Table of Contents

Introduction:

Normalization is a process which involves restructuring the relational database to obtain a structure with low data redundancy and high data integrity.
The procedure to arrive at the fully normalized form include the following:
- UNF (un-normalized form)
- 1NF (First normalized form)
- 2NF (Second normalized form)
- 3NF (Third normalized form)

Given below is the sample output of Monash Library Services catalogue search:

APPENDIX A

Sample output from the Monash Library Service Catalog Search

Search "Coronel Morris" Selected Output:

Call No:	005.74 C822D 2018
Title:	Database systems : design, implementation, and management
Author:	Carlos Coronel. Steven Morris (Steven A.).
Subjects:	Database design; Database management; Databases
Description:	"Provide a balanced approach to databases as the market-leading DATABASE SYSTEMS: DESIGN, IMPLEMENTATION, and MANAGEMENT, 13E emphasizes the knowledge and skills necessary for success and makes databases accessible without overwhelming students. Readers gain a solid foundation in database design and implementation as diagrams, illustrations, and tables clarify in-depth coverage of database design. Students learn how successful database implementation involves designing databases to fit within a larger strategic data environment. Revised coverage of SQL introduces more examples and simpler explanations that focus on areas most important for a database career -- making supplementary SQL materials unnecessary. Additional coverage of Big Data Analytics and NoSQL, including related Hadoop technologies, offers a strong hands-on approach. MindTap® now include a digital auto-graded MySQL environment." -- Publisher.
Publisher:	Boston, MA, USA : Cengage Learning
Date Published:	2018
Edition	13th Edition
No pages:	802
Language:	English
Notes:	Includes index. Includes bibliographical references and index.
Identifier:	ISBN: 1337627909 Hardcover; ISBN: 9781337627900 Student Edition ; ISBN: 9781337688826 Loose Leaf

Availability for this title:

Central Branch (Branch 100)

	Barcode	Status
	33168034998647	Counter Reserve
	33168034998605	Item in place - Available
	33168034999231	On Loan Until 7/5/2018 21:00 PM

WestEnd Branch (Branch 102)

	Barcode	Status
	331680349913125	Item in place - Available

Figure. Page 5, Assignment 1, 2018, semester1, FIT9132, Introduction to Databases.

UNF (un-normalized form):

This form helps us to represent all the given fields/subjects on our sample output catalogue search as attributes in a table. I have represented a table(Library) and all the attributes in a round bracket as shown below:

TITLE (call_no, title, author_ID, author_name, subjects, description, publisher_ID, publisher_name, date_published, edition, no_pages, language, notes, identifier, branch_code, branch_name, barcode, status)

1NF (First normalized form):

To arrive at the first normalized form we will need to eliminate all the repeating groups in a table and identify primary key in each separated related table.

I use brackets to distinguish between similar related data sets as represented below:

TITLE (call_no, title, (author_ID, author_name), (subjects), description, (publisher_ID, publisher_name, date_published, edition), no_pages, language, notes, (identifier), (branch_code, branch_name, barcode, status))

I have sorted repeating groups from above form and identified primary keys which is depicted with an underlined attribute. For better understanding, I have also shown a dependency diagram for each table.
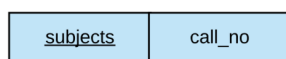
TITLE (call_no, title, description, publisher_ID, publisher_name, date_published, edition, no_pages, language, notes)
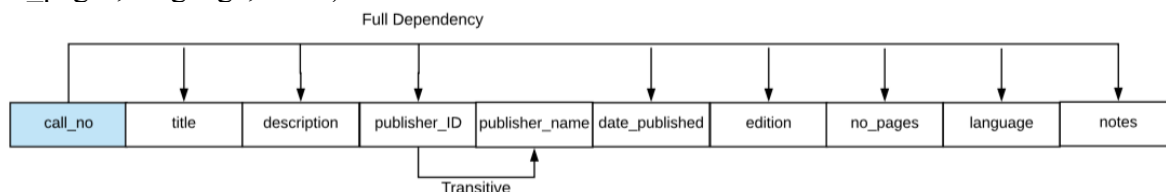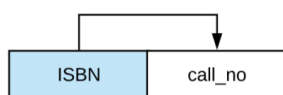

AUTHOR (author_ID, author_name, call_no)
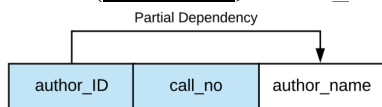
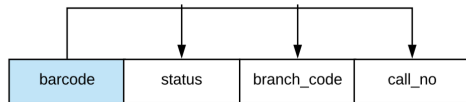
SUBJECT (subjects, call_no)
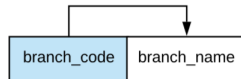

ISBN_identifier (identifier(ISBN), call_no)

BRANCH (branch_code, branch_name, (barcode, status, call_no))
We again find repeating group in the "BRANCH" table and we separate repeating group as below:

BOOK (barcode, status, branch_code, call_no)


BRANCH (branch_code, branch_name)

2NF (Second normalized form):

A relation is said to be in second normalised form if it satisfies all conditions of first normal form and none of the non-key attributes depend on a part of primary key.
This means that all the partial dependencies in first normal form should be separated for a relation to be considered as second normal form.

In our case we see a partial dependency in "AUTHOR" table.
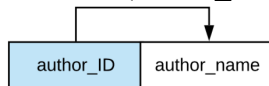**AUTHOR** (author_ID, author_name, call_no)
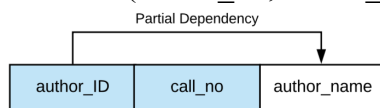
Therefore, this partial dependency is broken down as follows:

TITLE_AUTHOR (author_ID, call_no)


AUTHOR (author_ID, author_name)

3NF (Third normalized form):

Third normalization is performed to ensure data integrity. Once when the relation is in second normalised form, we check for transitive dependency (dependency among non-key attributes).

I have identified that "publisher_ID" is dependent on "call_no" and "publisher_name" is inturn dependent on "publisher_ID" which shows transitive dependency. Therefore, I have created another entity for publisher as shown below.

Please check out final third normalised form depected below:

**TITLE** (call_no, title, description, no_pages, language, notes, date_published, edition, publisher_ID)

Full Dependency

call_no	title	description	date_published	edition	no_pages	language	notes

PUBLISHER (publisher_ID, publisher_name)
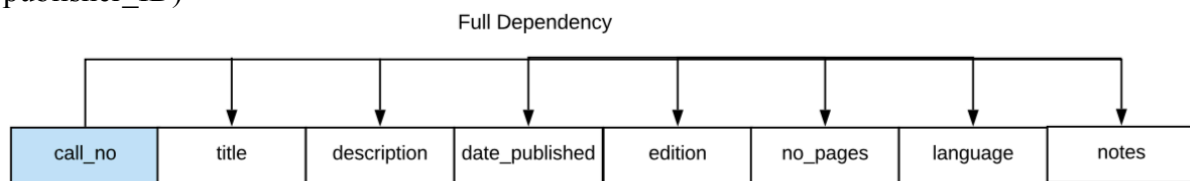
| publisher_ID	publisher_name
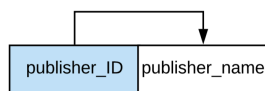
Conclusion:

We have arrived at eight entities to represent the given relational data set after the process of normalization till 3NF. The final set of entities with their underlined/coloured primary keys are as below:

TITLE (call_no, title, description, no_pages, language, notes, date_published, edition, publisher_ID)
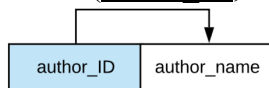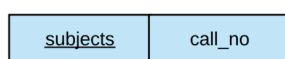
Full Dependency

call_no	title	description	date_published	edition	no_pages	language	notes

PUBLISHER (publisher_ID, publisher_name)
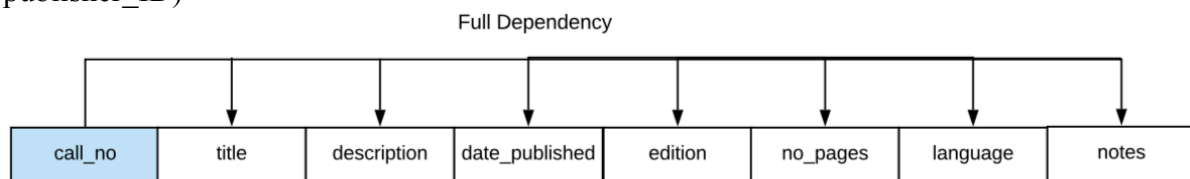
publisher_ID	publisher_name

TITLE_AUTHOR (author_ID, call_no)
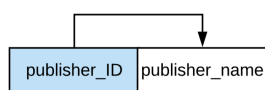
author_ID	call_no

AUTHOR (author_ID, author_name)

author_ID	author_name

SUBJECT (subjects, call_no)

subjects	call_no

ISBN_identifier (<u>identifier(ISBN)</u>, call_no)

ISBN	call_no

BOOK (<u>barcode</u>, status, branch_code, call_no)

barcode	status	branch_code	call_no

BRANCH (<u>branch_code</u>, branch_name)

branch_code	branch_name