RMIT UNIVERSITY

School of Computing Technologies

ISYS1055/ISYS3412 (Practical) Database Concepts

Assessment 2: SQL Programming and Normalisation

Assessment type: PDF Word limit: N/A

Draft Due Date: Sunday, 12 May 2024 at 23:59 (Melbourne time) – Week 9 (otherwise 2 mark deduction)

Final Due Date: Sunday, 26 May 2024 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 the below aspects:

- 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

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.

RMIT UNIVERSITY

Assessment details

Note: Some questions vary between ISYS1055 and ISYS3412 (In Part B Q8). A breakdown of marks is available in the Assignment 2 rubric on Canvas.

Part A: Relational Database Design (14 Marks)

1. Answer questions below regarding the relational database design for the R University book loan management database.

Book (Call_No, Title, AuthorID, Author_Bio, Location, Copy_Barcode, Status)

Semantics of attributes is self-explanatory. Some additional notes are as follows:

- A call number is the combination of letters and numbers that indicates where books can be found in the library.
- A book may have multiple book copies. Each book copy has a unique copy barcode. All copies of a book share same call number.
- A book may have one or more authors and an author can write more than one book.
- Each author has some description (Author_Bio).
- Status presents the status of a book copy e.g. available, on loan

Answer following questions:

1.1 List all likely functional dependencies on the above relation Book according to the business rules. If there are no functional dependencies among attributes, you must state so. Do not write redundant or trivial functional dependencies (e.g. Call_No → Call_No). Specify all candidate keys for **Book** relation and explain your answer. (3 marks)

1.2 Please identify the highest normal form of **Book** relation. Explain your answer. (2 marks)

1.3 Following results in Question 1.2, if **Book** relation is not in 3NF, decompose it into relations in 3NF. Merge relations with a common primary key and remove any redundant subset relations. Check if the merged relations are in 3NF, and apply further decomposition if needed until all final relations are in 3NF. Write down the final relational database schema and indicate the primary key (underlined) and foreign key(s) (with an asterisk*) in each relation. (2 marks)

Important: No marks are awarded to the final schema in Question 1.3 if you do not show the workings of FDs (Question 1.1) and normal form reasoning (Questions 1.2).

2. Given the following relation for real estate sales

RealEstate (PropertyID, Address, AgentName, Price, Commission, AgentPhone, CustomerName, CustomerPhone, SalesTax)

and the following functional dependencies;

PropertyID → Address, Price

AgentName → AgentPhone

Price → SalesTax

CustomerName → CustomerPhone

AgentName, Price → Commission

Answer following questions:

2.1 Find All Candidate Keys of the **RealEstate** relation. Explain your answer. **(2 marks)**

2.2 Please identify the highest normal form of **RealEstate** relation. Explain your answer. **(2 marks)**

2.3 Decompose the **RealEstate** relation into 3NF relations if it is not in 3NF already. Merge relations with a common primary key and remove any redundant subset relations. Check if the merged relations are in 3NF, and apply further decomposition if needed until all final relations are in 3NF. Write down the final relational database schema and indicate the primary key (underlined) and foreign key(s) (with an asterisk*) in each relation. **(3 marks)**

Important: No marks are awarded to the final schema in Question 2.3 if you do not show the workings of FDs (Question 2.1) and normal form reasoning (Questions 2.2).

Part B: SQL (16 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, zipcode, phonenumber, emailaddress, classification, studentno, idcardno)
publisher(publisherID, publisherfullname)
written_by(bookdescID*, authorID*, role)
published_by(bookdescID*, publisherID*, role)
subject(subjectID, subjecttype)

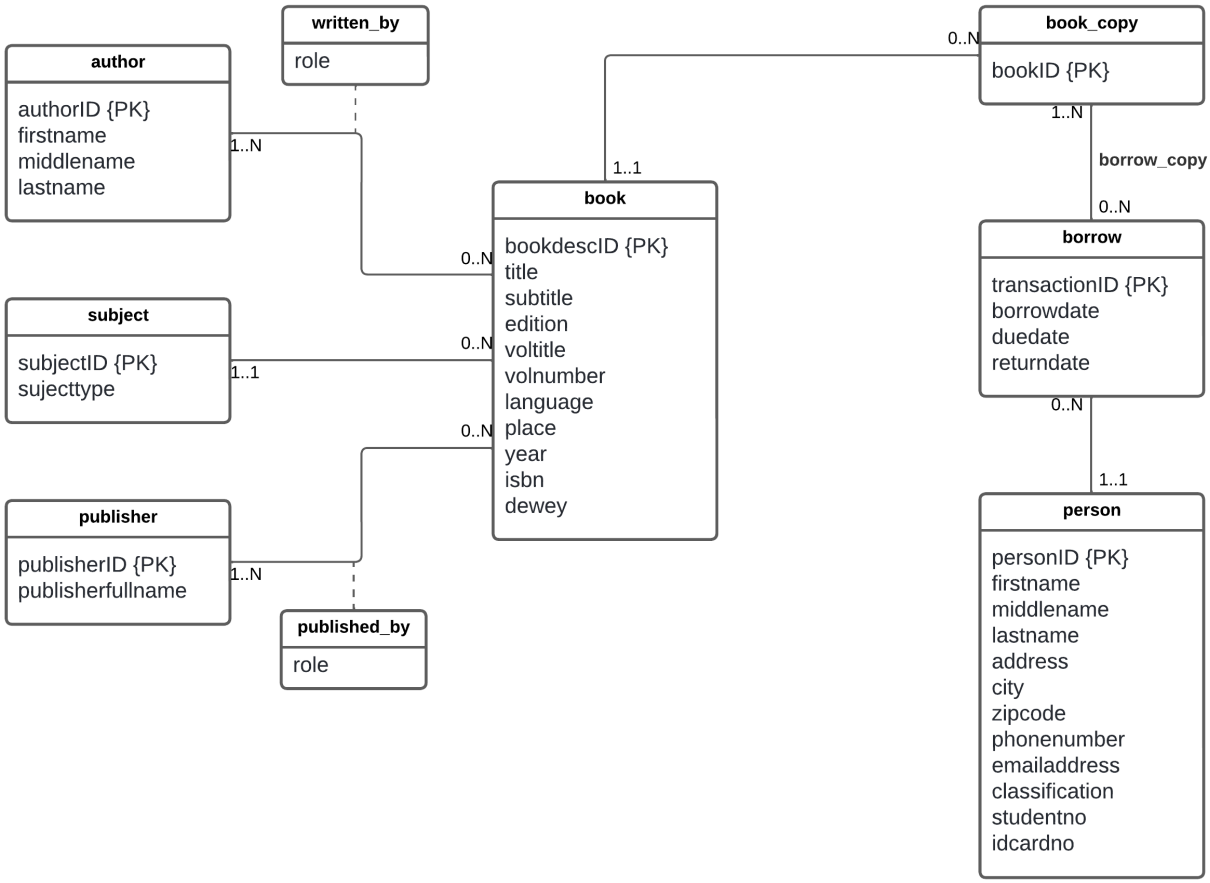
The primary keys are underlined. The foreign keys are denoted by asterisks (*).

Description of the schema

- **person** -- keeps track of the people who borrow books from the library (personal and contact details).
- **author** -- keeps track of personal information about authors.
- **publisher** -- keeps track of the publisher information.
- **subject** -- this relation keeps information about the subjects on which the library collection have books.
- **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 which takes values publisher/editor etc.
- **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.



You need to install SQLite Studio to complete questions in this part. The instructions for installing, configuring and using SQLite Studio is provided in the Week 2 Tute-Lab sheet. Also included is the pre-built Library database in SQLite format (**Library.db**), available for downloading at the following address:

https://rmit.instructure.com/courses/124806/pages/course-resources?module_item_id=6071010

Write **ONE** SQL query for each question. Note the below points:

- Your query must use only SQLite syntax.
- The output of your queries should not include duplicates, but you should use DISTINCT only if necessary.
- A hint of the expected number of results/rows for each query is given to help you determine if your understanding/solution is completely wrong. However it is important to note that a query that returns

the correct number of rows (or even the correct rows for the current instance) does not necessarily mean that the query is logically correct (or will always return the correct rows for different data).

- It is important that a query is logically correct with respect to the changing contents of the database. A query that returns the correct output for the current database content can still be logically incorrect. A query that returns nil output for the current database content can still be logically correct.
- You are advised to develop your query iteratively – start by selecting everything from the correct tables (including any appropriate JOIN conditions), then develop any GROUPing, the conditions for the GROUPing (HAVING) and WHERE conditions, the SELECT clause, and finally result ORDERing – this way you can verify that your query logic and result is correct so far before developing it further.
- While you can (and are encouraged to) include additional attributes in your select statement while debugging/developing, your final submitted answer must only return the requested attributes.
- Make use of the LOWER / UPPER functions to ensure your answer will work regardless of any variation in case.
- Avoid using LEFT/RIGHT joins and sub select queries unless absolutely necessary or requested.
- Do not use NATURAL joins or implicit joins unless the question explicitly requests it.
- Ensure that your queries are properly structured and formatted – Keywords in UPPERCASE, attribute and table names in lowercase, appropriate indentation and each clause on separate lines. eg:

```
SELECT ROUND(AVG(year), 0) AS "average year published"
FROM ((book b JOIN written_by wb
        ON b.bookdescid = wb.bookdescid)
             JOIN author a
                  ON wb.authorid = a.authorid)
WHERE subjectid IN (SELECT subjectid
                    FROM subject
                    WHERE subjecttype NOT LIKE '%computer%'
                    AND   subjecttype NOT LIKE '%science%')
ORDER BY "average year published" DESC;
```

If you run out of space horizontally you can reduce indentation for nested queries as follows:

```
SELECT ROUND(AVG(year), 0) AS "average year published"
FROM ((book b JOIN written_by wb
        ON b.bookdescid = wb.bookdescid)
             JOIN author a
                  ON wb.authorid = a.authorid)
WHERE subjectid IN (
        SELECT subjectid
        FROM subject
        WHERE subjecttype NOT LIKE '%computer%'
        AND   subjecttype NOT LIKE '%science%')
ORDER BY "average year published" DESC;
```

1. Display the name, and address of persons who fail to provide their emails. The output should have column headings “Name” and “Address”, and respectively display first name and last name separated by a space, and address and city separated by comma and space. (2 marks, 187 rows)
2. Compute the total number of books each publisher has published (as Publisher role). Output publisher full name together with its total number of published books, in decreasing order of total number of books. (2 marks, 194 rows)
3. Display publisher full name of all publishers that have never published / edited any books at all. There should not be any duplicate records in your result. (2 marks, 113 rows)

    a. Write your query using the JOIN operator.

    b. Write your query using an IN sub query.

4. The dates in relation “borrow” are stored as REAL type data in Julian Days. Find out the books that were overdue for more than 10 days when they were returned. Display book title, date of return, due date, and how many days were delayed from the due date. Display the dates in Australian date format. (2 marks, 19 rows)

5. Find out books that have never been borrowed. Display the book bookdescid, titles along with the year of publication, sorted by year of publication from newest to oldest. (2marks, 295 rows)

    a. Write your query using an EXISTS / NOT EXISTS sub query.

    b. Write your query using an IN / NOT IN sub query.

6. Find out the borrowers who had borrowed 5 or more books. Display “borrower name” (Concatenated first name and last name), and the number of books borrowed. (2 marks, 6 rows)

7. Display the name of authors who have written most books. If there is more than one author with the equal highest number of written books, show them all. Your query should output full name of those authors. (2 marks, 5 rows)

8. Find out the authors who are sole workers, i.e. only work on their own and never cooperate with other people on books as co-author, co-translator, etc. Display authorID, author name (first and last names, separated by a space) under the column heading of "Author Name". There should not be any duplicate records in your result.
    (This question is for ISYS1055 only; not required for ISYS3412) (2 marks, 321 rows)

8. Display the list of editors who have edited books on the subject "Communications". Your query should display publisher's full name.

    (This question is for ISYS3412 only; not required for ISYS1055) (2 marks, 4 rows).

Submission format

You should submit one PDF document with all answers together. (You should also submit a text/sql file for your queries in Part B questions 1-8 for ISYS1055 and questions 1-7 for ISYS3412). 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
- Utilising AI tools to generate content without referencing (or where AI tools are specifically prohibited)

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

Penalties for late submissions

Assignments received late and without prior extension approval or special consideration will be penalised by a deduction of 10% of the total score possible per calendar day late for that assessment.

Assessment declaration

When you submit work electronically, you agree to the assessment declaration.