School of Computing and Information Technology

Student to complete:

Family name	
Other names	
UOW Student number	

CSCI235
Database Systems

Final Examination Paper
Session 2 2020
3 June 2020

Exam duration	3 hours and 10 minutes
Weighting	40% of the subject assessment
Marks available	40 marks
Items permitted by examiner	Text-book, Lecture slides, and Tutorial notes
Directions to students	4 questions to be answered. Marks for each question are shown beside the question. All answers must be written / typed neatly. **No asking for explanation of question is allowed during the examination. However, if you suspect that there is a typo or incorrect question, you can ask the invigilator to confirm.**

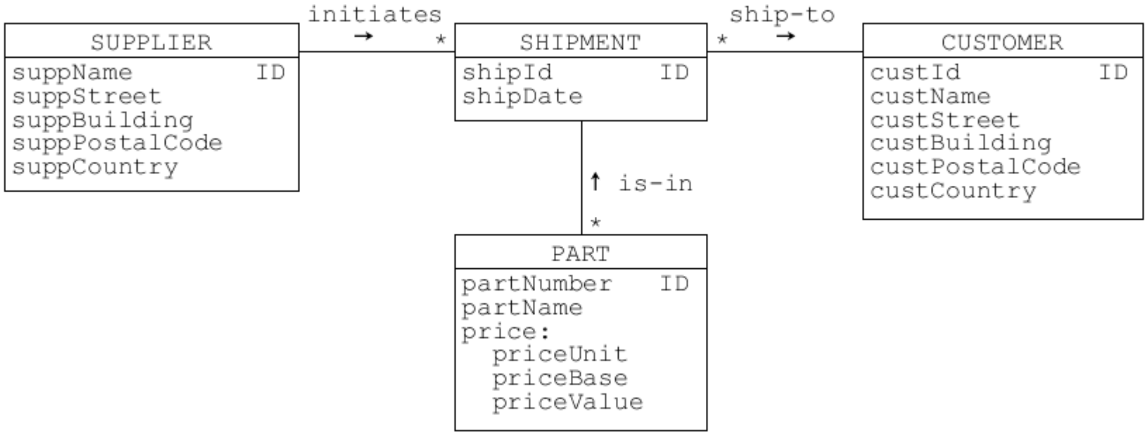
This examination is a take-it-home examination to be done on-line on the date of examination.

Version 1.1

Question 4 - (Total 10 marks)
JSON/BSON document

Time allocated: 40 minutes
Start time: 12:25 pm SGT
End time: 1:05 pm SGT
Submission time start: 1:00 pm SGT
Submission time end: 1:10 pm SGT

a) Consider the following conceptual schema of a sample database that contains information about the supplier’s shipments to customer.



Write a sample JSON document that has a structure consistent with the conceptual schema given above. Your document must contain information about at least one supplier, two shipments, two customers, and three parts; one of the shipment contains two parts and the other shipment contains one of the remaining part. The values for the attributes are up to you but must be sensible.

(5.0 marks)

b) Consider a sample BSON document given below. Assume that all documents in a collection `empeProject` has the same structure as the document shown below:

```
db.empeProject.insert([ {
  "Employee": [ { "empeId": "e001",
              "fName": "James",
              "lName": "Bond",
              "email": "jamesbond@hotmail.com",
              "experience": [
                        "Database Design",
                        "SQL",
                        "Java" ]
                  },
                  { "empeId": "e002",
               "fName": "Harry",
               "lName": "Potter",
               "experience": [
                        "Data Warehouse",
                        "SQL",
                        "Spark Scala",
                        "Java Scripts" ]
                  } ],
  "Project": [ { "projectId": "p001",
             "projectTitle": "Install MongoDB" },
                 {"projectId": "p002",
             "projectTitle": "Install Oracle" },
                 {"projectId": "p003",
             "projectTitle": "Install Hadoop" } ],
  "EmployeeProject": [ {  "empeId": "e001",
                     "projectId": "p001",
                     "hoursWorked": 4 },
                        { "empeId": "e001",
                     "projectId": "p003",
                     "hoursWorked": 2 },
                        { "empeId": "e002",
                     "projectId": "p003",
                     "hoursWorked": 5 } ]
} ] );
```

Use either a method $find()$ or a method $aggregate()$ available in MongoDB to write the implementations of the following queries. Implementation of each query is worth 1 mark.

(i) Find the first name (fName) and last name (lName) of all employee who have experience in Database Design. Do not show the object identifier (_id).

(1.0 mark)

(ii) Find the employee id (empeId) and hours worked in project (hoursWorked) of all employee who worked in project "p003".

(1.0 mark)

(iii) Find all employees who possess 4 experiences. Show only the employee's information.

(1.0 mark)

Use the method *update*() to write the implementations of the following data manipulation operations. Implementation of each data manipulation operation is worth 1 mark.

(iv) Add a new experience "HIVE" to the employee whose empeId is 'e001'.

(1.0 mark)

(v) Change the email account for employee e001 to "jamesbond$hotmail.com".

(1.0 mark)

End of specification

Answer:

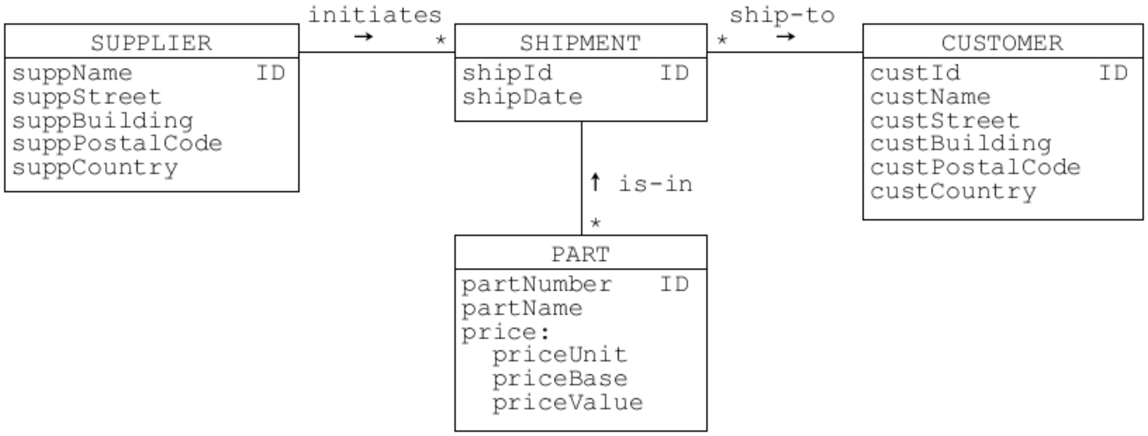
(a)

```
“Supplier”: {“suppName”: “sa la kau”
             “suppStreet”: “gangster street”
             “suppBuilding”: “gangster building”
             “suppPostalCode”: “369”
             “suppCountry”: “Singapore”
             “Initiates”: { “Shipment”: [      {      “shipID”: “369369”,
                                                      “shipDate: “03-06-2020”,
                                                      “ship-to”: {“Customer”: {
      “custID”: “001”,
      “custName”: “James”,
      “custStreet”: “aStreet”,
      “custBuilding”: “aBuilding”,
      “custPostal”: “12313”,
      “custCountry: “Singapore”
      }

                                               },

                                               {      “shipID”: “123456”,
                                                      “shipDate: “01-06-2020”,
                                                      “ship-to”: {“Customer”:

     “custID”: “002”,
      “custName”: “Tom”,
      “custStreet”: “aStreet2”,
      “custBuilding”: “aBuilding2”,
      “custPostal”: “1231333”,
      “custCountry: “Singapore”
      }

                                               }

                                        ]
                           }
```

```
“Part”: {“partNumber”: “23”,
“partName”: “L Shape”,
```

Answer:

```
“price: [{“priceunit” : “56”,
        “pricebase”: “12”,
        “priceValue”: “13”
        }]
“is-in”: {:Supplier.Initiates.Shipment.shipID: [“369369”]
}
```



Answer:

B (i)

```
db.empeProject.find({“Employee.experience{$exists: {“Database Design”}},{“fname”:1, “lname”:1, “_id”:0})
```

(ii)

```
db.empeProject.find({$and[{“Project.EmployeeProject.projectID”: “p003”},
{ Project.EmployeeProject.hoursWorked: {$gte: 3} }]},
{“Project.EmployeeProject.empID”:1 })
```

(iii)

```
db.empeProject.aggregation({“Employee.experience”: {$size: {$eq: 4}}}{$project:
{“Employee.fname”, “Employee.lname”, “Employee.email”}})
```

(iv)

```
db.empeProject.update({“Employee.empID”: “e001”}, {$push:
{“Employee.experience”: “Hive”})
```

(v)

```
db.empeProject.update({“Employee.empID”: “e001”}, {$set: {“Employee.email”:
“jamesbond@hotmail.com”})
```

Answer: