Quiz #2

Variant 1

Part 1 (5 points)

Please answer the following questions (Explain your answer):

1. Suppose relation R(A,C), S(B,C,D) has the following tuples:

A	C
1	0
4	9
2	7
4	25
7	31

B	C	D
8	7	24
11	5	19
13	25	10
34	4	13

Compute the natural join of R and S. Assume each tuple has schema (A,B,C,D).

2. Consider the following schema:

```
 Book(ISBN, title, year) // ISBN and title cannot be NULL
 Author(ISBN, name) // ISBN and name cannot be NULL
```

and the following view definition over this schema:

```
  Create View V as
    Select Book.ISBN, count(*)
    From Book, Author
    Where Book.ISBN = Author.ISBN
    And Author.name Like 'A%'
    And Book.year > 2000
    Group By Book.ISBN
```

This view is not updatable according to the SQL standard, for a number of reasons. Write down all reasons for the view being non-updatable according to the standard?

3. Suppose a table T(A,B,C) has the following tuples: (1,1,3), (2,2,3), (5,1,4), (2,3,5), (2,4,1), (2,3,4), and (3,2,6). Consider the following view definition:

```
   Create View V as
     Select A+B as D, C
     From T
```

Consider the following query over view V:

```
   Select D, sum(C)
   From V
   Group By D
   Having Count(*) <> 1
```

Calculate the query result.

4. Consider the following relational schema:

```
   Course(courseName unique, department, instrID)
   Instructor(instrID unique, office)
   Student(studentID unique, major)
   Enroll(studentID, courseName, unique (studentID,courseName))
```

Suppose there are five types of queries commonly asked on this schema:

- Given a course name, find the department offering that course.
- List all studentIDs together with all of the departments they are taking courses in.
- Given a studentID, find the names of all courses the student is enrolled in.
- List the offices of instructors teaching at least one course.
- Given a major, return the studentIDs of students in that major.

Create for each query create appropriate index.

5. Consider the following query:

```
Select * From Apply, College
Where Apply.cName = College.cName
And Apply.major = 'CS' and College.enrollment < 5000
```

Which of the following indexes could NOT be useful in speeding up query execution?

A. Tree-based index on Apply.cName

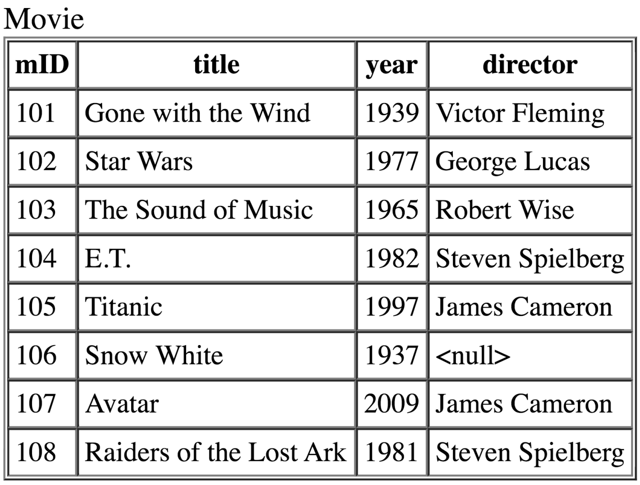
B. Hash-based index on Apply.major

C. Hash-based index on College.enrollment

D. Hash-based index on College.cName

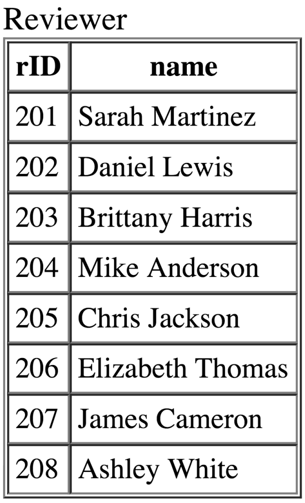
Part 2 (5 points)

Please write SQL queries for following tasks. Consider following schemas:



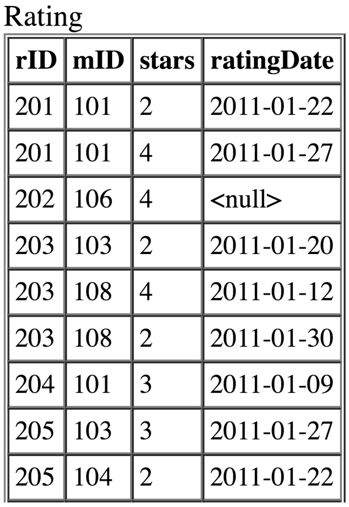
Movie

mID	title	year	director
101	Gone with the Wind	1939	Victor Fleming
102	Star Wars	1977	George Lucas
103	The Sound of Music	1965	Robert Wise
104	E.T.	1982	Steven Spielberg
105	Titanic	1997	James Cameron
106	Snow White	1937	<null>
107	Avatar	2009	James Cameron
108	Raiders of the Lost Ark	1981	Steven Spielberg



Reviewer

rID	name
201	Sarah Martinez
202	Daniel Lewis
203	Brittany Harris
204	Mike Anderson
205	Chris Jackson
206	Elizabeth Thomas
207	James Cameron
208	Ashley White



Rating

rID	mID	stars	ratingDate
201	101	2	2011-01-22
201	101	4	2011-01-27
202	106	4	<null>
203	103	2	2011-01-20
203	108	4	2011-01-12
203	108	2	2011-01-30
204	101	3	2011-01-09
205	103	3	2011-01-27
205	104	2	2011-01-22

6. Find the names of all movies which have three or more ratings.

7. For each movie that has at least one rating, find the highest number of stars that movie received. Return the movie title and number of stars. Sort by movie title.

8. Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date.

9. Create view for the next statement. Find the titles of all movies directed which have review from Mike Anderson.

10. Create index to speeding up previous query.