Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Practical No. 7

Study of MongoDB

Perform following problem statements using MongoDB

Problem Statement 1:

- First off, you need a database to connect to. MongoDB doesn't have a "create database" command. Instead, it is going to create one for you when you try to save something into it.
- Install and Connect to the mongoDB.
- Create a collection called 'games'. We're going to put some games in it.
- Add 5 games to the database. Give each document the following properties: name, genre, rating (out of 100). If you make some mistakes and want to clean it out, use remove()on your collection.
- Write a query that returns all the games.
- Write a query to find one of your games by name without using limit(). Use the findOne method. Look how much nicer it's formatted!.
- Write a query that returns the 3 highest rated games.
- Update your two favourite games to have two achievements called 'Game Master' and 'Speed Demon', each under a single key. Show two ways to do this. Do the first using update()and do the second using save(). Hint: for save, you might want to query the object and store it in a variable first.
- Write a query that returns all the games that have both the 'Game Maser' and the 'Speed Demon' achievements.
- Write a query that returns only games that have achievements. Not all of your games should have achievements, obviously.

Problem Statement 2: MapReduce question:

- Use mongoDB map-reduce on the example set of game data.
  http://docs.mongodb.org/manual/applications/map-reduce/
- Example document:

```
{
        "_id" : ObjectId("5144e16e54f9ef8613927ec4"),
        "name" : "Ape Escape",
        "publisher" : "KOEI Co., Ltd.",
"released" : "1995-03-27",
"rating" : 99,
"scores" : [
                    {
                    "name" : "derrick",
                    "score" : 705
                    },
                    {
                    "name" : "tim",
                    "score" : 379
                    },
                    {
                    "name" : "bryan",
                    "score" : 810
                    }
            ]
}
```

- Write a reduce that calculates the total score from all games for each player and check the output.

Problem Statement 3: REST API:

- Use the REST API to show all the game data stored in the db from the games collection.
- Output all of the available returned data in an html table in the following format:

Game	Publisher	Release Date	Rating	Average Score

Note:

1. Create a **document** of the above website with screenshots.
2. Scan the document and **create a pdf file** with **“ExamSeatNum_P#PS#” as its name**.
3. Upload the file on the **WCE Moodle** before the given deadline.