Solutions for Second-midterm practice exam.

Neo4J Queries:

Find any player who played for a team in the same state he was born in.

```
match(p1:Player)-[:PlayedIn]->(t:Team),
(p1)-[:BornIn]->(s:State)<-[:LocatedIn]-(t:Team)
return p1.name, t.name, s.name
```

Find any two teams that shared two or more players:

```
match(t1:Team)<-[:PlayedIn]-(p:Player)-[:PlayedIn]->(t2:Team)
with t1, t2, count(*) as cnt
where cnt >=2
return t1.name, t2.name, cnt
```

MongoDB Queries

```
db.teams.aggregate([
 { $lookup:
       {
          from: "players",
          localField: "name",
          foreignField: "current_team",
          as: "current_players"
       }
   },
   { $unwind: "$current_players" },
   { $project: {_id: 0, name: 1, playerName: "$current_players.name"}
}
]).pretty()


db.players.aggregate([
   { $match: {birth_state: "California" } },
   { $group: {_id: "$current_team", averageAge: {$avg:"$age"} } },
   { $project: {_id: 1, averageAge: 1} }
])
```

Question 3)

a. (3 points) **CAN ARISE**

```
T1: S(A)R(A)          S(B)R(B) Rel.
T2:      S(A)R(A)                 S(B)R(B)    S(B)R(B) Rel.
T3:                      S(B)R(B) Rel.
```

b. (3 points) **CAN ARISE**

```
T1: X(B)W(B)            S(A)R(A)Rel.
T2:           S(A)R(A)                    X(A)W(A) X(B)W(B)Rel.
T3:     S(A)R(A)               X(B)W(B)Rel.
```

c.

Schedule A. Because the schedule has only read operations, read locks are not necessary since no conflicts are possible (a read and write on the same element or two writes on the same element). read-uncommitted isolation level avoids read locks altogether leading to lower overhead.

Question 4)

Which FD is not enforced in the database instance above?
Library_ID → lib_address

Fix the data so the FD is enforced. You only need to rewrite one row. (1 point)

789	L1	**11 Burnside**	Jefferson	Intro. Databases	3

- Question 6)
  It turns out we need addToSet aggregate function to create a set of all titles for each book-ISBN. Then we can count the number of titles for each ISBN and see if there exists an ISBN with two or more titles, which is a violation for the FD.

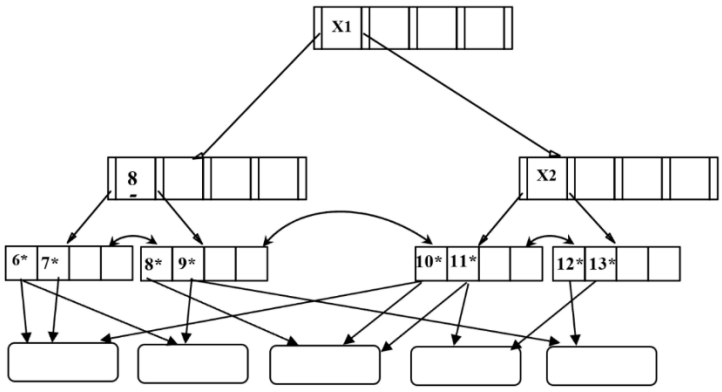
```
db.books.aggregate([
            {$group: {
             _id: "$Book-ISBN",
             allTitles:{addToSet:"$book_title"}
                       },
            {$unwind:"$allTitles"},
            {$group:{_id:"$_id", count:{$sum:1}}
            {$match: {count:{$gt:1}},
            ])
```

Question 7 Difficulty: 1

What does availability mean?

- ➡ The system features are available.
- All database nodes are available.
- All database replicas are available.
- All database shards are available.

Question 8 Difficulty: 1



In the index above, what is the value of X1 and X2?

- 8, and 10, respectively
- 9, and 11, respectively
- ➡ 10, and 12, respectively
- 12, and 11, respectively

Question 9 Difficulty: 1

The index in the previous question is:

Clustered, dense

Clustered, sparse

➡ Unclustered, dense

Unclustered, sparse

Question 10 Difficulty: 1

Considering OLTP and OLAP transactions. Which of the following statements is true?

➡ OLAP queries are usually slow while OLTP queries are usually fast.

OLAP queries are usually fast while OLTP queries are usually slow.

None of the above.