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.

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

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, unless otherwise specified, you can not use views (including temporary and parameterized views).

1. Find the sid and name of each student who majors in CS and who bought a 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 only using the `IN` or `NOT IN` set predicates.

   (c) Formulate this query in SQL by only using the `SOME` or `ALL` set predicates.

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

2. Find the bookno, title, and price of each book that was not bought by any Math student.

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

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

   (c) Formulate this query in SQL by only using the `SOME` or `ALL` set predicates.

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

3. Find the bookno, title, and price of each book that cites at least two books that cost less than $60.

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

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

   (c) Formulate this query in SQL by only using the `EXISTS` or `NOT EXISTS` set predicates.

4. Find the sid and name of each student along with the title and price of the most expensive book(s) bought by that student.

   (a) Formulate this query in SQL without using subqueries. (Observe that a most expensive book is a book wherefore there does not exists another book that is more expensive.)

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

5. Find the sid and name of each student who bought at most one book that cost more than $20.

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

7. Find the bookno, title, and price of each book that cites a book which is not among the most expensive books.

8. Find the sid and name of each student who has a single major and such that none of the book(s) bought by that student cost less than $40.

9. Find the bookno and title of each book that is bought by all students who major in both 'CS' and in 'Math'.

10. Find the sid and name of each student who, if he or she bought a book that cost at least $70 then he also bought a book that cost less than $30.

11. Find each pair (s_1, s_2) where s_1 and s_2 are the sids of students who have a common major but who did not buy the same books.

12. Find the tuple (s_1, b_1, s_2, s_b) such that if the student with sid s_1 bought book with bookno b_1 then the student with sid s_2 did not buy the book with bookno b_2.

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 bought fewer than 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 `citesBooks (b integer)` that returns the relation of books that are cited by book *b*. (For each book returned by `citesBooks` include all information, i.e., bookno, title, and price.)

   (a) Use this parameterized view to write a SQL query that finds the bookno and title of each book that cites the book with bookno 2001 as well as cites a book that cost less than $50.

   (b) Use this parameterized view to write a SQL query that finds the bookno and title of each book that cites at least two books.