

Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Проект по Базии данни

Тема: Библиотека

Съставил:

Ростислав Стоянов

ф-н 45244

Съдържание

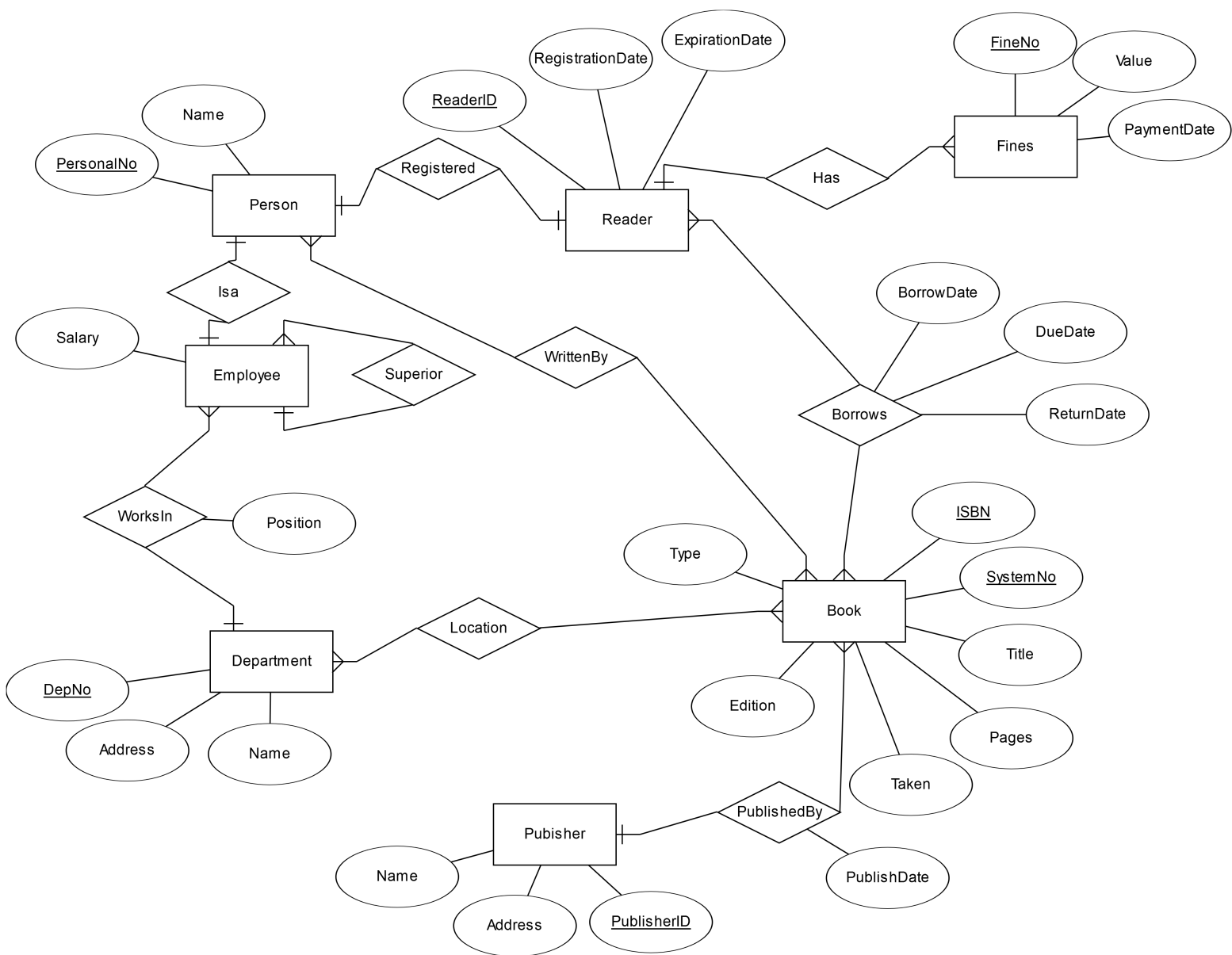
1	Описание	2
2	Е\R модел	3
3	Релационен модел	4
4	Функции	8
5	Тригери	8
6	Изгледи	9
7	Приложение за достъп до базата данни	9

1 Описание

Всяка библиотека има клонове. Всеки клон се описва с уникален номер, име и адрес. Във всеки клон работят различни служители - всеки от тях се описва с ЕГН, име и позиция. Служителите могат да заемат различни позиции, като всеки служител работи в определен клон и си има ръководител (освен ръководителя на библиотеката). Във всеки клон на библиотеката се съхраняват книги, като в различните клонове може да се съхраняват различни книги, а една и съща книга може да бъде намерена на повече от едно място. Атрибутите, описващи една книга са: ISBN, име, автор, издателство, година на издаване, издание, страници, системен номер, тип. Издателствата се характеризират с индивидуален номер, име и адрес. Книгите могат да бъдат заемани за определен период (ако са налични) от граждани. Ако книгата не се върне в посочения срок, се начислява глоба. Гражданите се описват с ЕГН и име.

2 E\R модел

E\R модел, илюстриращ база данни, спазваща изискванията от описанието, е представен на следната диаграма:



Фигура 1: E\R диаграма, представяща описания модел

3 Релационен модел

Използвайки NULL подхода получаваме следните релации :

Person(PersonalNo, Name, Salary);
Reader(ReaderID, RegistrationDate, ExpirationDate);
Fines(FineNo, Value, PaymentDate);
Book(SystemNo, ISBN, Title, Pages, Edition, Type);
Publisher(PublisherID, Name, Address);
Department(DepNo, Name, Address);
Registered(PersonalNo, ReaderID);
Superior(PersonalNo, SuperiorPersonalNo;
WorksIn(PersonalNo, DepNo, Position);
WrittenBy(SystemNo, PersonalNo);
Borrows(ReaderID, SystemNo, BorrowDate, DueDate, ReturnDate),
PublishedBy(SystemNo, PublisherID);
Location(DepNo, SystemNo);
Has(ReaderID, FineNo, Value, PaymentDate).

Забелязваме, че WorksIn, Has, Superior и PublishedBy са от вида много - един, поради което оптимизираме релационния модел, и получаваме:

Person(PersonalNo, Name, Salary, SuperiorPersonalNo, DepNo, Position);
Reader(ReaderID, RegistrationDate, ExpirationDate);
Fines(FineNo, Value, PaymentDate, ReaderID);
Book(SystemNo, ISBN, Title, Pages, Edition, Type, PublisherID, PublishDate);
Publisher(PublisherID, Name, Address);
Department(DepNo, Name, Address);
Registered(PersonalNo, ReaderID);
WrittenBy(SystemNo, PersonalNo);
Borrows(ReaderID, SystemNo, BorrowDate, DueDate, ReturnDate);
Location(DepNo, SystemNo).

Във всяка една от релациите, компонентите са атомарни, което означава че всички релации са в 1НФ. Търсим функционални зависимости с цел да нормализираме получения релационен модел, ако е необходимо:

1. Person:

- $PersonalNo \rightarrow Name, Salary, SuperiorPersonalNo, DepNo, Position$
- $PersonalNo, Name, DepNo, Position \rightarrow Salary, SuperiorPersonalNo$
- $PersonalNo, DepNo, Position \rightarrow Name, Salary, SuperiorPersonalNo$

За да е релацията в 2НФ, то всеки атрибут трябва да е функционално зависим от атрибутите, съставляващи първичния ключ, но не и от негово подмножество. В частност първичния ключ на тази релация се състои от един елемент и единственото възможно подмножество е празното, което не определя никой атрибут. За 3НФ е необходима 2НФ и ако $A_1, A_2, \dots, A_n \rightarrow B$ е нетривиална ФЗ която е в сила за R, то или $\{A_1, A_2, \dots, A_n\}$ да е супер-ключ за R или B да е част от ключ. В случая и двете условия са изпълнени, т.е. нашата релация е в 3НФ.

2. Reader:

- $ReaderID \rightarrow RegistrationDate, ExpirationDate$
- $ReaderID, RegistrationDate \rightarrow ExpirationDate$

Аргументите, които важат за предишната релация Reader, са валидни и за тази.

3. Fines:

- $FineNo \rightarrow Value, PaymentDate, ReaderID$
- $FineNo, ReaderID \rightarrow Value, PaymentDate$
- $FineNo, ReaderID, Value \rightarrow PaymentDate$

И тук, както в предните два случая можем да приложим същите аргументи и да заключим, че релацията е в 3НФ.

4. Book:

- $SystemNo \rightarrow ISBN, Title, Pages, Edition, Type, PublisherID, PublishDate, Taken$
- $ISBN \rightarrow Title, Pages, Edition, Type, PublisherID, PublishDate$

Въпреки, че релацията е в 2НФ, втората функционална зависимост нарушава правилото за 3НФ и затова се налага да нормализираме. В резултат на нормализацията получаваме следните две релации, които са в 3НФ:

Inventory(SystemNo, ISBN, Taken), Book(ISBN, Title, Pages, Edition, Type, PublisherID, PublishDate)

5. Publisher:

- $PublisherID \rightarrow Name, Address$
- $PublisherID, Name \rightarrow Address$
- $Name, Address \rightarrow PublisherID$

Отново, както и в релацията Person можем да видим, че релацията е в 3НФ, като разликата тук е, че в последната функционална зависимост дясната страна е част от ключ.

6. Department: Следваме същата логика както за предишната релация - няма нужда да правим нормализация.
7. Registered, WrittenBy: Тъй като и двете релации съдържат само по два атрибута те са в НФБК. Единствената промяна, която ще направим е да подменим атрибута SystemNo с ISBN за да избегнем повторения (може да има много книги с един и същ ISBN, но различни SystemNo).
8. Borrows:

- $ReaderID, SystemNo \rightarrow BorrowDate, DueDate, ReturnDate$
- $ReaderID, SystemNo, BorrowDate \rightarrow DueDate, ReturnDate$
- $ReaderID, SystemNo, ReturnDate \rightarrow BorrowDate, DueDate$

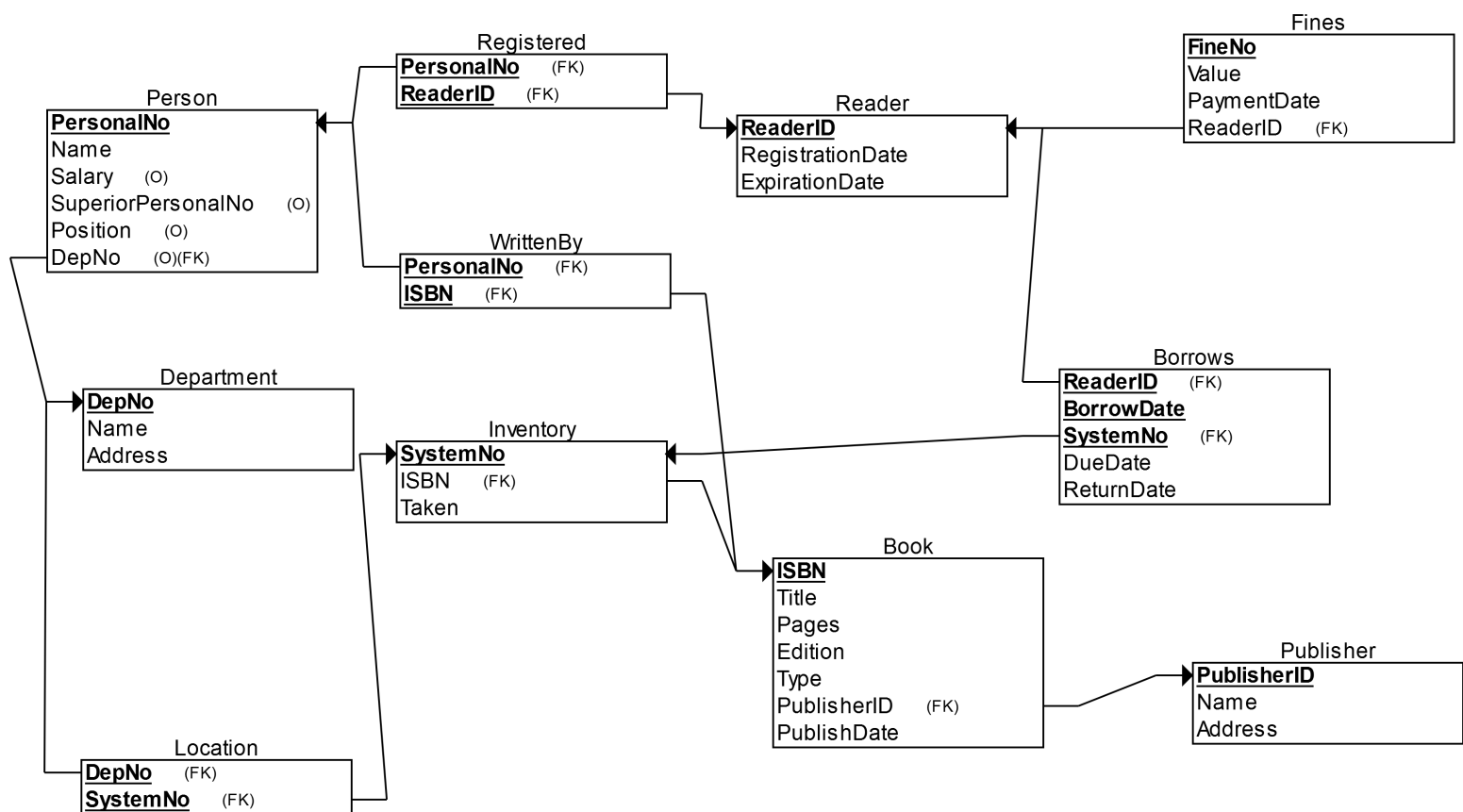
В случая всеки атрибут зависи от ReaderID и SystemNo, но поотделно нито читател, нито книга определят заемане, т.е. имаме 2НФ. За всяка от посочените ФЗ, лявата част е суперключ, следователно имаме 3НФ и не се налага да нормализираме.

9. Location:

- $SystemNo \rightarrow DepNo$

Тук ключът е {DepNo, SystemNo} и очевидно релацията е в 2НФ, а понеже и функционалната зависимост удовлетворява условието дясната страна да е част от ключ, то релацията е в 3НФ.

На фигура 2 е илюстриран релационен модел, описващ базата данни.



Фигура 2: Релационен модел на базата данни

4 Функции

Дефинирани са четири функции: `most_read`, `register_person`, `dep_contains_isbn`, `borrow_book`.

Функцията `most_read` не приема никакви параметри и връща ISBN-а на книгата, която е била заемана най - често от всички книги (независимо от кой клон на библиотеката).

Функцията `register_person` приема два параметъра - низ от 10 символа съдържащ `personalNo` и цяло число, което означава идентификатора на новия читател. Функцията регистрира нов читател с посочените параметри и връща `true` ако действието е било успешно. Ако вече съществува читател с подадения идентификатор или не съществува човек, чиито номер да отговаря на подадения, то функцията хвърля изключение.

`Dep_contains_isbn` е функция, която приема два параметъра и връща стойност от булев тип. Двата параметъра означават съответно ISBN на книга и номер на клон на библиотеката, като съответно типът на първата променлива е `char(13)`, а на втората `int`. При некоректни параметри (т.е. не съществуващ клон или книга с посочения ISBN), функцията връща информация за настъпилото изключение, а в противен случай връща стойност, която зависи от това дали съответната книга може да бъде намерена в искания клон.

Последната дефинирана функция `borrow_book` приема три параметъра - идентификатор на читател, ISBN на книга и номер на клон, а връща променлива от тип `varchar(255)`, която означава системен номер на книга. Целта на функцията е да заеме първата свободна книга, чиито ISBN и клон отговарят на изисканите и да върне нейният системен номер. Отново при некоректни параметри функцията дава на потребителя информация за настъпилото изключение.

5 Тригери

Във файлът `createTrig.sql` са дефинирани следните два тригера : `trigger _ins_borrow` и `trigger _publisher_book`.

Първият тригер - `_ins_borrow` се изпълнява след добавяне на нов запис в таблицата `Borrows`. Той се грижи за отбелязването на книгата, отговаряща на новодобавения запис, като взета в таблицата `Inventory`.

Тригерът `trigger _publisher_book` се активира след добавяне на запис в таблицата `Book`. При своето изпълнение, ако не съществува издателство с необходимия идентификатор в таблицата `Publisher`, той добавя ново издателство, без формално име и адрес, за да запази валидността на външния ключ.

6 Изгледи

Трите изгледа: `not_returned`, `authorsbooks` и `publisher_books` са дефинирани във файла `createView.sql`.

Изгледът `not_returned` връща всички книги, които са взети и все още не върнати, т.е. имат отбелязан статус `taken` в таблицата `Inventory`. Информацията, която се връща е системния номер на книгата, идентификатор на читателя и краен срок за връщане.

`Authorsbooks` е изглед дефиниран така, че да ни връща информация за това, коя книга от кого е написана - връщат се ISBN, име на книгата и автор, където за всяка уникална двойка (произведение, автор) се среща по един запис.

Последният дефиниран изглед - `publisher_books` ни връща информация за това коя книга от кое издателство е - връщат се ISBN, заглавие, име на издателство, което отговаря за издаването на книгата и дата на издаване.

7 Приложение за достъп до базата данни

Приложението за достъп до базата данни е написано на Java и работи по следния начин : При стартиране потребителят трябва да въведе хоста, порта, на който работи базата данни, името на базата данни, `credentials` за достъп до базата и `schema`. След това той има избор от критерии, по които може да търси книга - ISBN, име, автор, издателство или системен номер. След като направи избор, потребителят въвежда низ, по който низ, както и по избрания критерии, той получава отговор на своята заявка. На фиг.3 са показани примерни резултати при изпълнение на заявки, като базата данни е попълнена с информация от скрипта `dataLoad.sql`.

Забележка: Скриптовете използвани при реализацията на проекта, както и примерното приложение са написани и тествани за PostgreSQL11 сървър. В зависимост от това дали се използва различна версия или различен сървър, може да възникнат проблеми при използването на скрипта и приложението.

```
-----
Choose filter to run find book:
1)ISBN
2)Title
3)Author
4)Publisher
5)SystemNo
6)Exit
2
Input argument:
Complete

Result:
| isbn| title|pages|edition| backtype|publisherid|publishdate|
|9780131873254|Database Systems: The Complete Book|| 1248|| 2||Paperback|| 1|| 2008-06-15|
-----
```

```
Choose filter to run find book:
1)ISBN
2)Title
3)Author
4)Publisher
5)SystemNo
6)Exit
4
Input argument:
Pearson

Result:
| isbn| title|pages|edition| backtype|publisherid|publishdate| | | | | | |
|9780131873254|| Database Systems: The Complete Book|| 1248|| 2||Paperback|| 1|| 2008-06-15|
|9780132943260|Database Systems: A Practical Approach to Design, Implementation, and Management|| 1440|| 6||Paperback|| 1|| 2014-01-18|
```

Фигура 3: Примерни заявки, изпълнени от приложението