

School of Computing Technologies

ISYS1055 Database Concepts

Assessment 2: SQL Programming and Normalisation

**Assessment type:** PDF**Word limit:****Due Date:** Sunday, 22 May at 23:59 (Melbourne time)
– Week 11**Weighting:** 30%, 30 Marks

Overview

The objective of this assignment is to reinforce what you have learned in the lectures and tute/ lab sessions. Specifically, it covers the advanced concepts in the relational database design, using SQL for querying a relational database and analyse different database models for different applications.

Assessment criteria

This assessment will measure your ability to:

- Appreciate a good database design
- Apply good database design guidelines, such as normalisation, to build a well-designed database schema
- Write efficient SQL statements for retrieving data for specific user requirements
- Analyse different database models for different applications
- Write a technical report suitable for a non-technical audience, presenting your findings and recommendations.

Course learning outcomes

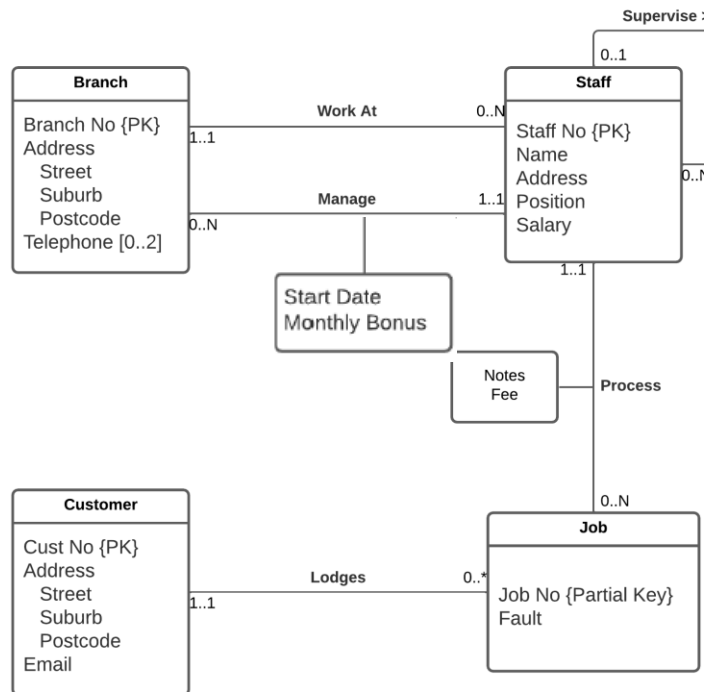
This assessment is relevant to the following course learning outcomes:

CLO1	describe the underlying theoretical basis of the relational database model and apply the theories into practice;
CLO3	develop a sound database design;
CLO4	develop a database based on a sound database design;
CLO5	Apply SQL as a programming language to define database schemas, update database contents and to extract data from databases for specific users' information needs.

Assessment details

Part A: Relational Database Design (30 Marks)

Consider the following ER diagram, which shows aspects of the business of a computer service company. As a part of the Assignment 1 activities, we have learned how to map entities and relationships in a E-R model into a relational database schema. We have used the 7-step mapping process.



As a result of **incorrect** application of 7-step mapping process, it was mapped into following relations.

Branch(BranchNo, B_Street, B_Suburb, B_Postcode, ManagerStaffNo*, Start_Date, Monthly_Bonus, Telephone1, Telephone2, Telephone3, S_Name*)

Staff(StaffNo, S_Name, S_Address, Position, Salary, Branch_No*, SupervisorStaffNo*, SupervisorStaffName)

Job(JobNo, CustNo*, Fault, StaffNo*, Notes, Fee)

Customer(CustNo, C_Street, C_Suburb, C_Postcode, C_Email)

- For each of these relations, write down all functional dependencies. If there are no functional dependencies among attributes, you must state so. Do not write down trivial functional dependencies, such as Email → Email.
- Write down the highest normal form each of these relations are in. For each of these relations, state the reasons why it doesn't meet the next normal form requirements. This is not required if the relation is in 3NF.
- If they are not in 3NF, decompose them into 3NF relations. Write down the full database schema at the end of this step, eliminating decomposed relations and replacing them with newly created relations.
- Where possible, combine the relations resulting from Part 3. Write down the full database schema at the end of this step, eliminating combined relations and replacing them with newly created relations.

5. Write down the final relational database schema.

Important: No marks are awarded to the final schema in Part 5 if you do not show the workings of decompositions (in Part 3) and combining relations (in Part 4). Indicate the primary key (underlined) and foreign key(s) (with an asterisk*) in each relation.

Part B: SQL (40 Marks)

LibraryDB is a database system that keeps track of information concerning the books and their circulation in an imaginary library.

Disclaimer: The data that populates the database are artificially constructed and by no means correspond to actual real-world data.

The schema for the LibraryDB database is given below.

```
borrow(transactionID, personID*, borrowdate, duedate, returndate)
author(authorID, firstname, middlename, lastname)
book_copy(bookID, bookdescID*)
book(bookdescID, title, subtitle, edition, voltitle, volnumber, language, place, year, isbn, dewey,
      subjectID*)
borrow_copy(transactionID*, bookID*)
person(personID, firstname, middlename, lastname, address, city, postcode, phonenumber,
        emailaddress, studentno, idcardno)
publisher(publisherID, publisherfullname)
written_by(bookdescID*, authorID*, role)
published_by(bookdescID*, publisherID*, role)
subject(subjectID, subjecttype)
```

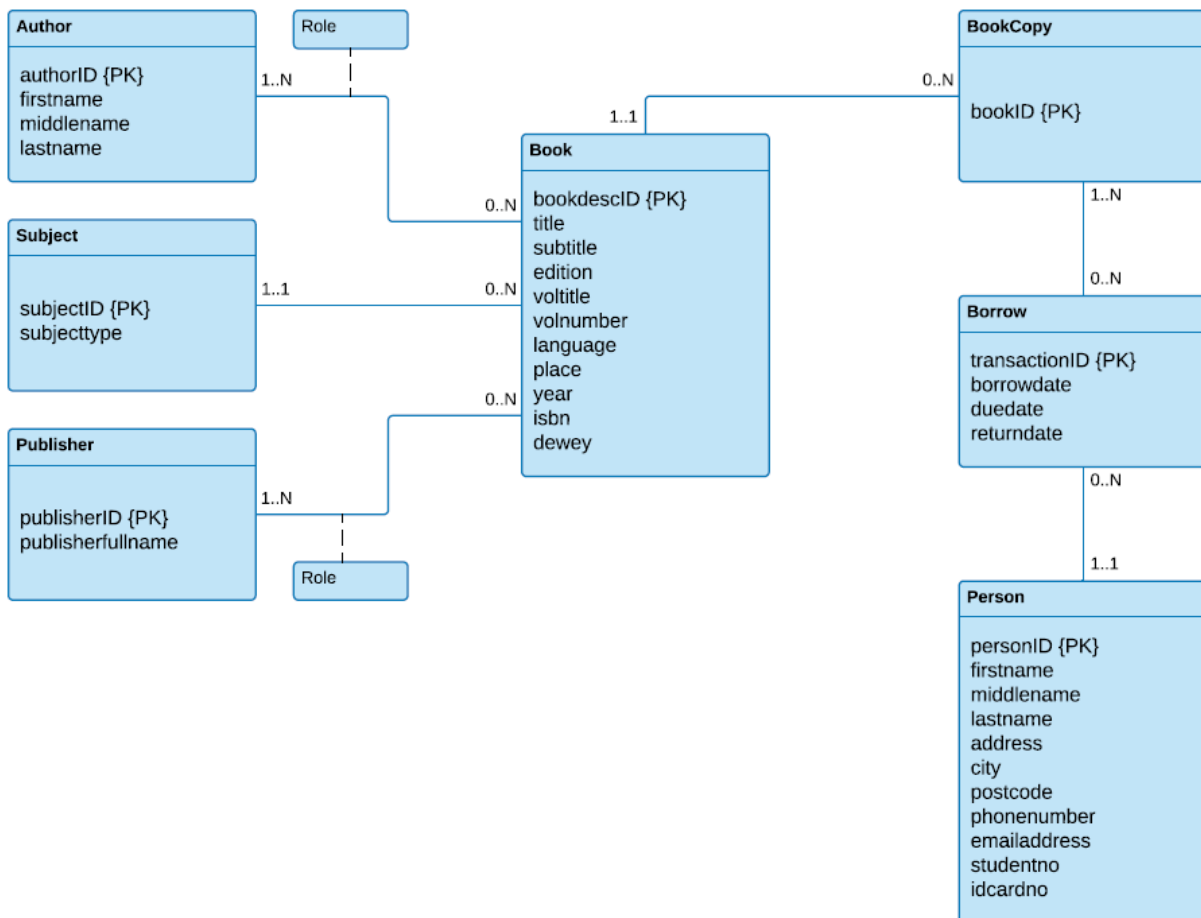
The primary keys are underlined. The foreign keys are denoted by asterisks (*).

Description of the schema

- person -- keeps track of the people who borrow books from the library. The attributes contain personal and contact information.
- author -- keeps track of personal information about authors.
- publisher -- keeps track of the publisher information. To make simple, most of the attributes have been truncated in the sample database.
- subject -- this relation keeps information about the subjects on which the library collection have books (such as Mathematics, Database, etc)
- book -- contains information about the books that are available in the library. Every book can have one or more physical copies in the collection. Each book can have one or more authors and it is published by one or more publishers.
- book_copy -- keeps track of the physical copies of the books in the library collection.
- borrow -- keeps track of the check-ins and check-outs of the books. Every transaction is done by one person, however may involve with one or more book copies. If there is no return date, it means the book has been checked out but not returned.
- written_by -- associates books with authors. A book may be associated with several authors and an author may be associated with several books. There is also an attribute 'role' that specifies the role of the author for the book (author/ editor/ translator/ etc).

- published_by -- associates publishers with books. There is an attribute 'role' here too.
- borrow_copy -- associates physical copies of books with a transaction. Members are allowed to borrow several books in a single transaction.

A conceptual data model (shown as an entity-relationship diagram) which represents these data is given below.



If you wish to do this part of the assignment from home, you can install SQLite Studio. The instructions for installing, configuring and using SQLite Studio is provided in the Week 3 Labsheet. Also included is the pre-built Library database in SQLite format (**Library.db**), available for downloading at: https://rmit.instructure.com/courses/79700/pages/course-resources?module_item_id=3204804.

Write SQL queries for the following tasks.

1. Display all the fields of books published after 1980 in the book relation only.
2. Display the first name and last name of the persons who translated books on the subject " IMAGE PROCESSING".
 - a. Write your query using a sub query.
 - b. Write your query using JOINS.
3. Who wrote the book " COMPUTING METHODS "? Display the first name, middle names, and last name of the author. Each author's role in the writing of the book is described in "role" attribute in written_by table.

4. Display the titles of books that have never been borrowed.
5. A borrower wants to borrow the book titled "COMPUTING METHODS", but all of its copies are already borrowed by others. Write two queries to display other recommended titles using the following methods.
 - a. Using partial matching of the book title -- note that the borrower is interested in a "COMPUTING" book.
 - b. By searching of other books written by the same author (i.e. the author of "COMPUTING METHODS")
6. Display the list of publishers who have published books on the subject "COMPUTING SOFTWARE". Your query should display publisher's full name.
7. Display the name of authors who have never written or translated any books.
 - a. Write your query using OUTER JOINS.
 - b. Write the query again without using OUTER JOINS.
8. Display full names of publishers with whom the author Alfred Aho did not publish his book(s). Your query must use EXISTS (/NOT EXISTS) clause.
9. Display the first name and last name of authors who had written more than 3 books. Along with each name, display the number of books as well.
10. Display the name of the author who had written most books. If there are more than one authors with the highest number of written books, show them all. Your query should show the names.

Provide detailed answers to the following questions.

11. According to this database schema, it is assumed that all books borrowed in one transaction are due at the same time and should be returned at the same time. However, this is an unreasonable assumption and quite contrary to the common practice across all forms of libraries. The books may be due on different days due to the renew activity, and they can be returned separately no matter if they have the same due date.
Provide the required changes to the database schema to accommodate this requirement. Your answer should include a portion of the ER model where the changes are applied and the updated relational database schema. If required, you may add new relations and attributes and also new foreign keys.
12. Libraries allow customers to place holds onto book, but not borrow it immediately. It can be borrowed by the customer who places the hold within one week. A customer can place hold onto a book unavailable and is put into a **holds queue**. When the book becomes available, it can be borrowed by the customer who places the hold first (i.e. the customer at the front of the queue). If the customer has not borrowed a book after holding it for one week, this book will be released for others to borrow. This database schema cannot keep information on holding of books. Extend this schema to accommodate this requirement. Your answer should include a portion of the ER model where the changes are applied and the updated relational database schema. If required, you may add new relations and attributes and also new foreign keys.
13. The library allows customers to renew the books in one transaction separately, and each book can be renewed for at most twice. Can this database schema support this operation? If so, how are they handled? If not, what changes required to the database schema?

Your answers should include a portion of the ER model where the changes are applied and the updated relational database schema. If required, you may add new relations and attributes and also new foreign keys.

Part C: Research Question (30 Marks)

[jobactive Job Seeker](#) is a digital free service app developed by the Department of Employment to help Australians find a job. With it, Job seekers can find and search for jobs around them using Geo location, suburb name, post code or a keyword. Job seekers can also find employment service providers around them with one tap or by using Geo location or post code. Thousands of jobs are advertised every day.

This app will record the following data about jobs: (1) Location; (2) Company; (3) Hours; (4) Job Title; (5) Job No; (6) Employment Type; (7) Duration; (8) Remuneration; (9) Position Description; (10) Closing Date. When a job is advertised, the employer will input these data that will be searched later in the system by job seekers to find their interested jobs.

Your Research Task:

As a database expert, you are invited to make a recommendation for the backend database solution to store these data about jobs. Commercial DBMS vendors can supply one of the following platforms for this purpose. (1) traditional relational database systems (such as Oracle and SQL Server); or (2) no-SQL database systems (such as MongoDB). A final decision will be made by Department of Employment based on your recommendations. Write a report identifying the advantages and disadvantages of both approaches specifically for this application and a conclusion making your recommendations. Your report may include case studies for both paradigms and draw conclusions based on their findings.

Approximate report length should be around 1000 – 1500 words. You must be careful about quoting texts extracted from other sources. You can paraphrase them with proper referencing. Before you start your report, please refer RMIT Library Referencing Guide, available at:

<https://www.rmit.edu.au/library/study/referencing>

Referencing guidelines

Use [RMIT Harvard](#) referencing style for this assessment.

You must acknowledge all the sources of information you have used in your assessments.

Refer to the [RMIT Easy Cite](#) referencing tool (<https://www.lib.rmit.edu.au/easy-cite/>) to see examples and tips on how to reference in the appropriated style. You can also refer to the library referencing (<https://www.rmit.edu.au/library/study/referencing>) page for more tools such as EndNote, referencing tutorials and referencing guides for printing.

Submission format

You should submit one PDF document with all answers together. You may use SQLiteStudio to work on your assignment. You must not submit result sets from SQL queries, only the SQL queries are to be submitted. You may use Word or any other word processor to compile your submission, by collating everything into one document. At the end, convert it into PDF format. Do not submit Word files. If that option is not available on your system there are free pdf converters online you can utilise. e.g. <http://convertonlinefree.com/>

Academic integrity and plagiarism

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas.

You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct.

Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

Penalties for late submissions

Late submissions of assignments will be penalised as follows. For 1 to 5 days late (including weekends), a penalty of 10% (i.e., 10% out of total marks, not 10% out of your marks) per day. For assignments more than 5 days late, 100% penalty applies.

Assessment declaration

When you submit work electronically, you agree to the [assessment declaration](#).