

**School of Computing and Information Technology****Student to complete:**

Family name	<input type="text"/>
Other names	<input type="text"/>
Student number	<input type="text"/>
Table number	<input type="text"/>

**CSCI235**  
**Database Systems****Final Examination Paper**  
**Session 2 2021**

Exam duration	3 hours and 40 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 in the answer booklet provided.

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

**Version 2.0**

## Question 4 (Total 10 marks)

### BSON document

**Time allocated: 50 minutes**

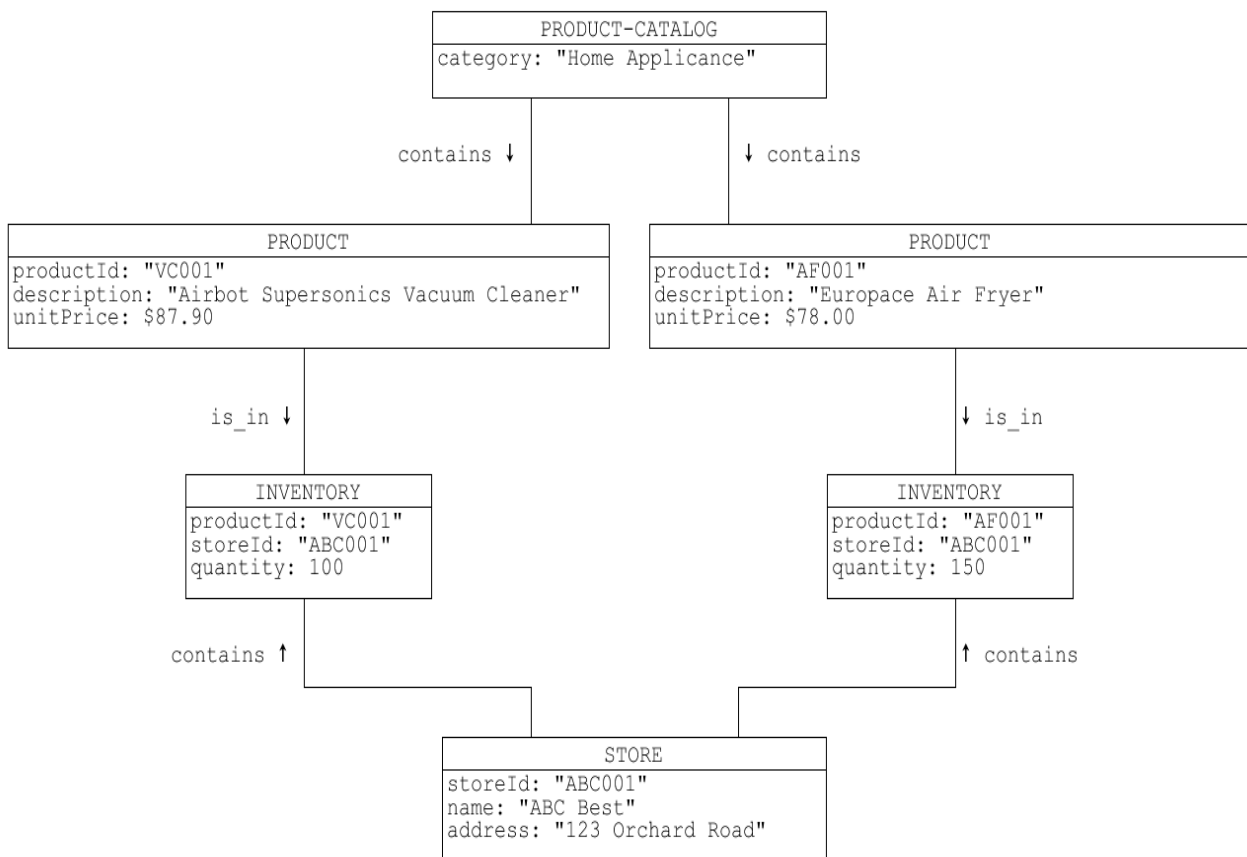
**Start time: 12:40 pm SGT**

**End time: 1:30 pm SGT**

**Submission time start: 1:25 pm SGT**

**Submission time end: 1:40 pm SGT**

- a) Consider the following class instance diagram of a sample database that contains information about a product-catalog, products, store, as well as the inventory for each product in the stores.



Write a sample JSON document that has a structure consistent with the class instance diagram given above.

**(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 first name (`fName`), last name (`lName`) and experience (`experience`) of the employee with employee id (`empeId`) = "e002". Display only the employee first name, last name and the experience. Do not display the object identifier (`_id`).

**(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 QUESTION 4

## END OF EXAMINATION