Module 14 Query Practice

1. Find the titles of books that cost more that $30.

QUERY

```
SELECT DISTINCT B.Title
FROM Book B WHERE B.Price > 30;
```

RES

Title
Theory o
Deductive
Discrete I
Graph Th
Compilers
Particle P
Complexi

Note that while creating alias of a table, we can actually ignore the “AS” keyword. Alias will be created with or without “AS” keyword.

2. Find Sid’s and Sname’s of students who major in “CS” and who buy a book that costs less that $20.

QUERY

```
SELECT S.Sid, S.Sname
FROM Student S, MajorsIn M, Buys T, Book B
WHERE  S.Sid = M.Sid and M.Major = 'CS' AND
       S.Sid = T.Sid and T.BookNo = B.BookNo AND B.Price < 20;
```

3. Find Sid’s and Sname’s of students who major in ”CS” but do not buy any book.

```
SELECT S.Sid, S.Sname
FROM Student S, MajorsIn M
WHERE M.Sid = S.Sid and M.Major = 'CS' AND
      NOT EXISTS( select *
                  from Buys T
                  where S.Sid = T.sid);
```

The inner query returns the details of all the students who buy books and the NOT EXISTS operator negates the result, providing us details of the students who do not buy any book.

4. Find Sid's and Sname's of students who neither major in "CS" nor buy any book that is not cited.

QUERY

```
SELECT S.Sid, S.Sname
FROM   Student S
WHERE  S.Sid not in (select M.Sid from MajorsIn M where M.Major = 'CS') AND
       S.Sid not in (select T1.Sid
                     from   Buys T1
                     where  T1.BookNo not in (select T2.CitedBookNo
                                              from Cites T2));
```

The NOT IN part filters out the records with not major in CS and books that are not cited.

5. Find the BookNo's of books that are cited by at most two books.

QUERY

```
(SELECT B.BookNo
FROM Book B, Cites C
WHERE B.BookNo = C.CitedBookNo
group by(B.BookNo)
having count(DISTINCT C.BookNo) <= 2)
UNION
(SELECT  B.BookNo
 FROM    Book B WHERE
         B.BookNo not in
         (select C.CitedBookNo
          from Cites C));
```

If you look at the question carefully, it asks for books that are cited by at most 2 books, which means books cited by 0,1 or 2 books. The first query returns all the books cited by <= 2 books and the second query returns details of such books which are not cited by any.

6. Find the BookNo's of books that cite books that cost less than $20.

QUERY

```
SELECT C.BookNo
FROM   Cites C, Book B
WHERE  C.CitedBookNo = B.BookNo and B.Price < 20;
```

RE

7. Find the Sid's of students that have exactly two majors and buy at least one book.

QUERY

```
SELECT DISTINCT S.Sid
FROM   Student  S, Buys B, MajorsIn M
WHERE  S.Sid = B.Sid and S.Sid = M.Sid
group by(S.Sid)
having count(DISTINCT M.Major) = 2;
```

RES

The count operator takes care of exact 2 majors whereas a simple join between "Student" and "Buys" tables ensures that a student buys at least one book.

8. Find the records in the form (s1,s2,b) where s1 and s2 are different Sid's and b is the BookNo of a book that is either bought by s1 or by s2 (So the books are not bought by both s1 and s2).

QUERY

```
SELECT S1.Sid AS Sid1, S2.Sid AS Sid2, T1.BookNo
FROM   Student S1, Student S2, Buys T1
WHERE  S1.Sid <> S2.Sid AND
       (S1.Sid = T1.Sid OR S2.Sid = T1.Sid) AND

       (S1.Sid, S2.Sid, T1.BookNo) NOT IN

       (SELECT T2.Sid, T3.Sid, T2.BookNo
        FROM   Buys T2, Buys T3
        WHERE  T2.Sid <> T3.Sid AND T2.BookNo = T3.BookNo);
```

The result actually contains 208 records but I couldn't output all the records, hence only first 10 rows are displayed. The question asks for the result in the format <Student1, Student2, BookNo> where the book is bought by either Student1 or Student2 but not both. In scenarios like this, *self join* plays an important role.

We need the details from the same table where one record is not equal to another. Hence the line S1.sid <> S2.sid in the first query ensures that S1 and S2 are different in each resultant record.

The second query with NOT IN operator contains details in the form <S1, S2, B> where a book is bought by both S1 and S2. For example, book 2002 is bought by both student 1001 and 1002. We don't need records like this and hence the NOT IN negates these results from the output.

9. Find the records in the form (s1,s2) where s1 and s2 are different Sid's and such that s1 and s2 buy exactly one book in common.

QUERY

```
SELECT    B1.Sid, B2.Sid
FROM      Buys B1, Buys B2
WHERE     B1.Sid <> B2.Sid AND B1.BookNo = B2.BookNo
group by  B1.Sid, B2.Sid
having    count(B1.BookNo) = 1;
```

Here again the *self join* ensures that different Sid's are considered and the count operator filters students who buy only one book in common.

10. Find the Sid's of students who buy all books that cost more than $30.

QUERY

```
SELECT  S.Sid
FROM    Student S
WHERE   NOT EXISTS( select *
                    from  Book B
                    where  B.Price > 30 and
                    NOT EXISTS (select *
                                from  Buys T
                                where T.Sid = S.Sid
                                and T.BookNo = B.Bookno));
```

The sub query (after the first NOT EXISTS) returns details of such students who have not purchased some books that cost more than $30. They might have purchased books that cost more than $30 but not all of them. Hence the first NOT EXISTS operator negates such students from the result.

11. Find the records in the form (s1,s2) where s1 and s2 are different Sid's and such that student s2 buys all the books student s1 buys.

QUERY

```
SELECT DISTINCT S1.Sid, S2.Sid
FROM   Buys S1, Buys S2
WHERE  S1.Sid <> S2.Sid
       and NOT EXISTS( select T2.BookNo
                       from  Buys T2
                       where  S1.Sid = T2.Sid and
                       T2.BookNo NOT IN
                       (select T1.BookNo from Buys T1 where S2.Sid = T1.
```

The sub query returns such books which are bought by S1 but not by S2. Inside the sub query, the first join with T2 and S1 gives us the books bought by S1 and the NOT IN operator filters the students(S2) who did not buy the same book.

For example, the sub query yields Book 2001 which is bought by 1002(S1) but not 1001(S2).

Now the initial NOT EXISTS operator removes the above results, which essentially gives us the results where S2 buys all the books which are bought by S1.

12)Find the BookNo's of books with the lowest price.

QUERY

```
SELECT B.BookNo
FROM   Book B
WHERE  B.Price = (select min(DISTINCT B1.Price)
                  from  Book B1);
```

RESULT

BookNo
2006

This query should be fairly straightforward to understand.

13)For each book specify the number of books they cite, provided that this number is less than 20.

QUERY

```
(SELECT C.BookNo, count(DISTINCT CitedBookNo)
 FROM   Cites C
 group  by(C.BookNo)
 having count(CitedBookNo) < 20)
 UNION
(SELECT B.BookNo, 0
 FROM   Book B
 WHERE  B.BookNo not in (select C1.BookNo from Cites C1)
);
```

RESULT

BookNo	count(DISTINCT CitedBookNo)
2001	2
2003	3
2002	0
2004	0
2005	0
2006	0
2007	0
2008	0
2009	0
2010	0

The first query essentially gives us the result but this does not cover the books that do not cite any other books. Hence the second query, with 0 appended in the count column combines the above result set with rest of the books that do not cite any other books.