.BranchID = a3.BranchID

where a1.Title like @Title + '%'

and a2.BranchName = isnull(@Branch,a2.BranchName)

go

exec BookCount @Title = 'The Lost Tribe', @Branch = 'Sharpstown'

---------------------------------------------------------------------------------------------------------

--Number of copies of 'The Lost Tribe' at each Libary Branch

exec BookCount @Title = 'The Lost Tribe'

go

-- Borrowers who did not check out books

create proc dbo.SeeNoLoans

as

select a1.Name from Borrower a1

full outer join Book\_Loans a2 on a2.CardNo = a1.CardNo

where a2.BookID IS NULL

go

exec SeeNoLoans

-----------------------------------------------------------------------------------------------------------