

**COMSATS University, Islamabad**

**Department of Computer Science**

**First Sessional Exam, Fall 2020**

**Part-2**

Class/Section: - BSCS – 7 C Marks: - 20

Subject: - Special Topics in CS-1 Time: - 30 Minutes

Instructor: - Muhammad Raza Tayyab Dated: - **April 01, 2021**

**Part C (Total Marks 20) CLO-2**

Create a mongoose schema naming students and courses. Students will have fields like name, age, year, email, phone number, course enrolled etc.

Courses will have fields like department, name, credit hours, subject Outline etc.

1) Create a schema for students and courses and create reference of courses in Students.

2) Insert a course and attach that course to all students in a specific year.

3) Using an aggregate pipeline, write a query to fetch all the courses with students.

4) Search all the courses from the course collection where credit hours are greater than equal to 3 and the department is “IT”.

**Answer 1**

const coursesSchema = mongoose.Schema({

    name: String,

    department: String,

    credit: Number,

    outline: String,

});

const studentSchema = mongoose.Schema({

   name: String,

   age: Number,

   year: Number,

   email: String,

   phone: String,

   courses: [mongoose.Schema.Types.ObjectId]

});

**Answer 2**

    Courses.findOne({name: /CS1/i}).exec((err, course) => {

        Students.find({year: 2021}).exec((err, stds) => {

            for (std of stds){

                Students.findByIdAndUpdate({\_id: std.\_id}, {courses: std.courses.concat(course.\_id)},

                (err) => {

                    if (err) throw err;

                    console.log("Std Updated");

                });

            }

        });

    });

**Answer 3**

**Answer 4**

    Courses.find({credit: {$gte: 3}, department: /IT/i}).exec((err, courses) => {

        if (err) throw err;

        console.log(JSON.stringify(courses));

    })