CSCI 4320
Spring 2023
Exam 3

Last Name: _____

First Name: _____

Write your answers on the answer sheets provided.

1. (4 pts) Name one advantage and one disadvantage of indexing fields in a table.
2. (4 pts) Under what circumstances would a hashed index be prefered to an ordered index (and vice-versa)?
3. (4 pts) What are the two parts of a trigger? What does each accomplish?
4. (3 pts) Under what circumstances might one prefer a NoSQL DBMS to a relational DBMS?
5. (3 pts) What does the MongoDB `$near` operator do?
6. Answer the following questions about the CAP theorem.
   (a) (6 pts) Expand the acronym CAP and for each word give a brief definition each word.
   (b) (4 pts) Summarize the theorem in terms of the way it applies to the design of distributed databases.
7. Suppose the following tables represent annual rainfall measurements by year for various locations.

location

id	city	state
23	Abilene	TX
24	Anson	TX
25	Baird	TX
26	Clyde	TX
27	Buffalo Gap	TX
49	Hobbs	NM
...	...	...

rainfall

id	year	amount
23	2006	26.4
23	2007	43.2
23	2008	33.0
24	2006	31.0
24	2007	null
24	2008	29.0
25	2007	null
25	2008	null
...	...	...

   (a) (2 pts) Write a(n) SQL statement that will create an ordered index on the `city` field of the `location` table.
   (b) (4 pts) Write a(n) SQL statement that will list all rainfall by year for each city in Texas after 2006. The query should not show null rainfall entries and should clump entries for the same city together (in order of year).
   (c) (10 pts) Write a command to create a PL/PgSQL function named `fillInTheGaps` that is expected to be executed once a year or so. The function should check if there is an entry in the `rainfall` table for each city for the current year. Any city not having an entry in the `rainfall` table for the current year should have a new entry inserted with `null` in the `amount` column. You may assume that the year field is stored as an integer.

(d) (6 pts) Suppose the database above is to be converted to MongoDB and a decision has been made that rainfall data will be embedded with each city rather than having two separate collections for rainfall and location. Write the MongoDB commands that would be necessary to insert Anson's rainfall data into a collection called `rain` within a database named `alltherain`.

(e) (4 pts) Write a MongoDB query that will display all rainfall data from Texas cities assuming the structure you used in problem 7d was applied to insert thousands of entries from across 50 states. Construct the query so that the built-in `_id` field is not displayed.

8. Suppose we have two related MongoDB collections whose contents are structured as follows. NOTE: Questions pertaining to this code are on page 3.

```
// artist collection (partial)
{
   "_id" : ObjectId("5e9b5fbe97a3852304b54f48"),
   "name" : "Dog Man",
   "hometown" : "Coyote, Wyoming"
}
{
   "_id" : ObjectId("5e9b5fbe97a3852304b54f49"),
   "name" : "The Computering Whiners",
   "hometown" : "Seattle, Washington"
}
// etc.

// album collection (partial)
{
   "_id" : ObjectId("5e9b5fbf97a3852304b54f4b"),
   "artist_id" : ObjectId("5e9b5fbe97a3852304b54f48"),
   "title" : "Howling Knights",
   "year" : 2003,
   "genre" : [
      "Jazz",
      "Soul"
   ]
}
{
   "_id" : ObjectId("5e9b5fbf97a3852304b54f4c"),
   "artist_id" : ObjectId("5e9b5fbe97a3852304b54f48"),
   "title" : "Woof, Woof to You Too!",
   "year" : 2004,
   "genre" : [
      "Pop"
   ]
}
// etc.
```

(a) (3 pts) Write a MongoDB query to list all albums that were released prior to 2000.

(b) (10 pts) Write a MongoDB aggregate query with the following pipeline stages: join the two tables, hide the `_id` and `artist_id` fields, show only albums that have "Pop" in the list of generes, sort with most recent albums listed first.

(c) (4 pts) Give one advantage of keeping the data in separate collections and one advantage that would be gained by nesting the artist information inside each album document.

(d) (6 pts) Write a section of MongoDB code that will iterate over every document in the album collection and will display the first genre of each album.