E44065020_何子安_HW2

Tags | Database System Homework #2 (Chapter 4)

> This report is written in Notion, and being export as PDF.

4.7. Consider the LIBRARY relational database schema shown in Figure 4.6. Choose the appropriate action (reject, cascade, set to NULL, set to default) for each referential integrity constraint, both for the deletion of a referenced tuple and for the update of a primary key attribute value in a referenced tuple. Justify your choices.

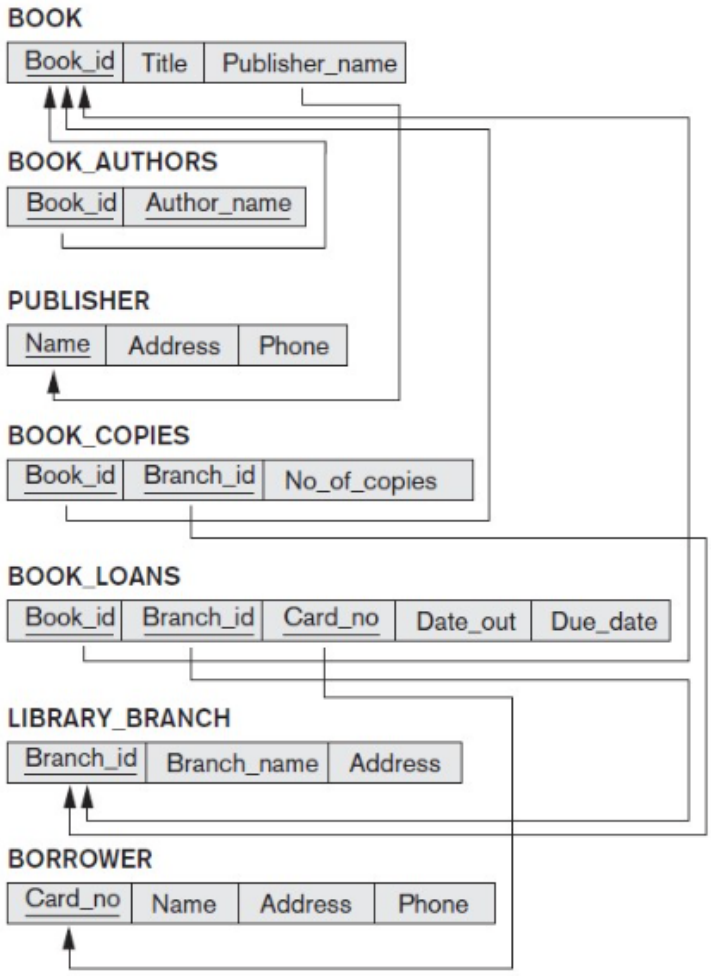

Figure 4.6
A relational database schema for a LIBRARY database.

a. BOOK.PublisherName --> PUBLISHER.Name
   REJECT on DELETE, because we should not delete a PUBLISHER tuple which

has existing BOOK tuples that reference the PUBLISHER.
CASCADE on UPDATE, if a PUBLISHER's Name is updated, the change should be propagated automatically to all referencing BOOK tuples.

b. BOOK_LOANS.BookId --> BOOK.BookId
CASCADE on both DELETE or UPDATE, if a BOOK is deleted, or the value of its BookId is updated/changed, the deletion or change is automatically propagated to the referencing BOOK_LOANS tuples.

c. BOOK_COPIES.BookId --> BOOK.BookId
CASCADE on both DELETE or UPDATE, if a BOOK is deleted, or the value of its BookId is updated/changed, the deletion or change is automatically propagated to the referencing BOOK_COPIES tuples.

d. BOOK_LOANS.BranchId --> LIBRARY_BRANCH.BranchId
CASCADE on both DELETE or UPDATE, if a LIBRARY_BRANCH is deleted, or the value of its BranchId is updated/changed, the deletion or change is automatically
propagated to the referencing BOOK_LOANS tuples.

4.8. Write appropriate SQL DDL statements for declaring the LIBRARY relational database schema of Figure 4.6. Specify the keys and referential triggered actions.

```
CREATE TABLE BOOK (
  BookId CHAR(20) NOT NULL,
  Title VARCHAR(30) NOT NULL,
  PublisherName VARCHAR(20),
  PRIMARY KEY (BookId),
  FOREIGN KEY (PublisherName) REFERENCES PUBLISHER (Name) ON UPDATE CASCADE);
```

```
CREATE TABLE BOOK_AUTHORS (
  BookId CHAR(20) NOT NULL,
  AuthorName VARCHAR(30) NOT NULL,
  PRIMARY KEY (BookId, AuthorName),
  FOREIGN KEY (BookId) REFERENCES BOOK (BookId) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE PUBLISHER (
  Name VARCHAR(20) NOT NULL,
  Address VARCHAR(40) NOT NULL,
  Phone CHAR(12),
  PRIMARY KEY (Name));
```

```
CREATE TABLE BOOK_COPIES (
  BookId CHAR(20) NOT NULL,
  BranchId INTEGER NOT NULL,
  No_Of_Copies INTEGER NOT NULL,
  PRIMARY KEY (BookId, BranchId),
  FOREIGN KEY (BookId) REFERENCES BOOK (BookId) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (BranchId) REFERENCES BRANCH (BranchId) ON DELETE CASCADE ON UPDATE CASC
ADE);
```

```
CREATE TABLE BOOK_LOANS (
  CardNo INTEGER NOT NULL,
  BookId CHAR(20) NOT NULL,
  BranchId INTEGER NOT NULL,
  DateOut DATE NOT NULL,
  DueDate DATE NOT NULL,
  PRIMARY KEY (CardNo, BookId, BranchId),
  FOREIGN KEY (CardNo) REFERENCES BORROWER (CardNo)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (BranchId) REFERENCES LIBRARY_BRANCH (BranchId)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (BookId) REFERENCES BOOK (BookId)
  ON DELETE CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE LIBRARY_BRANCH (
  BranchId INTEGER NOT NULL,
  BranchName VARCHAR(20) NOT NULL,
  Address VARCHAR(40) NOT NULL,
  PRIMARY KEY (BranchId));
```

```
CREATE TABLE BORROWER (
  CardNo INTEGER NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Address VARCHAR(40) NOT NULL,
  Phone CHAR(12),
  PRIMARY KEY (CardNo));
```

4.12. Specify the following queries in SQL on the database schema of Figure 1.2.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

a.

```
SELECT Name FROM STUDENT
WHERE Major='CS'
```

b.

```
SELECT Course_name
FROM COURSE, SECTION
WHERE
  COURSE.Course_number=SECTION.Course_number AND
  Instructor='King' AND
  (Year='07' OR Year='08')
```

c.

```
SELECT Course_number, Semester, Year, COUNT(*)
FROM SECTION, GRADE_REPORT
WHERE Instructor='King' AND SECTION.Section_identifier=GRADE_REPORT.Section_identifier
GROUP BY Course_number, Semester, Year
```

d.

```
SELECT Name, Course_name, C.Course_number, Credit_hours, Semester, Year, Grade
FROM STUDENT ST, COURSE C, SECTION S, GRADE_REPORT G
WHERE
  Class=4 AND
  Major='CS' AND
  ST.Student_number=G.Student_number AND
```

```
  G.Section_identifier=S.Section_identifier AND
  S.Course_number=C.Course_number
```

4.13. Write SQL update statements to do the following on the database schema shown in Figure 1.2.

a.

```
INSERT INTO STUDENT
VALUES ('Johnson', 25, 1, 'MATH')
```

b.

```
UPDATE STUDENT
SET CLASS = 2
WHERE Name='Smith'
```

c.

```
INSERT INTO COURSE
VALUES ('Knowledge Engineering','CS4390', 3,'CS')
```

d.

```
DELETE FROM STUDENT
WHERE Name='Smith' AND Student_number=17
```