



## Assignment 1 - Database Design - Monash Library Services (MonLib)

---

The local Monash Municipality maintains several libraries for its residents across the municipality.

For each branch Monash Library Services assigns a branch code (an incremental number for each branch with the first branch using a code of 100). The branch name, address and contact phone number are also recorded. Each branch is assigned a manager. Due to the small size of some of the branches a particular manager may manage several branches. Each manager is assigned a manager id. Monash Library Services record a manager's name and contact phone number. All managers are assigned to one particular branch as their home branch.

Monash Library Services maintain records of current loans of books to borrowers.

Each borrower is identified by a borrower number and each copy of a title by a barcode number (the library may have more than one copy of any given title). When a borrower first registers to borrow books the branch where they register is recorded as their home branch. The name and address of each borrower is held so that communications, such as overdue loan reminders, can be sent when necessary.

The information held about a title is its Dewey Decimal call no (20 characters) - this call no is used to identify a particular title and for shelving books. The title, description (1000 characters), author's name/s, publisher's name, publication year, international standard book number (ISBN - a given title may have several ISBN's), purchase price, classification (Reference or Fiction), and the number of pages are recorded. For a Fiction title a reading level is also recorded as an integer from 1 (Easy) to 30 (Very Difficult). A given title may be written by a number of different authors, however the library regards a title as only being published by a single publisher. The library assigns its own unique in-house numerical codes to identify authors and publishers.

A title may cover a number of different subjects, which the library wishes to record so that borrowers can use an online catalogue system to select books by subject as well as title and author's name.

There is a restriction on the number of books a borrower may have on loan at any one time and the loan period. These limits depend on the borrower's classification (Junior, Adult, or Organisation).

Some book copies are placed on counter reserve, and are not available for loan - they may only be used in the library. A flag is added to a book to indicate if it is on counter reserve or not. There may also be other copies of the same title which are available for normal loan.

When a book is borrowed (goes out on loan), the return date is automatically recorded based on the current date and the borrower's classification. A record of all loans which take place is maintained. When a book is returned from a loan its actual return date is recorded. A fine notice is generated and is sent to the borrower if the return date is after the loan due date. In the fine notice, the fine amount will be included<sup>1</sup>. When the borrower pays the fine, the payment date will be recorded.

Borrowers may reserve books currently out on loan. The date and time on which the reserve was placed is recorded. A given book may be reserved by several borrowers, the book is made available based on the order in which the reserve was placed by the borrower.

When a borrower returns a book, they may if they wish renew their loan and take the book out for a further loan period provided the borrower has not been flagged to prevent further borrowings and the book has not been reserved by another borrower.

A special borrower's status flag is maintained - borrowers who hold overdue books or who have reached their loan limit or who have an unpaid fine, are flagged to prevent further borrowings.

In designing the database, you need to ensure as a minimum that the following operational requirements of the library are met:

- a. Allow the catalogue to be searched based on authors, subject category, title, etc
- b. Ensure the accuracy of the status of its catalogue eg which items are on loan, which books are available, is there any lost items (overdue for more than 3 months)?
- c. Manage the loan operations on items in its catalogue eg who borrowed an item? Has the item being returned? Is there any fine to be issued due to an overdue loan?
- d. Manage the information about its members, eg what is the address of a member? Is there any fine owed by a member?
- e. Manage the information about the branches and their manager.
- f. Report usage patterns, such as
  - i. what items are popular?
  - ii. what is the average number of items borrowed by a member for a given time period?
  - iii. what subject/topic is popular?
  - iv. which branch has the highest loan activity?

---

<sup>1</sup> Assume the fine is calculated by another system and our library system needs to record only the amount due.

## TASKS

1. Using LucidChart, prepare an **conceptual model** (Entity Relationship Diagram) for Monash Library (MonLib).
  - For this conceptual model, *include what you see as identifiers (keys) for each entity and the other attributes required for that entity to support the operational activities of the database.*
  - Surrogate keys must not be added to this model. *Participation and connectivity for all relationships must be shown on the diagram.*

It is quite acceptable that your stage 2 and 3 tasks result in changes to this conceptual model (in fact we would expect that they will). In this situation, you need to update your initial conceptual model to reflect that changes you may have made after completing task 2 and 3.

You are required to maintain a *history of your models development* on the Google Drive share you have been assigned. At regular intervals during the development generate a PDF file of the current version of your diagram. Include the date when you generate the PDF in the filename, eg 2018-24-03-er.pdf. You must have minimum of three PDF files that clearly show your project development.

You need to upload the *first* PDF file of your conceptual model by **5 PM Monday 9th April 2018 (week 6)** to the Google drive share you have been provided. **Failure to complete this requirement will mean that your assignment will not be assessed.**

For the final submission on Moodle, you need to submit only one PDF file of the conceptual model, which is the **final version** that you want us to mark.

2. Perform **normalisation** to the 3NF for the data as depicted in Appendix A

During normalisation, you must:

- **Not** add surrogate keys to the normalisation.
- Clearly identify the Primary Key in all relations.
- Clearly identify the partial and transitive dependencies (if they exist) in all 1NF relations. You may use a dependency diagram or use your own notation (see the normalisation sample solution for a possible alternative representation).

3. Based on the final version of your conceptual model, **prepare a logical level design** for the MonLib database.
  - The logical model must be drawn using the Oracle Data Modeler. The information engineering or Crow's foot notation must be used in drawing the model.
  - All entities depicted must be in 3NF
  - All attributes must be commented in the database.
  - Sequences must be used to generate numeric primary keys and check clauses must be applied to attributes where appropriate.
  - Be sure to include the legend as part of your model.
  - Similar to task 1, you are required to maintain a *history of your models development* on the Google Drive share you have been assigned. At regular intervals during development, combine all files of the project using zip to create a zip file. Give the zip file a filename which includes the date and time eg. 2018-08-04-lib.zip and place a copy on your assigned share. You must have a minimum of three archives (zip) files which clearly show your projects development.

4. **Generate the schema for the database** in Oracle Data Modeler and use the schema to create the database in your Oracle account. The **only** edit you are permitted to carry out to the generated schema file is to add header comment/s containing your details (student name/id) and drop sequence commands.
- Capture the output of the schema statements using the spool command.
  - Ensure your generated script includes drop table and sequence statements at the start of the script.
  - Name the schema file as **monlib\_schema.sql**.

## Submission Requirements

Due: Monday 23rd April (Week 8) 5PM

The following files are to be submitted:

- A pdf file containing your final version of conceptual model. Name the file **monlib\_conceptual.pdf**
- A pdf document showing your full normalisation of Appendix 1 showing all normal forms (UNF, 1NF, 2NF and 3NF). Name the file **monlib\_normalisation.pdf**
- A pdf file containing the final logical Model you created in Oracle Data Modeller. Name the file **monlib\_logical.pdf**.
- A zip file containing your Oracle data modeller project (in zipping these files be sure you include the .dmd file and the folder of the same name). Name the file **monlib\_oraclemodel.zip**.
  - This model must be able to be opened by your marker and contain your full model **otherwise your task 2 will not be marked**. For this reason, you should carefully check that your model is complete - you should take your submission archive, copy it to a new temporary folder, extract your submission parts, extract your model and ensure it opens correctly **before submission**.
  - To be eligible for marking a *clear development history* (at least three versions) must be maintained on your Google drive share.
- A schema file (CREATE TABLE statements) generated by Oracle Data Modeller. Name the file **monlib\_schema.sql**
- The output from SQL Developer spool command showing the tables have been created. Name the file **monlib\_schema\_output.txt**
- A pdf document containing any assumptions you have made in developing the model or comments your marker should be aware of. Name the file **monlib\_assumptions.pdf**

Note that there are **seven required files**. These files must be zipped into a single zip file named a1-<yourauthcatid>.zip e.g., a1-xyz123.zip before the assignment due date/time. Submit the a1-xyz123.zip to Moodle before the due date.

**Late submission will incur penalties as outlined in the unit guide.**

## 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. MadeTapD now include a digital auto-graded MySQL environment. -- Publisher.
Publisher:	Boston, MA, USA : Cengage Learning
Date Published:	2013
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