**Normalisation Quiz**

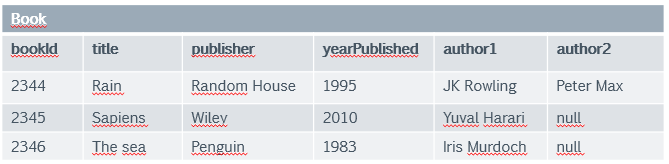
**Question 1**

Which of the following statements is NOT true about normalisation?

* It ensures that every attribute is in the appropriate entity by testing whether every attribute is uniquely defined by the primary key.

**Question 2**

We are returning to our online book store. We have noticed that not every book has a single author. We have come up with this design:

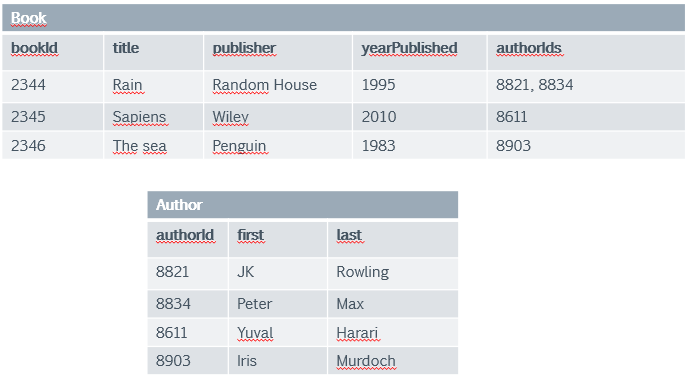


We are debating whether to add three more attributes, author3, author4 and author5. In this way, we are sure we can cover any book, because no book has had more than five authors yet. What do you think about this solution?

* It violates 1st normal form, because it has repeating groups.

**Question 3**

We have worked a bit more on the design, and we concluded that the author needs an entity of its own, because we will have more information to record about the author. So we start with a simple relationship that uses the authorId as a foreign key.

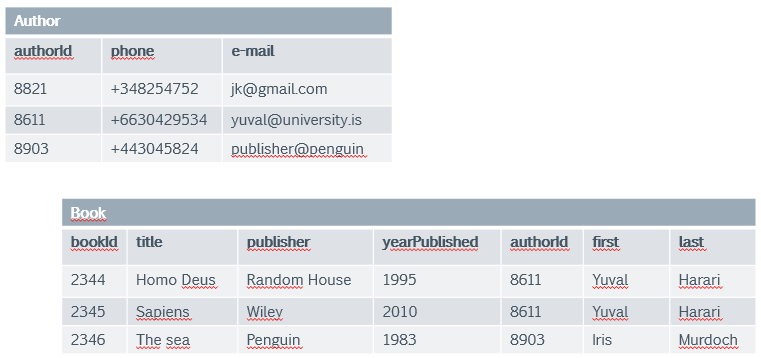


What are your thoughts about this design?

* The Book relation is not in 1NF because the authorIds are still a repeating group. The example also demonstrates why many-to-many is not possible.

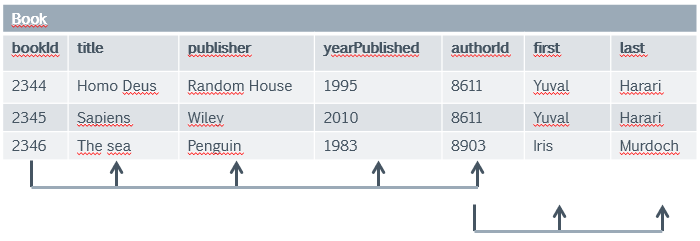
**Question 4**

We have returned to the assumption that a book has only one author. We have also added some author-specific information. We now have the following entities:



We will now examine the functional dependencies of the Book relation. This requires us to look at possible candidate primary keys and examine whether the non-key attributes are fully defined by the candidate primary key. We find the following:

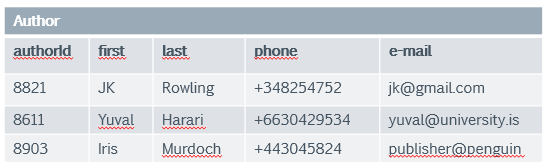
* The Book table is in 2NF, but not in 3NF:



The most likely candidate key is bookId. It defines title, publisher, yearPublished and authorId. But the first and last names of the authors are defined by authorId, not bookId, therefore we have a transitive dependency.

**Question 5**

We have arrived at the conclusion that the author table of the book store database should look like:



We now want to ascertain that it is in 1NF. How do we go about it?

* 1. We observe that there are no repeating groups, that the attributes are all atomic. The table is in 1NF.

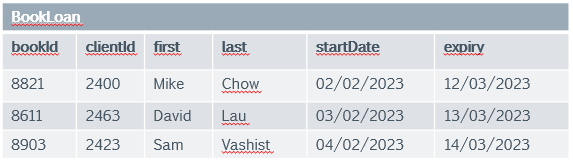
2. a) We consider all attributes for primary key candidates. We observe that only the authorId attribute is a good candidate. As it is a unique ID field, we do not have to consider composite keys.

2. b) We conclude that all non-key attributes are fully functionally dependent on the authorId key. The table is in 2NF.

3. We examine the table for possible transitive dependencies, trying to find a non-key attribute that defines another attribute better than the primary key. We don't find any and conclude that our table is in 3NF.

**Question 6**

We are now focussing on the BookLoan table we have visited before. The bookId and clientId attributes are foreign keys to the Book and Client tables respectively, first and last are the client's first and last name, startDate is when the loan is taken out and expiry is when the loan expires. We are assuming that there are several copies of a book that is identified by the bookId (there may be copies 1, 2 and 3 of book 8821).

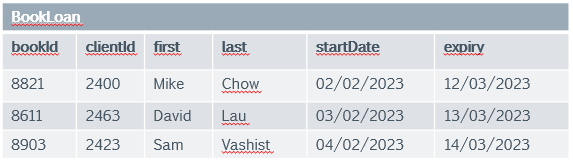


We have to test whether this table is in 3NF. Which of the answers is true:

* The BookLoan table is in 1NF but not in 2NF, because the first and last name attributes are only partially dependent on the composite key of bookId + clientId + startDate.

**Question 7**

We are still reviewing the BookLoan table. The bookId and clientId attributes are foreign keys to the Book and Client tables respectively, first and last are the client's first and last name, startDate is when the loan is taken out and expiry is when the loan expires. We have revised our assumption and now assign a bookId to every copy of a book. For example, we have 3 copies of War and Peace (by Leo Tolstoy). Their bookIds are 8831, 8832 and 8833 respectively. The table structure otherwise has not changed:

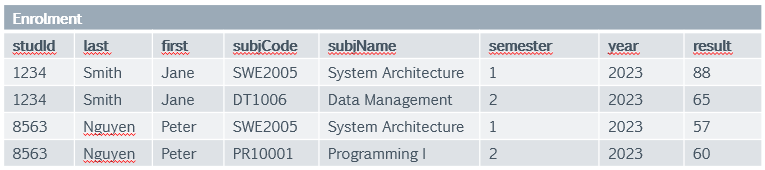


We have to test whether this table is in 3NF under the new assumptions. Which of the answers is true:

* The BookLoan table is in 2NF but not in 3NF, because there is a transitive dependency between the first and last name attributes and the composite key of bookId + startDate, because the composite key defines clientId and clientId defines first and last.

**Question 8**

A university has a similar flat file structure of their enrolments. As we can see, there is a lot of redundancy in their records - such as student names and subject names repeating. It is time for you to assist and help them.



The university has been meticulous in their approach and therefore, there are no repeating groups and the table is already in 1NF. Bring it into 2NF by developing a primary key that can define every non-key attribute. Then remove attributes which are only partially dependent on this key.

* The smallest composite key that can define a result multiple times (when a student repeats a subject) is studId, subjCode, semester and year. last, first and subjName are all partially dependent on this key, therefore we remove all three into their own tables:

