Database and File Systems Design Project

Professor Brian Howard

Murtaza Hassan

This is an adaptation of the Prevo Science Library at DePauw University where any student can rent books. The design focuses on database model where a student rents books required by the curriculum for the specific course book.

Diagram

Description automatically generated

Figure 1: Prevo Science UML Diagram

I have used arrowheads to represent foreign key relationships. Each relationship is a (many to one) strong-weak relationship. Each student respectively will have a student id, full name, email, phone number, and a department id (major) where they can rent as many books as they want. Each book has an isbn, title, author, and department id (major). The Book table is connected to Checkout table via isReturn field which tells us weather a book is in the library or a student has rented it. The same book can be rented again after checking in after a duration of at most seven days. Book table has a foreign key that connect it with Department table. It tells us about all the books of a specified major. Once a book is rented, isReturn value would be false that would be connected with a student. The Book table is also connected to the checkout table. After all the books are selected, students can rent the books from the library. All the books can be rented at a single time/date.

For the database to be efficient we have added some constraints in it. Every value is assigned a specific variable type to save the resources. Furthermore, each value is set to not null because every attribute is important for the database to be efficient and robust. Every table is assigned with a primary key. A specific constraint has been added to ensure that the student enters a valid DePauw email address.

Example Queries:

1. List the authors of all the books a student has selected for a specific major.
2. Given a student Id, be able to retrieve the books he/she has rented.
3. List all the books of a specific department.

Here is the SQL schema obtained from the UML diagram in Figure 1.

create table Student (

id integer not null,

name varchar (100) not null,

email varchar (100) not null,

phone varchar (12) not null,

primary key (studentId),

check (email like ‘%@depauw.edu’)

);

create table Book (

ISBN integer not null,

title varchar (150) not null,

author varchar (100) not null,

department\_id not null,

checkout\_isReturn Boolean,

primary key (ISBN),

foreign key (Department\_id) reference major

on delete no action,

);

create table Department (

id integer not null,

name varchar (10) not null,

location varchar (10) not null,

primary key(id),

);

create table Checkout (

id integer not null,

student\_id integer not null,

book\_isbn integer not null,

returnDate date not null,

isReturn Boolean not null,

primary key (id),

foreign key (student\_id) references Student

foreign key (Checkout\_isReturn) reference Book

on delete no action,

);