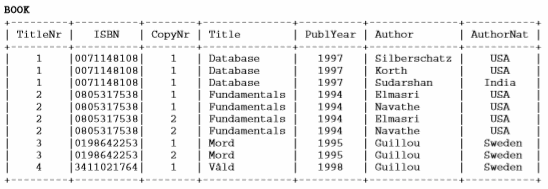
<https://beginnersbook.com/2015/05/normalization-in-dbms/>

**Normalization** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.



**Functional dependencies in the table above**:  
  
TitleNr  
ISBN  
CopyNr -> ISBN  
Title -> {ISBN, Author}  
PublYear -> {ISBN, Author}  
Author -> {ISBN, Title, PublYear}  
AuthorNat -> Author

**Update anomaly**: We have multiple rows for the same ISBN. If we have to update the number of copies, we have to update multiple rows. If somehow, the correct number of copies get updated against one author but not in other authors then as per the database, the same ISBM would be having two different CopyNr, which is not correct and would lead to inconsistent data.

**Insert anomaly**: Suppose a new book is ordered, and author’s Nationality is unknown then we would not be able to insert the data into the table if AuthorNat field doesn’t allow nulls.  
  
**Delete anomaly**: Suppose, if a publisher stopped publish the book anymore and library has no longer has the book due to lost or damaged, deleting rows will delete the book historical information as well as author information as well. We can avoid this by having additional flag to indicate whether the book is in circulation or not.

To overcome these anomalies we need to normalize the data.  
  
**First normal form (1NF)**: As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.  
  
The columns in BOOK table has only one value and it is atomic. It is in compliance of 1NF.

**Second normal form (2NF)**: A table is said to be in 2NF if both the following conditions hold:  
  
- Table is in 1NF (First normal form)  
- No non-prime attribute is dependent on the proper subset of any candidate key of table.  
  
An attribute that is not part of any candidate key is known as non-prime attribute.

Candidate/Primary keys: TitleNr, CopyNr and Author  
Alternative keys: ISBN, CopyNr and Author  
Non prime attribute: AuthorNat

Non prime attribute “AuthorNat” dependent on “Author” which is subset of candidate key. This violates the rule for 2NF as the rule says “**no** non-prime attribute is dependent on the proper subset of any candidate key of the table”. A separate table “AuthorNationality” with Author & Nationality column will make it 2NF compliant.

**Third Normal form (3NF)**: A table design is said to be in 3NF if both the following conditions hold:  
- Table must be in 2NF  
- [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.  
  
An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:  
  
- X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table  
- Y is a prime attribute of table  
  
An attribute that is a part of one of the candidate keys is known as prime attribute.

Super keys: {ISBN}, {ISBN,Title},{ISBN,Title,PublYear},{ISBN,Title,PublYear,Author} … so on  
Candidate keys: ISBN  
Non prime attributes: Except ISBN, all other attributes are non prime attributes as they are not part of candidate keys.

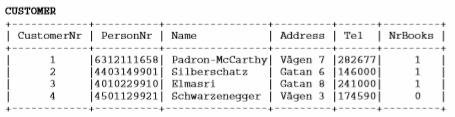
Title, PublYear,AuthorNat dependent on Author, and author is dependent on ISBN that makes non-prime attribute Title, PublYear, AuthorNat transitively dependent on ISBN. This violates the rule of 3NF.

**Boyce Codd normal form (BCNF)**: It is an advance version of 3NF that’s why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every [functional dependency](https://beginnersbook.com/2015/04/functional-dependency-in-dbms/) X->Y, X should be the super key of the table.

**Functional dependencies in the table above**:  
TitleNr  
ISBN  
CopyNr -> ISBN  
Title -> {ISBN, Author}  
PublYear -> {ISBN, Author}  
Author -> {ISBN, Title, PublYear}  
AuthorNat -> Author

**Candidate key**: {ISBN}

The table is not in BCNF as neither ISBN nor Author alone are keys.



**Functional dependencies in the table above**:  
  
CustomerNr ->  
PersonBr ->  
Name ->  
Address ->  
Tel ->  
NrBooks

**Update anomaly**:

**Insert anomaly**:   
  
**Delete anomaly**:

To overcome these anomalies we need to normalize the data.  
  
**First normal form (1NF)**: As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.  
  
**Second normal form (2NF)**: A table is said to be in 2NF if both the following conditions hold:  
  
- Table is in 1NF (First normal form)  
- No non-prime attribute is dependent on the proper subset of any candidate key of table.  
  
An attribute that is not part of any candidate key is known as non-prime attribute.

Candidate/Primary keys: CustomerNr  
Alternative keys: PersonNr  
Non prime attribute:

**Third Normal form (3NF)**: A table design is said to be in 3NF if both the following conditions hold:  
- Table must be in 2NF  
- [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.  
  
An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:  
  
- X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table  
- Y is a prime attribute of table  
  
An attribute that is a part of one of the candidate keys is known as prime attribute.

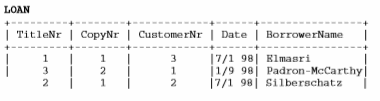
Super keys:   
Candidate keys:   
Non prime attributes:

**Boyce Codd normal form (BCNF)**: It is an advance version of 3NF that’s why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every [functional dependency](https://beginnersbook.com/2015/04/functional-dependency-in-dbms/) X->Y, X should be the super key of the table.

**Functional dependencies in the table above**:

**Candidate key**:

The table is not in BCNF as neither ISBN nor Author alone are keys.



**Functional dependencies in the table above**:  
  
CustomerNr ->  
PersonBr ->  
Name ->  
Address ->  
Tel ->  
NrBooks

**Update anomaly**:

**Insert anomaly**:   
  
**Delete anomaly**:

To overcome these anomalies we need to normalize the data.  
  
**First normal form (1NF)**: As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.  
  
**Second normal form (2NF)**: A table is said to be in 2NF if both the following conditions hold:  
  
- Table is in 1NF (First normal form)  
- No non-prime attribute is dependent on the proper subset of any candidate key of table.  
  
An attribute that is not part of any candidate key is known as non-prime attribute.

Candidate/Primary keys: TitleNr and CopyNr  
Alternative keys:   
Non prime attribute:

**Third Normal form (3NF)**: A table design is said to be in 3NF if both the following conditions hold:  
- Table must be in 2NF  
- [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.  
  
An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:  
  
- X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table  
- Y is a prime attribute of table  
  
An attribute that is a part of one of the candidate keys is known as prime attribute.

Super keys:   
Candidate keys:   
Non prime attributes:

**Boyce Codd normal form (BCNF)**: It is an advance version of 3NF that’s why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every [functional dependency](https://beginnersbook.com/2015/04/functional-dependency-in-dbms/) X->Y, X should be the super key of the table.

**Functional dependencies in the table above**:

**Candidate key**:

The table is not in BCNF as neither ISBN nor Author alone are keys.

ER Diagram:

Exercise

2.1

Table : BOOK

Functional Dependencies:

TitleNr → ISBN

ISBN → Title

Author → AuthorNat

Table is in 1NF, there is a partial dependency of attribute AuthorNat on part of the primary key attribute Author . So the table is not in 2NF form

Table : CUSTOMER

CustomerNr → PersonNr

PersonNr → Name, Address, Tel, NrBooks

Table Customer is in 2NF

CustomerNr → PersonNr and PersonNr → Name, Address, Tel, NrBooks

So there is a transitive dependency between CustomerNr → Name, Address, Tel, NrBooks. So the table is not in 3NF

Table : LOAN

TiltleNr, CopyNr → CustomerNr

CustomerNr → BorrowerName

Table LOAN is in 2NF,

TiltleNr, CopyNr → CustomerNr and CutomerNr → BorrowerName so ther is a transitive dependency between TiltleNr, CopyNr → BorrowerName

So the table is not in 3NF

2.2

All three tables Book, Customer, Loan does not fulfill BCNF requirements

Book

Following are the problems/issues due to the table not being in BCNF normalization

Update Anomaly

One Author might have written multiple books, so there will be multiple rows for the same author. If someone updates the Author nationality in one row and does not update in other rows for the same author. The database will be inconsistent.

Delete Anomaly

Since multiple records are stored for the same title/author due to different copy numbers. While deleting a title/author if they don’t delete for all the copy number, the database will be inconsistent

Insert Anomaly

For each new ISBN number being added to the system , if there are multiple copies of that book , multiple rows of the same information has to be inserted into the table.

For example if ISBN 1111, has 10 copies then 10 rows will be inserted. Lot of redundant information

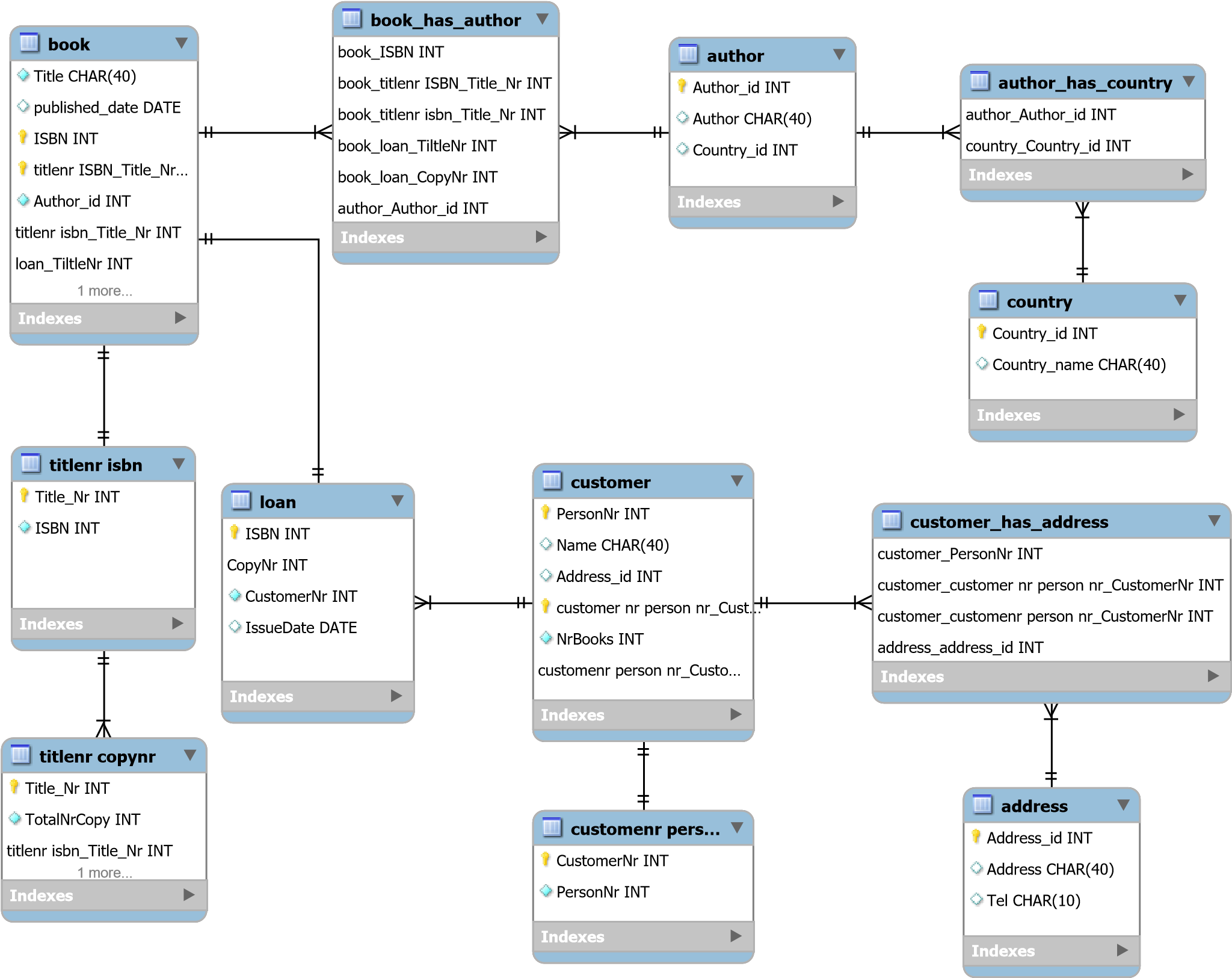
Loan, Customer

Update Anomaly

If a customer name changes both Loan and Customer table has to be updated. If accidentally one table is updated Not the other then the database will become inconsistent.

2.3

Entity Diagram



Tables for new library system

Titlenr isbn( TiltleNr, ISBN)

FD TitleNr → ISBN

Tiltlecopynr(TitleNr,TotalNrCopy,TotalNrCopyIssued)

FD TitleNr → CopyNr

Book(ISBN,Title,Author\_id)

FD ISBN → Title, Author

Author(Author\_id,Author,Country\_id)

FD Author\_id → Author, Country\_id

Country(Country\_id,Country\_name)

FD Country\_id → Country\_name

CustomernrPersonnr(CustomerNr,PersonNr)

FD CustomerNr → PersonNr

Customer(PersonNr,Name,Nrbooks,Address\_id)

FD PersonNr → name,Nrbooks,Address\_id

Address(Address\_id,Address,Tel)

FD Address\_id → Address. Tel

Loan(ISBN,CopyNr,CostomerNr,IssueDate)

FD ISBN , CopyNr → CustomerNr, IssueDate