

# Database Fundamentals

## Lab 4 Exercise 1 Suggested Solution

### Objective:

1. recap on ERDs
2. Relational modelling
3. Bring relations to 3<sup>rd</sup> normal form

### For each of the following descriptions:

1. create an ERD, either on paper or using Workbench, which includes attributes and primary keys
2. Convert the ERD to a set of well structured relations.

The exercises will be corrected in the lab. There is no need to upload them to moodle

### Exercise 1:

A library needs a database to record details about its members, its books and what books are out on loan. Members can borrow a number of books, and a book can be borrowed by many members.

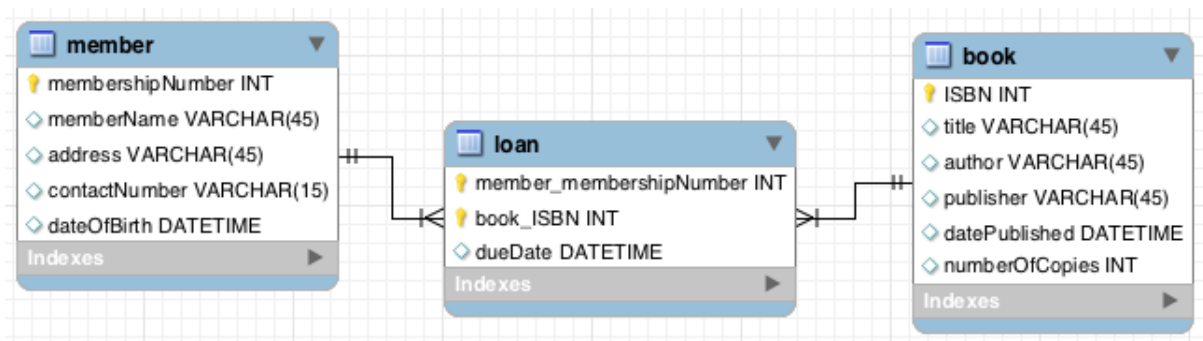
For members, the library needs to record their membership number, name, address, contact number and their date of birth.

For each book, the library records the ISBN number, title, author, publisher, print date and number of copies.

When a customer borrows a book, the library records who has the book, and when it's due back.

### Solution:

#### ERD:



Note: There is a m:n relationship between member and book. When you add this using MySQL workbench, it automatically adds a link entity. This link entity has a composite primary key (made up of two attributes), ISBN and membershipNumber. It is therefore used to store all information about a member borrowing a book, and so holds the due date.

# Database Fundamentals

## Lab 4 Exercise 1 Suggested Solution

Relational model:

Member (membershipNumber(PK), memberName, addressline1, addressline2, addressline3, contactNumber, dateOfBirth)

Loan(membershipNumber(PK, FK), ISBN (PK, FK), dueDate)

Book (ISBN(PK), title, author, publisher, datePublished, numberOfCopies)

Note: address is a composite attribute, and so is split into three attributes in the member table.

The solution assumes a book can only have one author. If a book can have more than one author, then author is a multivalued attribute, which is handled the same way as a repeating group, i.e. it must be in its own table as follows:

Book-Author(ISBN(PK, FK), authorID(PK), author)

Are these relationals well-structured, i.e. in 3<sup>rd</sup> normal form?

Member and Loan are fine.

In the book table, **publisher** is a transitive dependency, this attribute should be in a publisher table, leaving just a **foreign key** in the book table to link the two tables as follows:

Book (ISBN(PK), title, author, publisherID(FK), datePublished, numberOfCopies)

Publisher(publisherID(PK), publishername)

In the Book-Author table, author is a partial dependency, as its not dependent on ISBN - it should be in an 'author' table as follows with the PK in the new author table being the part of the composite key that author is dependent on.:

Book-Author(ISBN(PK, FK), authorID(PK,FK))

Author (authorID(PK), author)

Final tables populated with sample data will look like the following:

## Database Fundamentals

### Lab 4 Exercise 1 Suggested Solution

#### Book

ISBN	title	publisherID	date published	numberOfCopies
1123445543	Database	P001	2007	10
45687645	Java	P001	2008	15

#### Book-Author

ISBN	author
1123445543	A001
1123445543	A002
45687645	A003

#### Publishers

publisherID	Publishername
P001	O'Reilly
P002	Pearson

#### Authors

AuthorID	AuthorName
A001	Richie
A002	Lafore
A003	Bloch

#### Member

membershipNumber	memberName	addressLine1	addressLine2	addressLine3	contactNumber
B00002222	Murphy	1 main stree	blanchardstown	dublin 15	085123456
B3456434534	Ryan	2 Main Street	blanchardstown	dublin 15	085654432

## Database Fundamentals

### Lab 4 Exercise 1 Suggested Solution

#### Loan

membershipNumber	ISBN	dueDate
B00002222	1123445543	01/06/2010
B00002222	45687645	07/06/2010
B3456434534	1123445543	24/05/2010