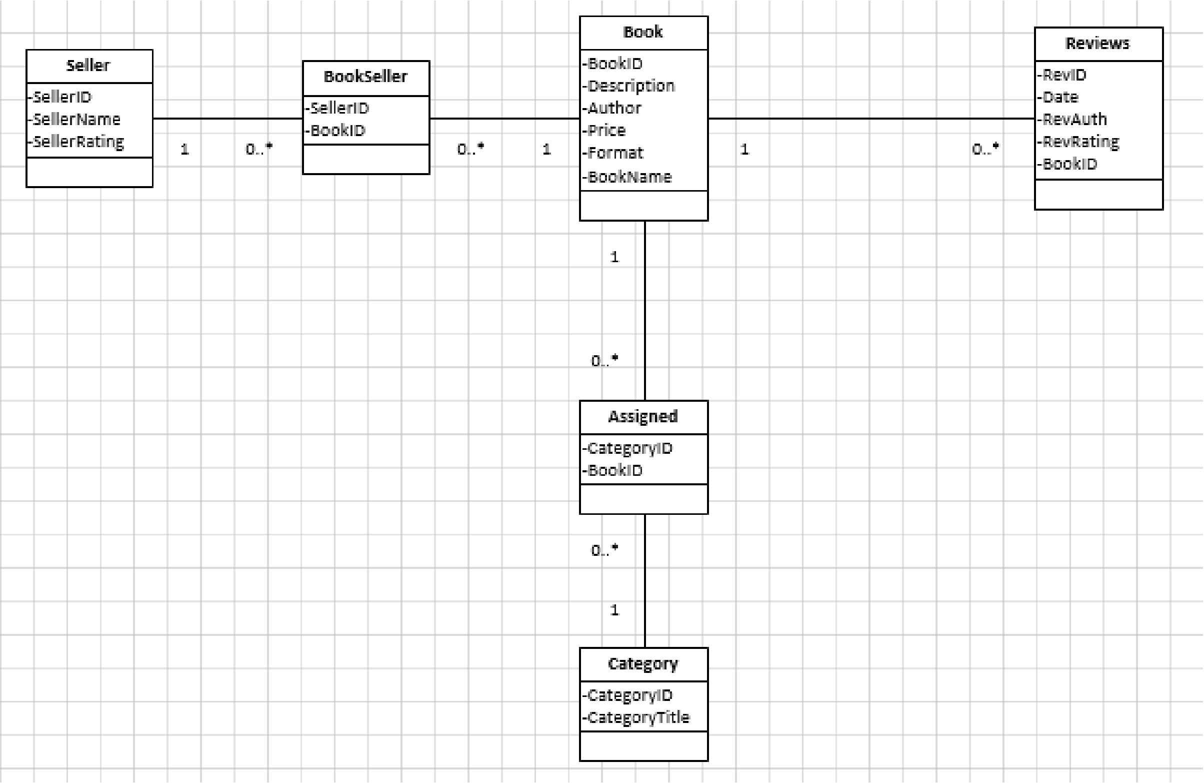
Scenario Modelling and Database Implementation

CE205 Assignment 1 2015-16

Evaldas Senavaitis (1402039)

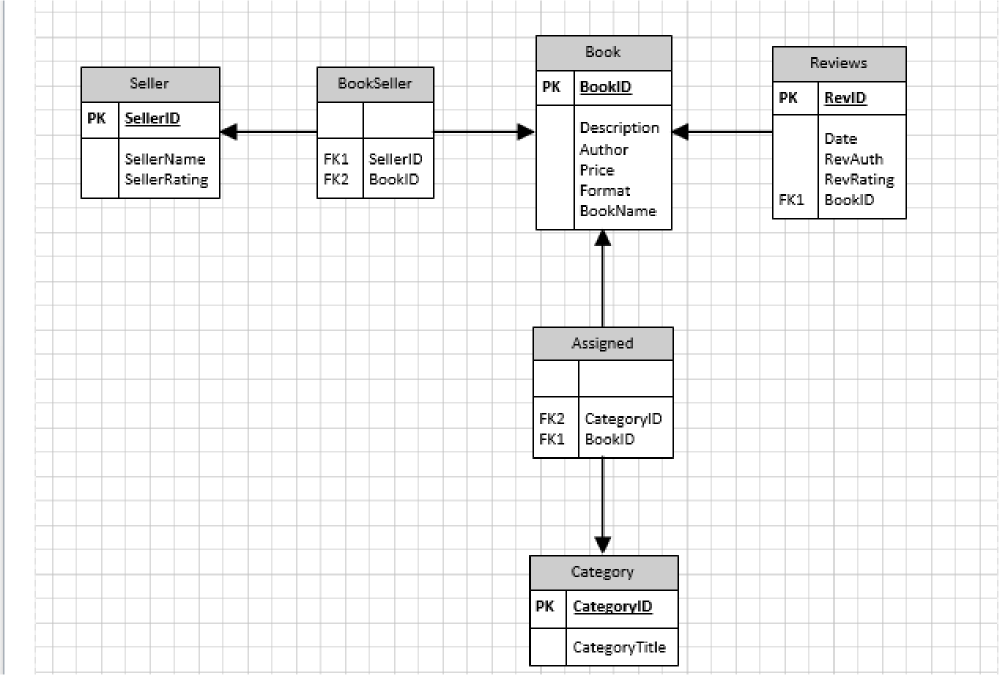
1. Choice of Entities, Attributes and Relationships

I determined four entities as you asked and I think they are Seller, Category, Reviews, BookSeller, Assigned and Book as required. I chose these, because I believe they are most obvious and required to suit 3NF form. To determine the attributes you don't need much knowledge, as they are really simple, basic attributes of those entities. I came up with relationships easily because I imagined how the real website works or any other business would. Relatioships works simple in this design as it is simple database. Book depends on seller, review and category, as shown in the design pretty much everything points to book as it the main focus in the database. To see in detail my entities, attributes and relationships look at diagram below.



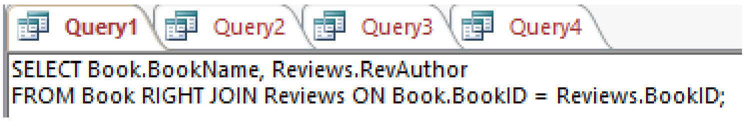
2. Conversion of Model to 3NF

I made my design to 3NF from the start as database is not complicated, of course it could be scaled in the future no problem. I knew that seller can have many books and book can have many sellers, same follows to category, as book can have many categories and categorie can have many books, but review can only point to one book, but book can have many reviews. Knowing this I can make two new entities bookseller and assigned for many to many relationships to work and to pass as 3NF standart diagram . All this is shown in my 3NF diagram below.



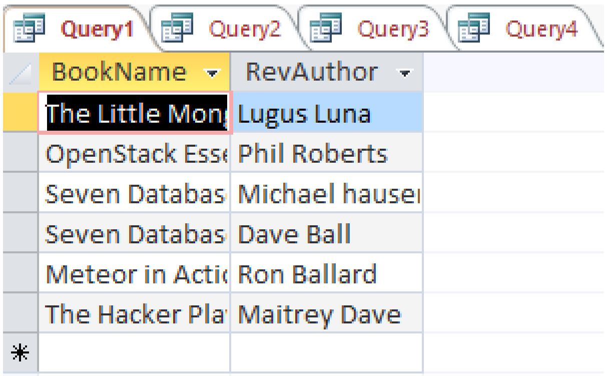
3. Sample Queries and Output

Querie 1:



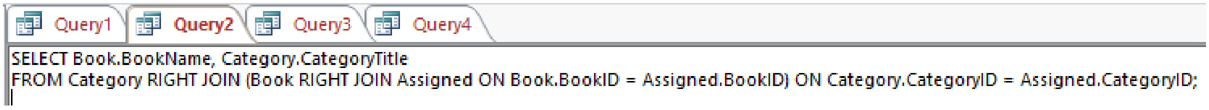
```
SELECT Book.BookName, Reviews.RevAuthor
FROM Book RIGHT JOIN Reviews ON Book.BookID = Reviews.BookID;
```

Querie 1 output:



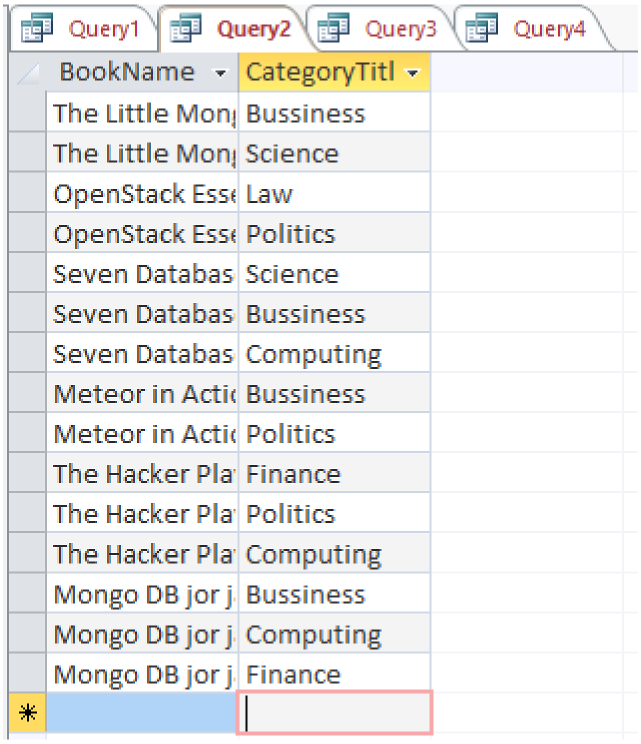
BookName	RevAuthor
The Little Mon	Lugus Luna
OpenStack Esse	Phil Roberts
Seven Databas	Michael hauser
Seven Databas	Dave Ball
Meteor in Actio	Ron Ballard
The Hacker Pla	Maitrey Dave

Querie 2:



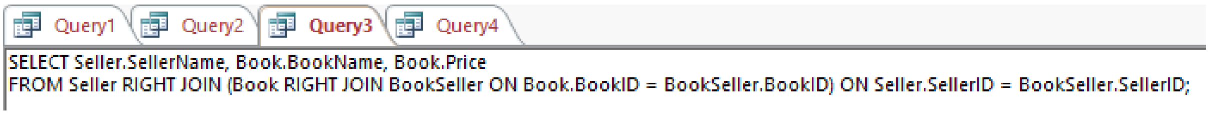
```
SELECT Book.BookName, Category.CategoryTitle
FROM Category RIGHT JOIN (Book RIGHT JOIN Assigned ON Book.BookID = Assigned.BookID) ON Category.CategoryID = Assigned.CategoryID;
```

Querie 2 output:



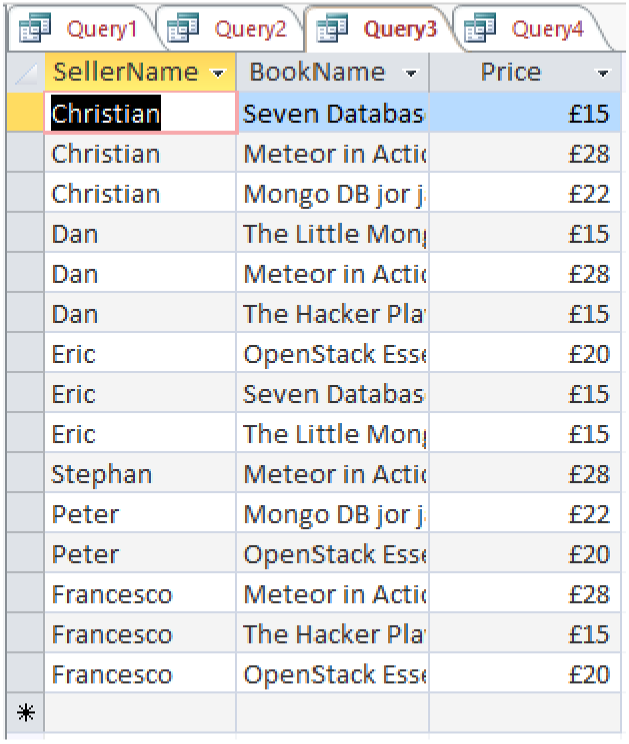
BookName	CategoryTitl
The Little Mon	Bussiness
The Little Mon	Science
OpenStack Esse	Law
OpenStack Esse	Politics
Seven Databas	Science
Seven Databas	Bussiness
Seven Databas	Computing
Meteor in Acti	Bussiness
Meteor in Acti	Politics
The Hacker Pla	Finance
The Hacker Pla	Politics
The Hacker Pla	Computing
Mongo DB jor j	Bussiness
Mongo DB jor j	Computing
Mongo DB jor j	Finance

Querie 3:



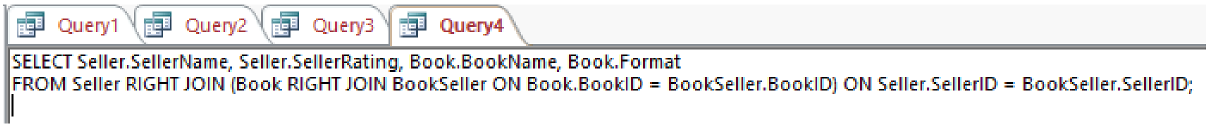
```
SELECT Seller.SellerName, Book.BookName, Book.Price
FROM Seller RIGHT JOIN (Book RIGHT JOIN BookSeller ON Book.BookID = BookSeller.BookID) ON Seller.SellerID = BookSeller.SellerID;
```

Querie 3 output:



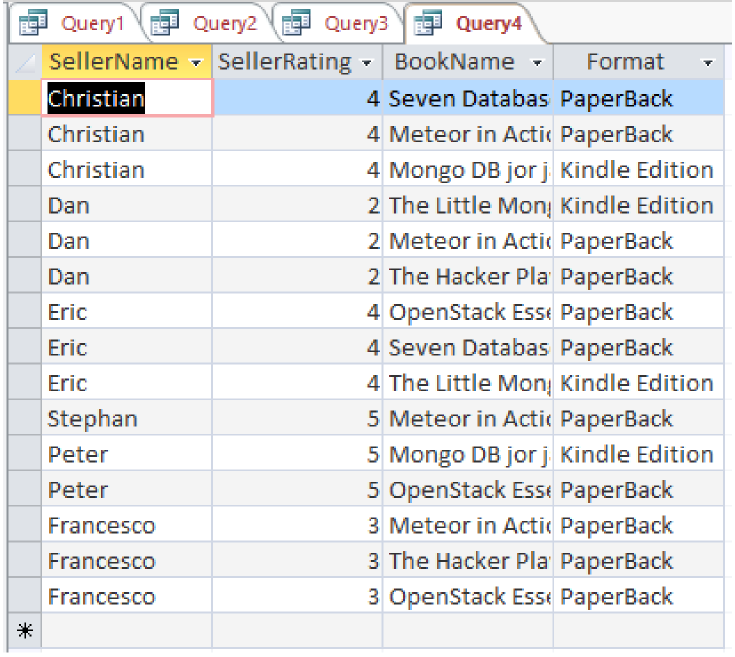
SellerName	BookName	Price
Christian	Seven Databas	£15
Christian	Meteor in Actio	£28
Christian	Mongo DB jor j	£22
Dan	The Little Mon	£15
Dan	Meteor in Actio	£28
Dan	The Hacker Pla	£15
Eric	OpenStack Esse	£20
Eric	Seven Databas	£15
Eric	The Little Mon	£15
Stephan	Meteor in Actio	£28
Peter	Mongo DB jor j	£22
Peter	OpenStack Esse	£20
Francesco	Meteor in Actio	£28
Francesco	The Hacker Pla	£15
Francesco	OpenStack Esse	£20

Querie 4:



```
SELECT Seller.SellerName, Seller.SellerRating, Book.BookName, Book.Format
FROM Seller RIGHT JOIN (Book RIGHT JOIN BookSeller ON Book.BookID = BookSeller.BookID) ON Seller.SellerID = BookSeller.SellerID;
```

Querie 4 output:



SellerName	SellerRating	BookName	Format
Christian	4	Seven Databas	PaperBack
Christian	4	Meteor in Actio	PaperBack
Christian	4	Mongo DB jor j	Kindle Edition
Dan	2	The Little Mon	Kindle Edition
Dan	2	Meteor in Actio	PaperBack
Dan	2	The Hacker Pla	PaperBack
Eric	4	OpenStack Esse	PaperBack
Eric	4	Seven Databas	PaperBack
Eric	4	The Little Mon	Kindle Edition
Stephan	5	Meteor in Actio	PaperBack
Peter	5	Mongo DB jor j	Kindle Edition
Peter	5	OpenStack Esse	PaperBack
Francesco	3	Meteor in Actio	PaperBack
Francesco	3	The Hacker Pla	PaperBack
Francesco	3	OpenStack Esse	PaperBack

I chose querie 1 because it shows that IDs and keys work on the both tables and relations are good as well, it shows book and review that each book has.
Querie 2 shows that book can have multiple categories and categories can have multiple books in them, and they are all related by IDs.
Querie 3 shows that sellers can have multiple books and book can have multiple sellers.
Querie 4 shows similiarities to querie 3, but it displays seller rating and book formats.

I believe my queries shows that all primary and foreign keys work in my design, this concludes that with some advanced queries you could get any output you would like. Overall assignment is not clear at all, because what kind of queries you would like to see for it to prove that is working, so I am providing these four simple queries.