Laboratory Exercise
CRUD Operation

Objectives:

At the end of the exercise, the students should be able to:

- Insert, read, update, and delete a document data; and
- Use various operator on querying documents.

Software Requirement:

- MongoDB Community Server 4.4 or higher
- Mongo Shell or Internet Browser

Procedures:

1. Using Mongo shell, create a database named *MongoLab1* and insert the documents below into the *employees* collection.

```
{
        id: 1,
        Name: “Steve Badiola”,
        Salary: 16,099.55,
        Position: “President”
        Rank: 1
        ReportingTo: null
}, {

        id: 2,
        Name: “Jamir Garcia”,
        Salary: 14,567.12,
        Position: “Vice-President”
        Rank: 2
        ReportingTo: [President]

}, {

        id: 3,
        Name: “Reg Rubio”,
        Salary: 13, 891.22,
        Position: “Secretary”
        Rank: 3
        ReportingTo: [Vice-President]

}, {

        id: 4,
        Name: “Ian Tayao”,
        Salary: 13,000,
        Position: “Treasurer”
        Rank: 4
        ReportingTo: [Secretary, Vice-President]
}
```

2. **Remove** all the “*ReportingTo*” field in *employees* collection where its value is **null**.
3. Using the ***$inc*** operator, **update** all the documents by increasing their salary by *5000*.
4. **Update** the document of “*Reg Rubio*” and “*Ian Tayao*” by adding the “*President*” to their Reporting field.
5. **Find** the document where their salary is above “*21,000.00*”.
6. **Find** the document where their name starts with “S” and “R”.
7. **Find** the document that is not reporting to the “*President*”.
8. **Update** all the document by adding a sub-document named “*contact*” with a key-value, “*email”: “name@gov.ph*” (Example: steve.badiola@gov.ph) and “*phone”: “+1 1234567*” (**note**: +1 value should depend on their rank number).

Output:

```
{
        "_id" : 1,
        "Name" : "Steve Badiola",
        "Salary" : 21599.55,
        "Position" : "President",
        "Rank" : 1,
        "contact" : {
                "email" : "steve.badiola.gov.ph",
                "phone" : "+1 1234567"
        }
}

{
        "_id" : 2,
        "Name" : "Jamir Garcia",
        "Salary" : 19567.12,
        "Position" : "Vice-President",
        "Rank" : 2,
        "ReportingTo" : [
                "President"
        ],
        "contact" : {
                "email" : "jamir.garcia@gov.ph",
                "phone" : "+2 1234567"
        }
}

{
        "_id" : 3,
        "Name" : "Reg Rubio",
        "Salary" : 18891.22,
        "Position" : "Secretary",
        "Rank" : 3,
        "ReportingTo" : [
                "Vice-President",
                 “President”

        ],
        "contact" : {
                "email" : "reg.rubio@gov.ph",
                "phone" : "+3 1234567"
        }
}
```

```
{
        "_id" : 4,
        "Name" : "Ian Tayao",
        "Salary" : 18000,
        "Position" : "Treasurer",
        "Rank" : 4,
        "ReportingTo" : [
                "Secretary",
                "Vice-President",
                “President”
        ],
        "contact" : {
                "email" : "ian.tayao@gov.ph",
                "phone" : "+4 1234567"
        }
}
```

GRADING RUBRIC:

CRITERIA	PERFORMANCE INDICATORS	POINTS
Correctness	The code produces the expected result.	30
Logic	The code meets the specifications of the problem.	30
Efficiency	The code is concise without sacrificing correctness and logic.	20
Syntax	The code adheres to the rules of the database management system.	20
Total		**100**