Q1

/\*Q2 \*/

SELECT subject.subjectid, COUNT(book.BOOKDESCID) AS 'Total no. of books'

FROM subject

LEFT JOIN book

ON subject.SUBJECTID = book.SUBJECTID

GROUP BY subject.SUBJECTID

ORDER BY 'Total no. of books' DESC;

/Q3 a/

SELECT DISTINCT person.FIRSTNAME, person.LASTNAME, person.CITY

FROM person, borrow

WHERE person.personid = borrow.personid;

/Q3 b/

SELECT DISTINCT person.FIRSTNAME, person.LASTNAME, person.CITY

FROM person

JOIN borrow

ON person.personid = borrow.personid;

/Q3 c/

SELECT DISTINCT FIRSTNAME, LASTNAME, CITY

FROM person

WHERE personid IN (

SELECT personid FROM borrow

);

/\*Q4 \*/

SELECT DISTINCT bk.bookdescID AS 'Book Desc', bk.title AS 'Book Title'

FROM book bk JOIN book\_copy bkc

ON bk.bookdescID = bkc.bookdescID

JOIN written\_by w

ON bk.bookdescID = w.bookdescID

JOIN subject sub

ON bk.subjectID = sub.subjectID

WHERE sub.subjecttype = 'Databases'

GROUP BY bk.bookdescID, bk.title

HAVING COUNT(DISTINCT w.authorID) > 2;

/\*Q5 \*/

SELECT title AS "Book Title",

firstname || ' ' || lastname AS "Borrower Name",

date(br.returndate) AS "Date of Return",

date(br.duedate) AS "Due Date",

julianday(br.returndate) - julianday(br.duedate) AS "Days Delayed"

FROM book b

JOIN book\_copy bc

ON b.bookdescid = bc.bookdescid

JOIN borrow\_copy br\_c

ON bc.bookdescid = br\_c.bookid

JOIN borrow br

ON br\_c.transactionid = br.transactionid

JOIN person p

ON br.personid = p.personid

WHERE br.returndate > br.duedate

ORDER BY b.title;

/\*Q6 \*/

SELECT b.bookdescid AS 'Book description id', b.title AS title, b.year AS year

FROM book AS b

WHERE b.bookdescid NOT IN (

SELECT bc.bookdescid

FROM book\_copy bc JOIN borrow\_copy AS boc

ON bc.bookid = boc.bookid JOIN borrow AS bo

ON boc.transactionid = bo.transactionid

)

ORDER BY title ASC, year DESC;

/\*Q7 \*/

SELECT author.firstname, author.lastname, written\_by.role, book.title

FROM author JOIN written\_by

ON author.authorID = written\_by.authorID

JOIN book

ON written\_by.bookdescID = book.bookdescID

WHERE written\_by.role = 'Author'

AND author.authorID IN (

SELECT authorID

FROM written\_by

WHERE role = 'Author'

GROUP BY authorID

HAVING COUNT(DISTINCT bookdescID) > 1

)

ORDER BY author.lastname, author.firstname;

/\*Q8 \*/

SELECT b.TITLE

FROM book b

WHERE UPPER(b.TITLE) LIKE '%NETWORK%'

AND NOT EXISTS (

SELECT 1 FROM written\_by w

WHERE w.BOOKDESCID = b.BOOKDESCID

AND w.AUTHORID NOT IN (

SELECT a.AUTHORID

FROM author a

WHERE a.LASTNAME IN ('Miller', 'Noel')

AND a.FIRSTNAME IN ('Tim', 'Jason')

)

)

GROUP BY b.TITLE;

/\*Q9 \*/

SELECT DISTINCT b2.title, a.firstname || ' ' || a.lastname AS 'Author Name', b2.year

FROM book b1 JOIN written\_by wb1

ON b1.bookdescid = wb1.bookdescid

JOIN author a

ON wb1.authorid = a.authorid

JOIN written\_by wb2

ON a.authorid = wb2.authorid

JOIN book b2

ON wb2.bookdescid = b2.bookdescid

WHERE b1.title = 'COMPUTER SCIENCE' AND b1.bookdescid <> b2.bookdescid

ORDER BY a.lastname, a.firstname, b2.year DESC;

/\*Q10 \*/

SELECT pr.firstname || ' ' || pr.lastname AS 'Borrower', bk.title AS 'Book Title', sub.subjecttype AS 'Book Subject', strftime('%Y-%m-%d', br.borrowdate) AS 'Borrow Date', strftime('%Y-%m-%d', BR.returndate) AS 'Return Date'

FROM borrow AS br JOIN borrow\_copy AS bwcy

ON br.transactionID = bwcy.transactionID

JOIN book\_copy AS bkcy

ON bkcy.bookID = bwcy.bookID

JOIN book AS bk

ON bkcy.bookdescID = bk.bookdescID

JOIN subject AS sub

ON bk.subjectID = sub.subjectID

JOIN person AS pr

ON br.personID = pr.personID

WHERE sub.subjecttype LIKE '%Image Processing%'

ORDER BY br.borrowdate;

Part C: Research questions

1. The Dewey Call Number and the subject class of books are subject to data integrity restrictions imposed by the Dewey Decimal Classification (DDC) system. The Dewey Call Number must correspond to the correct topic class and any further subclasses of that subject. It follows that the subject class and all applicable subclasses must make up the Dewey Call Number. The database structure for the library does imposes this restriction. The "DeweyCallNumber" field in the "Book" database is specifically linked to the "DeweyClass" table, which establishes the subject class and additional subclasses. This makes sure that the database can only use Dewey Call Numbers and subject classes that are legitimate. The relational data model's data integrity methods can also be utilised to confirm the validity of the Dewey Call Number mentioned. This can be achieved by making sure that only legitimate Dewey Call Numbers can be input by designating the "DeweyCallNumber" field as a foreign key that references the "DeweyClass" database.

|  |  |  |
| --- | --- | --- |
| DeweyID | ClassName | Subclass |
| 005 | Computer programming, programs, data |  |

|  |  |  |  |
| --- | --- | --- | --- |
| BookID | Title | Author | DeweyID |
| 1 | Introduction to SQL Programming | John Smith | 005 |

In this example, the "DeweyID" foreign key in the "Book" table references the "DeweyID" primary key in the "DeweyClass" table, ensuring that the Dewey call number ("005") exists in the "DeweyClass" table. This enforces the data integrity constraint imposed by the DDC system.

1. The current Library database ER model has this problem because Person (as a library user) and Author are separate entity types with no relationships between them. This choice in design prevents the database from keeping track of relationships between a Person and a book's Author. A one-to-many link between Person and Author can be added to improve the ER model and the relational database structure that goes with it. This would make it possible to record the fact that a Person can borrow books from numerous Authors. This could be achieved by including a foreign key in the "Person" field that refers to the "Author" table.
2. A Person can borrow a certain book from the library, and a record of the transaction is preserved in the "borrow" table. This is how the real-world scenario of "users borrow books" is modelled in the current Library database. The "borrow" table has an artificial primary key "transactionID" which allows the database to uniquely identify each transaction. The database cannot uniquely identify each transaction without the artificial primary key, making it challenging to keep track of book borrowing. For instance, it would be challenging to determine which book was borrowed, when it was borrowed, and from whom without an artificial primary key. Furthermore, it would be challenging to maintain data integrity without the artificial primary key because it would be impossible to determine which book was borrowed when the same one was borrowed multiple times.