DataBase Project

Presented for :

Dr.A.H.El-Hosseini

Presented By :

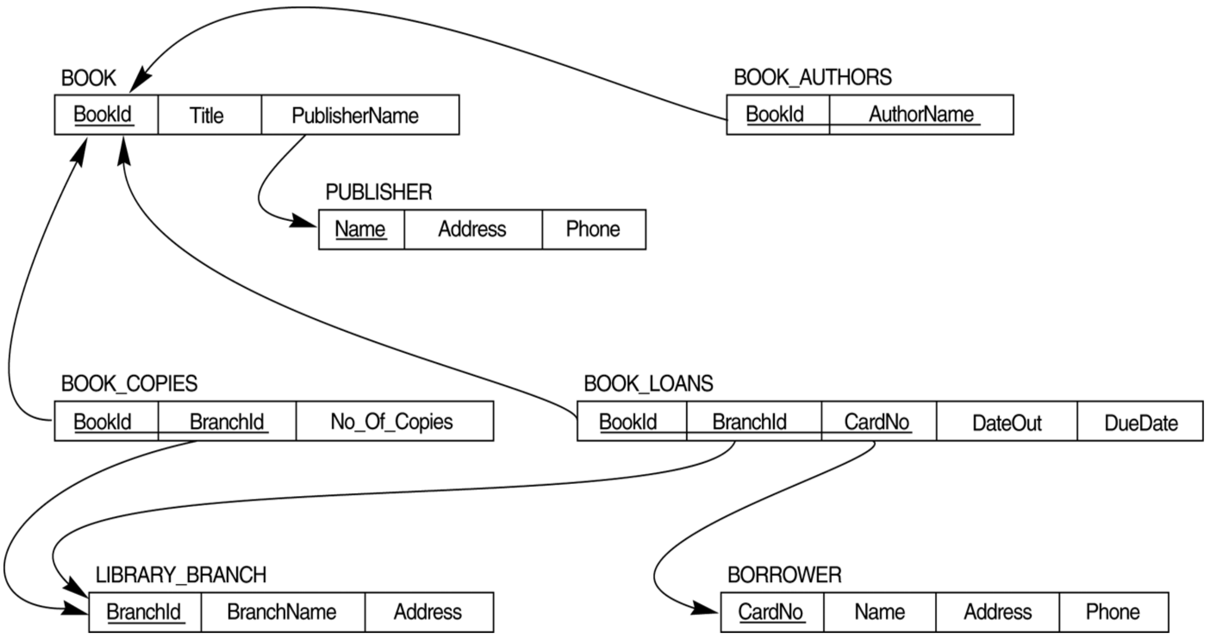
Mohammad Haidar 3967

Data Base description:

This database is for a library that rent books, and it has multiple banches,

We need to organize the books and the renting operation by creating a simple and clear and easy to use database.

Tables architecture:



Creation of tables:

- CREATE TABLE BOOK (`BookId` INT NOT NULL,`Title` VARCHAR(45) NULL, `PublisherName` VARCHAR(45) NULL PRIMARY KEY (`BookId`));

- CREATE TABLE BOOK_AUTHORS (`BookId` INT NOT NULL,`AuthorName` VARCHAR(45) NOT NULL, PRIMARY KEY (`BookId`));

- CREATE TABLE PUBLISHER (`Name` VARCHAR(45) NOT NULL, `Address` VARCHAR(45) NULL , `Phone` VARCHAR(45) NULL PRIMARY KEY (`Name`));

- CREATE TABLE BOOK_COPIES (`BookId` INT NOT NULL, `BranchId` INT NOT NULL ,`No_Of_Copies` INT NULL PRIMARY KEY (`BookId`));

- CREATE TABLE BOOK_LOANS (`BookId` INT NOT NULL, `BranchId` INT NOT NULL , `CardNo` INT NOT NULL, `DateOut ` VARCHAR(45) NULL, `DueDate` VARCHAR(45) NULL PRIMARY KEY (`BookId`));

- CREATE TABLE LIBRARY_BRANCH (`BranchId` INT NOT NULL, `BranchName` VARCHAR(45) , `Address` VARCHAR(45) NULL NULL PRIMARY KEY (`BranchId`));

- CREATE TABLE BORROWER `CardNo` INT NOT NULL, `Name` VARCHAR(45) , `Address` VARCHAR(45) NULL NULL, `Phone` VARCHAR(45) PRIMARY KEY (`CardNo `));

Adding Foreign Keys :

We add the Foreign Keys after inserting elements into he database , else we need to disable the foreign keys to insert new elements.

We add elements with the following statement :

“ INSERT INTO table_name (column1,column2,column3,...)

VALUES (value1,value2,value3,...); “

- ALTER TABLE BOOK_AUTHORS ADD CONSTRAINT fk_ BOOK_AUTHORS FOREIGN KEY (BookId) REFERENCES BOOK(BookId);
- ALTER TABLE BOOK ADD CONSTRAINT fk_ BOOK FOREIGN KEY (PublisherName) REFERENCES PUBLISHER (Name);

- ALTER TABLE BOOK_COPIES ADD CONSTRAINT fk_ BOOK_COPIES_1 FOREIGN KEY (BookId) REFERENCES BOOK (BookId);
- ALTER TABLE BOOK_COPIES ADD CONSTRAINT fk_ BOOK_COPIES_2 FOREIGN KEY (BranchId) REFERENCES LIBRARY_BRANCH (BranchId);

- ALTER TABLE BOOK_LOANS ADD CONSTRAINT fk_ BOOK_LOANS_1 FOREIGN KEY (BookId) REFERENCES BOOK (BookId);
- ALTER TABLE BOOK_LOANS ADD CONSTRAINT fk_ BOOK_LOANS_2 FOREIGN KEY (BranchId) REFERENCES LIBRARY_BRANCH (BranchId);

- ALTER TABLE BOOK_LOANS ADD CONSTRAINT fk_ BOOK_LOANS_3 FOREIGN KEY (CardNo) REFERENCES BORROWER (CardNo);

Searching for elements in the DataBase:

1. How many copies of the book titled “X” are owned by the library branch whose name is “Y” :

```
    •  SELECT bc.No_Of_Copies
FROM BOOK b, BOOK_COPIES bc, LIBRARY_BRANCH bl
WHERE       b.BookId = bc.BookId AND
            bc.BranchId = bl.BranchId AND
Title=X AND BranchName=Y;
```

2. How many copies of the book titled “X” are owned by each library branch :

```
    •  SELECT BranchName, No_Of_Copies
FROM ((BOOK NATURAL JOIN BOOK_COPIES ) NATURAL JOIN
        LIBRARY_BRANCH )
WHERE Title=X;
```

3. Retrieve the names of all borrowers who do not have any books checked out .

```
    •  SELECT       Name
FROM        BORROWER B
WHERE       CardNo NOT IN (SELECT CardNo FROM BOOK_LOANS );
```

4. For each book that is loaned out from the X branch and whose DueDate is today, retrieve the book title, the borrower's name, and the borrower's address.

```
    •  SELECT B.Title, R.Name, R.Address
FROM BOOK B, BORROWER R, BOOK_LOANS BL, LIBRARY_BRANCH LB
WHERE LB.BranchName=X AND LB.BranchId=BL.BranchId AND
BL.DueDate='today' AND BL.CardNo=R.CardNo AND BL.BookId=B.BookId;
```

5. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

```
    SELECT      L.BranchName, COUNT(*)
FROM          LIBRARY_BRANCH L, BOOK_LOANS BL
WHERE         BL.BranchId = L.BranchId  GROUP BY   L.BranchName;
```

6. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

```
    SELECT      B.Name, B.Address, COUNT(*)
FROM          BORROWER B, BOOK_LOANS L
WHERE         B.CardNo = L.CardNo
GROUP BY      B.CardNo, B.Name, B.Address
HAVING        COUNT(*) > 5;
```

7. For each book authored (or co-authored) by X retrieve the title and the number of copies owned by the library branch whose name is Y

```
    SELECT      Title, No_Of_Copies
FROM   (((BOOK_AUTHORS NATURAL JOIN BOOK) NATURAL JOIN
BOOK_COPIES) NATURAL JOIN LIBRARY_BRANCH)WHERE Author_Name= X
AND BranchName= Y;
```