Assignment 2

In this assignment, you will practice working with SQL as discussed in lectures SQL Part 1, SQL Part 2, and Views.[1] Your solutions, containing the PostgreSQL statements for solving the problems, should be submitted to IU-Canvas in the required format. (See previous announcement.) It is strongly recommended that you include comments in this file to elaborate on your solutions.

In this assignment, we will use the following relation schemas about students and books.

Student($\underline{Sid}, Sname$)
Major($\underline{Sid, Major}$)
Book($\underline{BookNo}, Title, Price$)
Cites($\underline{BookNo, CitedBookNo}$)
Buys($\underline{Sid, BookNo}$)

The relation Major stores students and their majors. A student can have multiple majors but we also allow that a student has no major. major. A tuple (b, c) in the relation Cites indicates that the book with book number b cites the book with book number c. Note that a book may cite multiple other books. Also, a book does not have to cited.

The primary keys of the relations are the underlined attributes and we assume the following foreign keys:

Attribute in Relation	References Primary Key of Relation
Sid in Major	Sid in Student
BookNo in Cites	BookNo in Book
CitedBookNo in Cites	BookNo in Book
Sid in Buys	Sid in Student
BookNo in Buys	BookNo in Book

Furthermore, assume the following domains for the attributes:

Attribute	Domain
Sid	INTEGER
Sname	TEXT
Major	TEXT
BookNo	INTEGER
Title	TEXT
Price	INTEGER
CitedBookNo	INTEGER

[1] **Restrictions on SQL code**: You can use views but you can not use the `GROUP BY` clause and aggregate functions. You can also not use the `INNER JOIN` (or other joins) operators. Solutions with SQL statements that do not obey these requirements will not receive credit.

To do this assignment, you will have to create the above relations, including the primary and foreign keys. For data, use the `data.sql` file provided with this assignment.

Formulate the following queries in SQL. In these queries, you can not use views (including temporary and parameterized views).

1. Find the bno and title of each book that cost more than $10 and that was bought by a student who majors in 'CS' or in 'Math'.

   (a) Formulate this query in SQL without using subqueries and set predicates.

   (b) Formulate this query in SQL by using the `IN` set predicate.

   (c) Formulate this query in SQL by using the `SOME` set predicate.

   (d) Formulate this query in SQL by using the `EXISTS` set predicate.

2. Find the sid and name of each student who did not buy any book that cost more than $10.

   (a) Formulate this query in SQL without using subqueries and set predicates.

   (b) Formulate this query in SQL by using the `NOT IN` set predicate.

   (c) Formulate this query in SQL by using the `ALL` set predicate.

   (d) Formulate this query in SQL by using the `NOT EXISTS` set predicate.

   Hint: To solve this problem, first consider the problem of finding students who did buy a book that cost more than $10.

3. Find the bookno, title, and price of each book that is cited by at least two books that cost more than $15.

   (a) Formulate this query in SQL without using subqueries and set predicates.

   (b) Formulate this query in SQL by using the `IN` set predicate.

   (c) Formulate this query in SQL by using the `EXISTS` set predicate.

4. Find the sid and name of each student who majors in 'CS' and who bought a book with the highest price.

   (a) Formulate this query in SQL without using subqueries.

   (b) Formulate this query in SQL by using subqueries and set predicates.

5. Find the bookno and title of each book that is cited by a book that does not have the lowest price.

6. Find the sid and name of each student who has a single major and such that each book bought by that student is a book that cost more than \$10. (In other words, if the student bought any books, than he or she only bought books that cost more than \$10.)

7. Without using the `ALL` or `SOME` set predicates, find the booknos and titles of books with the next to lowest price.

8. Find the sid and name of each student who did not buy all books that cost more that \$75.

9. Find the sid and name of each student who bought all books that cost more that \$75.

10. Find each pair (s, b) where s is the sid of a student and b is the bookno of a book whose price is the highest among the books bought by that student.

11. Find the triples (b, s_1, s_2) such that b is the bno of a book and such that if student with sid s_1 bought book b then student with sid s_2 also bought book b. Furthermore, $s_1 \neq s_2$.

12. Find each pair (s_1, s_2) where s_1 and s_2 are the sids of two different students and such that student s_1 and student s_2 bought exactly one book in common.

13. Define a view `bookAtLeast30` that defines the books whose price is at least \$30.

    Consider the query "Find the sid and name of each student who buys at least two books that cost less than \$30."

    Write a SQL that uses the view `bookAtLeast30` to solve this query.

    After solving this problem drop the view `bookAtLeast30`

14. Reconsider the query in Problem 13. Redo this problem but this time by using temporary views (i.e., use the `WITH` statement).

15. Write a parameterized view `citedByBook (b integer)` that returns the relation of books that are cited by book b. (For each book returned include all information, i.e., bno, title, and price.)

    (a) Use this parameterized view to write a SQL query that finds the bno and price of each book that is cited by the book with bno 2001 and also cited by the book with bno 2002.

    (b) Use this parameterized view to write a SQL query that finds the bno and title of each book that cites at most 1 book.